2013-08-30

An Agile Approach for Healthcare Simulation Modeling

Xu, Yunxiao Samuel

THE UNIVERSITY OF CALGARY

An Agile Approach for Healthcare Simulation Modeling

by

Yunxiao Samuel Xu

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN

PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

HASKAYNE SCHOOL OF BUSINESS

CALGARY, ALBERTA

AUGUST, 2013

Abstract

This research presents an agile healthcare simulation modeling (AHSM) approach to support the full life cycle of decision-making for managing the operations of multi-unit healthcare systems (MUHSs) in a rapid and flexible manner. The approach involves developing and maintaining simulation models at the unit level, and integrating the unit simulation models to address different decision problems at different levels of the MUHS. At the operational level, AHSM includes two stages: 1) ***Planning*** - identify the hierarchical structures of a MUHS and perform an architectural design of the overall SM studies; and 2) ***Executing*** - develop unit, subsystem and the overall MUHS simulation models. The existence of the MUHS hierarchical structures is supported by a conceptual model, nearly decomposable-complex adaptive systems (ND-CASs), which is introduced to represent and describe the operations of MUHSs.

Following the introduction of the AHSM approach, a system decomposition technique based on principal component analysis (PCA) is proposed for identifying the hierarchical structures of MUHSs with a large number of units, and a neural network (NN) based simulation metamodeling technique is suggested to simplify and represent unit simulation models. In addition, a flexible model integration (FMI) approach is recommended to guide the use of different model integration approaches in the development of MUHS simulation models.

The real-world potential of AHSM is demonstrated via three application examples.

Acknowledgements

I am very grateful to my supervisor, Dr. Paul Rogers, for providing guidance, direction and critique during the course of my doctoral studies. He not only gave me the support, advice and encouragement whenever needed to make this dissertation possible, but also provided me with many opportunities to get most out of this experience.

I am deeply thankful to Dr. Thomas Rohleder, who introduced me to the field of healthcare operations management and provided much valuable advice during my time at the University of Calgary.

I am also thankful to Dr. Diane Bischak who provided many valuable inputs to my research. Especially, I want to express my gratitude for her help in developing my teaching skills while in the OPMA department.

I also want to express my gratitude to Dr. Alejandro Ramirez-Serrano who provided many valuable suggestions for my research at the proposal stage. He also provided much constructive advice for improving the quality of the final version of my dissertation.

I also want to thank Dr. Tarek Y. ElMekkaway and Dr. Fiona M. Clement for spending time reviewing my dissertation and providing many valuable comments during the oral exam.

There are also many others at the University of Calgary who should be acknowledged, of whom only a few are mentioned here: Dr. David Cooke, Dr. Jaydeep Balakrishnan, and Dr. Giovani Caetano da Silveira.

I must also acknowledge financial support from the Health Quality Council of Alberta (HQCA) and the Calgary Health Region (CHR) via research grants on which Drs. Rogers and Rohleder were principal investigators.

Dedication

To my wife, Belinda and to my son, Michael

I will always be grateful for their patience, love and support!

Table of Contents

List of Tables

List of Figures

Table of Acronyms

A&E	Accident & Emergency
ABS	Agent-based Simulation
AD	Absolute Deviation
AHSM	Agile Healthcare Simulation Modeling
ALC	Alternative Level of Care
ANN	Artificial Neural Network
APD	Absolute Percent Deviation
BMJ	British Medical Journal
CAS	Complex Adaptive System
CHR	Calgary Health Region
CI	Confidence Interval
CLS	Calgary Laboratory Services
CRN	Common Random Numbers
CSP	Commercial Simulation Package
CTAS	Canadian Triage and Acuity Scale
DES	Discrete-event Simulation
DI	Diagnostic Imaging
DMI	Direct Model Integration
DS	Distributed Simulation
ED	Emergency Department
EP	Emergency Physician

FANN	Feed-forward Artificial Neural Network
FMC	Foothills Medical Center
FMI	Flexible Model Integration
HLA	High Level Architecture
HMI	Hybrid Model Integration
HQCA	Health Quality Council of Alberta
HSM	Healthcare Simulation Modeling
HW	Hospitalist Wards
LAMA	Leave Against Medical Advice
LoS	Length of Stay
LWBS	Leave Without Being Seen
MAD	Mean Absolute Deviation
MAPD	Mean Absolute Percent Deviation
MCS	Monte Carlo Simulation
MD	Medical Doctor
MET	Minor Emergency Treatment
MFNN	Multilayered Feed-forward Neural Network
MLP	Multilayer Perceptron
MMI	Metamodel-based Integration
MUHS	Multi-unit Healthcare System
ND	Nearly Decomposable
ND-CAS	Nearly Decomposable Complex Adaptive System

NN	Neural Network
OM	Operations Management
OR	Operations Research
PCA	Principal Component Analysis
RMSE	Root Mean Square Error
RSM	Response Surface Methodology
RTI	Runtime Infrastructure
SD	System Dynamics
SM	Simulation Modeling
SML	Standard ML (Programming Language)
UML	Unified Modeling Language
W.R.T.	With Regard To

Chapter 1 Introduction

1.1 Introduction

Healthcare systems are organizations of people, facilities and resources intended to deliver healthcare services to meet the health needs of target populations. In many western countries, healthcare systems are facing a crisis (Langabeer, 2008). On the one hand, populations are growing and aging which means that people are requiring more healthcare services, especially resource-intensive ones. On the other hand, as the major payers for healthcare expenditure, governments are trying to control the increase of healthcare spending in a time of global financial turmoil. In order to stay financially viable while delivering high quality healthcare services, healthcare organizations are under significant pressure to improve their operational efficiency. This leads to a great deal of attention from both academic and practitioner communities to the application of operations research (OR) tools in the field of healthcare operations management (Healthcare OM) to support decision-making at various levels.

Healthcare OM is defined as "*the analysis, design, planning, and control of all the steps necessary to provide a (healthcare) service for a client*" (Vissers & Beech, 2005). Healthcare OM problems are similar in many ways to traditional problems in OM (Brandeau et al., 2004), including strategic planning problems such as design of services, design of healthcare supply chains, facility planning and design, process selection, equipment evaluation and selection, and capacity planning. Healthcare OM problems also include tactical and operational planning problems such as demand and capacity

forecasting, capacity management, scheduling and staff planning, job design, and management of healthcare supply chains. Healthcare managers must measure and manage system performance and quality, manage inventory (e.g. supplies, drugs, or blood), and assess the performance of healthcare technologies.

However, healthcare has many unique characteristics when compared with other industries. Healthcare systems are intensely people-centred putting them among the most complex and dynamic systems in the world (Seila & Brailsford, 2009). Patients are people, healthcare services providers, e.g. nurses and physicians, are people, and healthcare managers and policy makers are people. The delivery process of healthcare services is the interaction between people, and is full of uncertainty because the condition of patients cannot be accurately predicted in advance, nor measured with certainty during the provision of care. There are many stakeholders involved in the operations of healthcare systems and they often have different or even conflicting views on how to operate the healthcare systems or evaluate the quality of healthcare services (Donabedian, 1966).

A typical healthcare system consists of many units. The term multi-unit healthcare system (MUHS) is used in this research as a generic term to represent any healthcare system consisting of two or more units. A unit in a MUHS is an independent entity providing a single service or a group of closely related services. For example, an emergency department (ED) or a group of EDs can be treated as a unit and the service managing the patient waiting lists can be treated as a unit as well (Langabeer, 2008). Though all the units in a MUHS need to work together to provide healthcare services, each unit also has

its own local and short-term priorities. The independent operations of each unit in a MUHS help create an environment that is less team-focused than in other industries. In addition, the operations of each unit in a MUHS and the MUHS as a whole must also deal with frequent changes that can happen at the organizational structure level and/or at the operational process level. All these factors can cause ambiguity in decision-making as necessary trade-offs need to be made between short-term/local and long-term/global priorities among the units.

1.2 The Challenges of Simulation Modeling in Healthcare

In the most generic sense, a model is a simplified representation of a system intended to help answers questions about the system (Kelton & Law, 2000). Modeling is the process of building, refining and analyzing that representation for greater insight and improved decision-making. Simulation is the manipulation of the inputs of a model to see how they affect the output measures of performance. A model to be studied by means of simulation is called a simulation model. Simulation modeling (SM) is the creation and use of simulation models to develop data as a basis for making various decisions. As shown in Figure 1-1, SM provides a virtual world as called by Schön (1983), for decision-makers to manage complex decision situations by focusing on the most important aspects while downplaying the less important ones. Given the dynamics and complexity of real systems, it can be very difficult if not impossible to solve many decision problems in the real world, which is represented in Figure 1-1 via the circle with the line through it. However, in the virtual world of models, a simulation model can be run to test different decision

alternatives and provide the conceptual solution, which can be used by decision-makers to make the final decision.

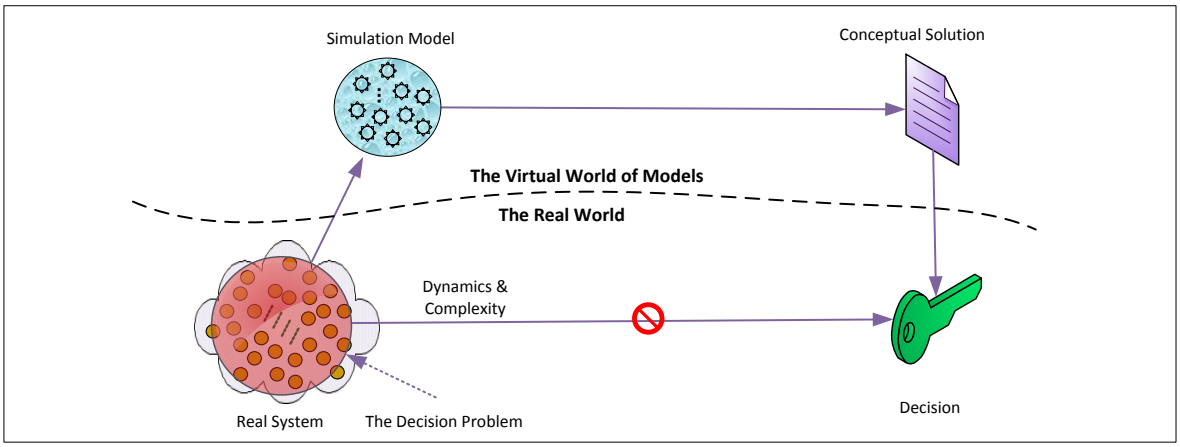


Figure 1-1 The Real World and the Virtual World of Models

SM is one of the most commonly used OR tools in healthcare and has been applied in the domain of healthcare for over four decades. It is regarded by many as the "technique of choice" for decision support in the complex and fast changing healthcare environment (Davies & Davies, 1994). However, SM has not yet become an integral part of analysis before major decisions in healthcare are made (Seila & Brailsford, 2009). Many researchers have tried to address this issue from different perspectives including: the complexity of healthcare systems (Jun et al., 1999); the availability of SM methods (Eldabi et al., 2007; Brailsford et al., 2006); the need to understand the priority of healthcare managers (Smith, 1995); unit- or facility-specific simulation models and their low reuse rate (Jun et al., 1999; Gunal & Pidd, 2009); and the speed of developing simulation models (Wilson, 1981).

The author believes that the two most important factors affecting the prevalence of SM applications in healthcare are concerns with rapidity and flexibility. Rapidity means how quickly a simulation model can be developed to address the decision problem. Flexibility means the reusability of existing simulation models or of knowledge acquired in previous SM projects to address new decision problems. Depending on the nature and domain of the decision problems, the measurement and evaluation of the rapidity and flexibility of SM applications can vary. For unit-/facility-specific decision problems, once the simulation model has been developed to represent the operations of the unit/facility, it is feasible to update that simulation model to reflect changes in the operations of the unit/facility. So the SM rapidity and flexibility can be measured and evaluated by analyzing the features of the simulation model, i.e. does it adopt an open structure to incorporate changes? Does it have a friendly user interface to accommodate the needs of users with different levels of knowledge on SM? But for decision problems concerning multiple units, the possibility of reusing an existing MUHS simulation model is relatively smaller as new decision problems can concern a different set of units. There could be a significant amount of work to develop a MUHS simulation model from another MUHS simulation model. So the SM rapidity and flexibility in MUHSs should be measured and evaluated by focusing on the SM process rather than only on the features of the simulation models.

As shown in Figure 1-2, current SM practices in healthcare do not provide the rapidity and flexibility required to support integrated decision-making for managing the operations of MUHSs. The unit- or facility-specific applications of healthcare SM (HSM) limit the flexibility of addressing various decision problems by reusing existing simulation models.

As a result, simulation models need to be developed from scratch almost every time when new decision problems occur, which in turn affects the rapidity with which SM can be used to address these decision problems.

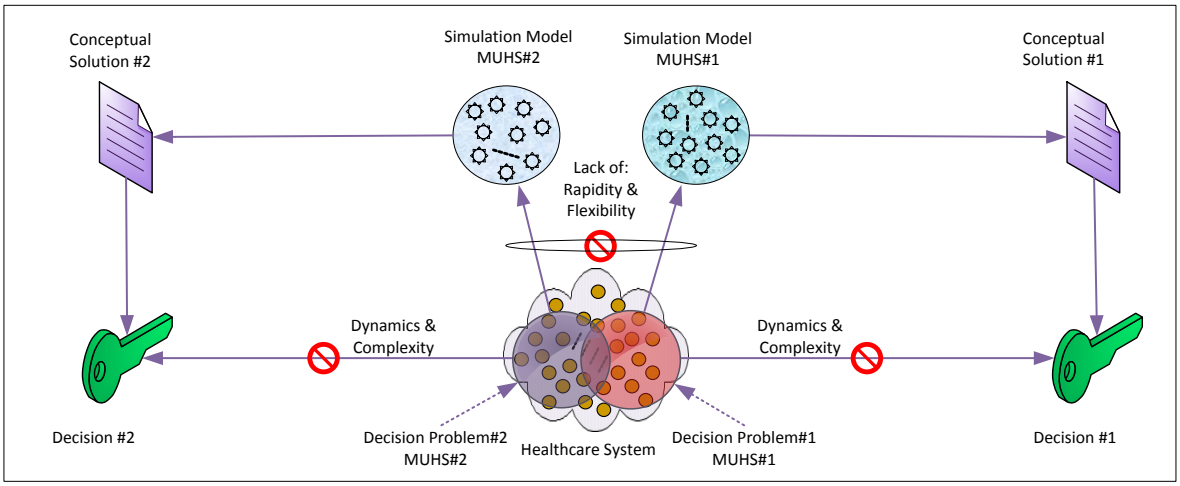


Figure 1-2 The Lack of Rapidity and Flexibility in the Current SM Practices in Healthcare

1.3 Research Motivations, Tasks and Approaches

This research is motivated to develop a HSM approach for improving the rapidity and flexibility of SM applications in MUHSs. For the HSM approach to be successful, it needs to address both the short-term and long-term requirements of SM applications in healthcare. To understand these requirements, this research makes the following main assumptions regarding healthcare SM applications:

Short-term

1) Healthcare organizations will not have in-house resources for the development and maintenance of large-scale healthcare simulation models.

2) Healthcare organizations will not invest money in developing or purchasing real-time software systems simulating the operations of the whole healthcare system.
3) The majority of SM applications in healthcare will still be unit-/facility-specific.
4) Healthcare organizations will continue to focus on improving the efficiency of their operations.
5) Senior management will have growing interest in SM applications in MUHSs, but is still hesitant to invest money in large-scale SM projects covering multiple units. Thus most of the SM applications in MUHSs will still be driven by specific decision problems.

Long-term

1) Healthcare organizations will still to focus on improving the efficiency of their operations.
2) SM will become the essential supporting tool in making decisions concerning the operations of MUHSs.
3) However, the majority of SM applications in healthcare will still be unit-/facility-specific, supporting the daily operations of the units.

Based on the above assumptions, one possible solution is to separate the development and maintenance of the unit simulation models from the use of these simulation models. The maintenance of the unit simulation models means updating the simulation models periodically to reflect the changes in the units.

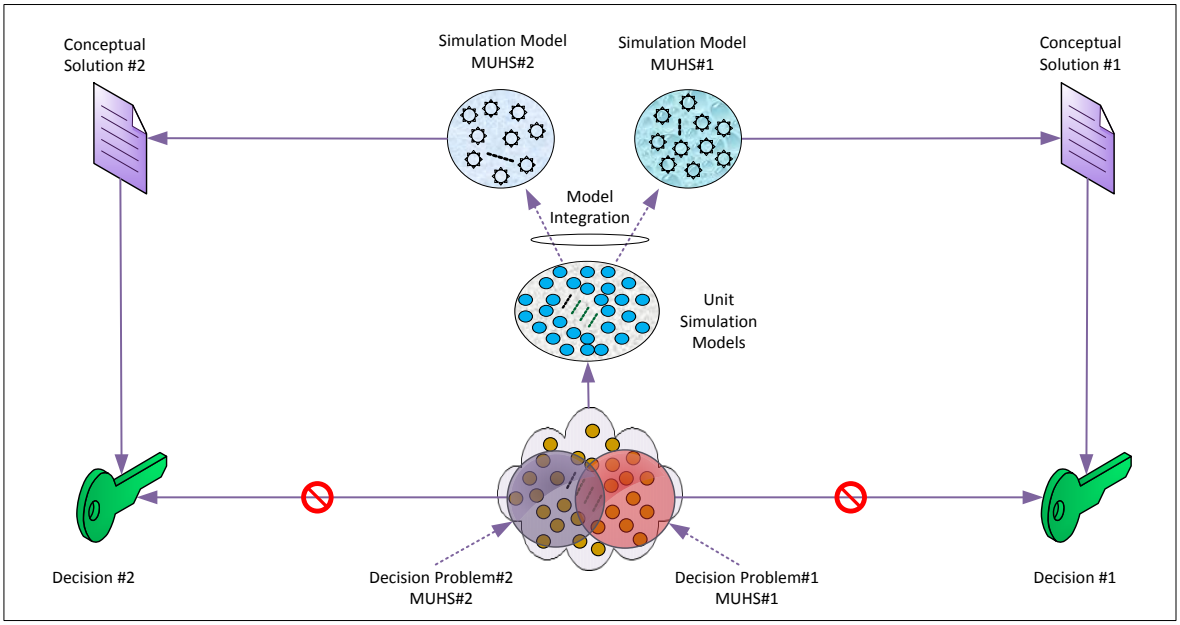


Figure 1-3 Separating the Development and Maintenance of the Unit Simulation Models from the Use of these Models

As shown in Figure 1-3, simulation models can be developed and maintained at the unit level. When new decision problems occur, these unit simulation models can be integrated to simulate the operations of the MUHS, within which the decision problems occur.

Conceptually, the proposed solution is very simple and straightforward; however, there are two major challenges to operationalize this solution, as highlighted in Figure 1-4. The first one is how to map the real-world decision problem into the virtual world of models and decide which unit simulation models need to be developed or reused? The second one is once the unit simulation models have been identified, how can they be efficiently integrated together to represent the real-world decision problem?

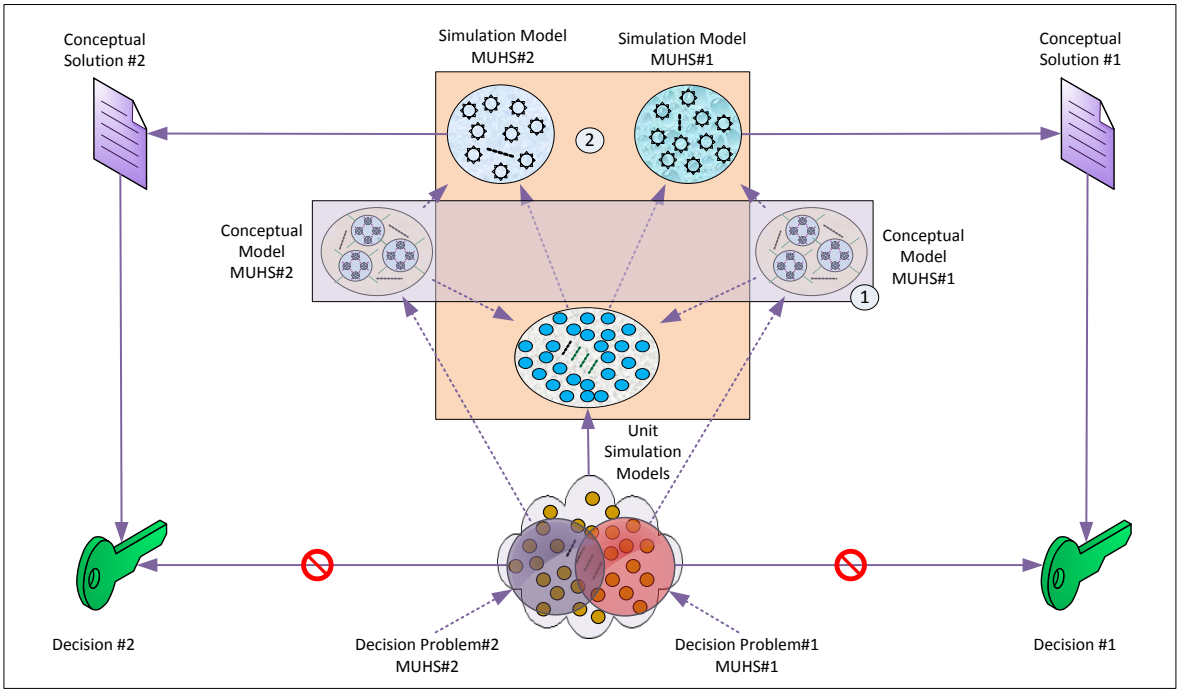


Figure 1-4 Operationalization of the Proposed Solution

This research is intended to tackle these two challenges with four planned research tasks. These research tasks and the corresponding research approaches are described as follows:

Research Task #1: Literature Review

The existing literature on related topics covered in this research is examined. The literature review is divided into two parts. The first deals with the complexity and dynamics of healthcare systems with a focus on the following two frequently quoted theories on complex systems: complex adaptive systems (CASs) and nearly decomposable (ND) systems. The second deals with various SM methods and their applications in healthcare.

Research Task #2: Develop a conceptual model to represent and describe the operations of MUHSs. This conceptual model is intended to serve as a bridge between the real world and the virtual world of models.

Research Task #3: Develop a HSM approach to effectively support the full lifecycle of decision-making for managing the operations of MUHSs in a rapid and flexible manner.

Research Task #4: Demonstrate the real-world potential of the proposed HSM approach via application examples

1.4 Organization of the Dissertation

The remainder of the dissertation is structured as follows:

Chapter 2 reviews related literature on the complexity and dynamics of healthcare systems, the theories of CASs and ND systems, and different SM methods and their applications in healthcare.

Chapter 3 introduces a conceptual model, nearly decomposable-complex adaptive systems (ND-CASs), to represent and describe the operations of MUHSs. The ND-CASs model is developed by integrating the key concepts from the theories of CASs and ND systems. Its potential benefits for supporting SM-based decision-making in MUHSs are also discussed.

Chapter 4 presents an agile HSM (AHSM) approach for the development of MUHS simulation models in a rapid and flexible manner.

Chapter 5 discusses three techniques/approaches to support the implementation of AHSM: system decomposition; neural network (NN) based simulation metamodeling; and flexible model integration (FMI).

Chapter 6 to 8 inclusive demonstrate the real-world potential of AHSM via three application examples. Chapter 6 shows how to decompose a MUHS into hierarchical layers of subsystems using the healthcare system in the Calgary Health Region (CHR) as an example. Chapter 7 uses an ED DES (discrete-event simulation) model as an example to show how to develop simulation metamodels by training NN models with the experimental outputs of the DES model. Chapter 8 demonstrates the application of the FMI approach with a focus on the underlying MMI approach in integrating multiple unit simulation models to develop the MUHS simulation model. The software packages used in the application examples are listed in Appendix A.

Chapter 9 summarizes the key findings, contributions and limitations of the research, and also explores some future research opportunities.

Chapter 2 Literature Review

2.1 Healthcare Systems as Complex Systems

A system is defined as "*a set or arrangement of things so related or connected as to form a unity or organic whole*" (Webster's New World Dictionary). The "things" in this definition may also be referred to as "elements" in other definitions. A complex system is a system consisting of a wide variety of elements which interact with each other in a nonlinear way. The study of complex systems can be referred to as complexity theory (McDaniel & Driebe, 2001). Complexity theory is not a single theory but instead it consists of a group of theories including CASs theory, ND systems theory, and quantum and chaos theory.

Applications of complexity theory in healthcare started to emerge in the mid-1990s. For example, a series in *Quality Management in Health Care* examines clinical pathways as nonlinear and evolving systems (Sharp & Priesmeyer, 1995; Priesmeyer & Sharp, 1995; Priesmeyer et al., 1996). McDaniel (1997) and McDaniel & Driebe (2001) construct the leadership imperatives of healthcare executives from the perspective of quantum and chaos theory, and apply complexity theory to the process of management in healthcare delivery. Zimmerman et al. (1998) contribute a primer on complexity theory with nine management principles for leadership and management in healthcare systems. Begun & White (1999) use complexity theory to examine the inertial patterns in the nursing profession and their resistance to change. Marion & Bacon (2000) discuss the fitness of three eldercare organizations from the perspective of complexity theory, emphasizing that

organizations are embedded in larger systems. Dooley & Plsek (2001) use models of complex natural processes to explain the generation of medication errors in hospitals and conclude that the recommendations in the Institute of Medicine report (Kohn et al., 2000) fail to lead to significant organizational learning and improvement. In 2001, a series of articles in the BMJ (British Medical Journal) marks the official arrival of the "received interpretation" of complexity in healthcare (Paley, 2010). Almost all the authors who write on this topic later cite one or more articles in this series (Plsek & Greenhalgh, 2001; Wilson & Holt, 2001; Plsek & Wilson, 2001; Fraser & Greenhalgh, 2001). Since then, there has been a growing interest in applying complexity theory to healthcare systems (Dattee & Barlow, 2010). Healthcare systems are often referred to as "*excellent examples of complex systems*," and should be studied as whole systems (Rowlands et al., 2005).

Before the introduction of complexity theory to healthcare, the "machine" metaphor was dominant in thinking about healthcare systems (Begun et al., 2002), involving receiving inputs, transforming them, and producing outputs, such as improved health. This machine metaphor shapes beliefs on how the "system" can be studied and improved (Morgan, 1997), i.e. break it into parts, and examine each part separately to understand its mechanics. If the system is not working as expected, then identify the problematic part and replace it. Yet, thinking of and operating healthcare systems as machines have not led to effective research and practice in healthcare. Healthcare delivery has failed repeatedly to achieve acceptable levels of satisfaction for both internal and external stakeholders (Begun et al., 2002). In the meantime, it is increasingly evident that healthcare systems do not meet the mechanistic expectations as suggested in the machine metaphor – they are much more complex and "messier" (Sibthorpe et al., 2004). As argued by Begun et al.

(2002), the introduction of complexity theory in healthcare broadens and deepens the scope of inquiry into healthcare systems and expands corresponding research methods.

2.2 Theories of Complex Systems

There are two frequently used theories on complex systems: the CASs theory and the ND systems theory. Both theories focus on certain aspects of complex systems, and their details are described in the following sections.

2.2.1 *The CASs Theory*

The CASs theory is one of the most popular complexity theories used in the study of complex systems (Sibthorpe et al., 2004). A CAS is defined as "*being composed of populations of adaptive agents whose interactions result in complex nonlinear dynamics, the result of which are emergent system phenomena*" (Brownlee, 2007). All CASs share some characteristics in common. The major ones include: dynamic, massively entangled, scale independent, transformative and emergent (Eoyang & Berkas, 1998). First, there are a large number of agents in a CAS. These agents are interconnected and open to external influences. As a result, changes in a CAS are constant and discontinuous, and a CAS exists in a state of dynamic flux (Briggs & Peat, 1989). Second, relationships in a CAS are complicated and enmeshed or "massively entangled" as described by Kontopoulos (1993). In a CAS, the adaptive agents and the variables describing those agents are large in number (Kelly, 1994). In addition to being numerous, variables and the interactions between the agents can be discontinuous and nonlinear. Some dimensions may lie dormant for a long period till they are sparked into action when some control parameters

reach critical values. Third, a CAS functions simultaneously at many different scales of organization (West & Deering, 1995). Individual agents take relatively independent actions, various grouping of agents emerge in the dynamical course of events, and the system as a whole exhibits identifiable behaviours. Behaviours in each of these domains are both similar to and different from behaviours of the others. Fourth, the agents in a CAS are open systems. They are transformed and transforming in their interactions. Agents do not simply adapt to the environment and each other. They co-evolve with each other and with the environment (McDaniel & Driebe, 2001). A CAS as a whole is also an open system, and transformation occurs across the system's external boundaries. Both changes and stability can be generated in the system via the feedback loops. Fifth, a CAS has mechanisms for self-organization and emergence. Self-organization is defined as "*a dynamical and adaptive process where systems acquire and maintain structure themselves, without external control*" (De Wolf & Holvoet, 2005). The "structure" can be spatial, temporal or functional. "No external control" means the absence of manipulation, direction, interference or involvement from outside of the system. Emergence denotes the principle that the global properties defining higher order systems or "wholes" can in general not be reduced to the properties of the lower order subsystems or elements (Heylighen, 1989). Such irreducible properties are called emergent.

De Wolf & Holvoet (2005) identified many characteristics of emergence through an extensive literature study. The most important ones are summarized as follows:

1) **Micro-macro Effect:** A micro-macro effect refers to properties, behaviours, structures, or patterns that are situated at a higher macro-level and arise from

interactions at the lower micro-level of the system. The "level" mentioned refers to a particular viewpoint. The macro-level considers the system as a whole and micro-level considers the system from the viewpoint of the individual elements that make up the system.

2) **Radical Novelty:** The global behaviours are novel w.r.t. (with regard to) the individual behaviours at the micro-level, i.e. the individuals at the micro-level have no explicit representation of the global behaviour. The global behaviours cannot be studied by physically taking a system apart and looking at the parts. However, they can be studied by looking at each of the elements in the context of the system as a whole.
3) **Coherence:** Coherence refers to a logical and consistent correlation of elements. Emergent behaviour appears as an integrated whole that tends to maintain a certain sense of identity over time (i.e. a persistent pattern).
4) **Interacting Elements:** The elements need to interact. Without interactions, interesting macro-level behaviours will never arise.
5) **Dynamical:** Emergent behaviour arises as the system evolves over time. An example of such is the new kind of behaviour that becomes possible at a certain point in time.
6) **Decentralized Control:** Decentralised control is using only local mechanisms to influence the global behaviour. There is no central control, i.e. no single part of the systems can direct the macro-level behaviour. This characteristic is a direct consequence of the radical novelty that is required for emergence.
7) **Two-way Link:** There is a bidirectional link between the macro-level and the micro-level. From the micro-level to the macro-level, the elements give rise to an emergent

structure. In the other direction, the emergent structure influences its elements. For example, path-formation with ants: the emergent path influences the movements of the micro-level ants because they follow the pheromones.

8) **Robustness and Flexibility:** Emergent behaviour is relatively insensitive to perturbations or errors as no single element is "in charge". The failure or replacement of a single element will decrease performance but will not cause a sudden loss of function. This flexibility means that the individual elements can be replaced, but the emergent structure can remain.

2.2.2 *The ND Systems Theory*

In his classic paper, 'The Architecture of Complexity', Simon (1962) proposes that hierarchy is a universal principle of the structure of complex systems as hierarchy is everywhere. Simon provides many examples of hierarchy in biological, physical, symbolic, and social systems. He emphasizes that hierarchy doesn't need to imply top-down relations of authority though he mentions organizations as examples of hierarchy. He also acknowledges that not all structures are hierarchical, giving the example of polymers, which are long chains of identical units, "*However*," he says, "*for present purpose we can simply regard such a structure as hierarchy with a span of one – the limiting case*."

Simon argues that hierarchical systems share the ND property: they are organized into hierarchical layers of subsystems in such a way that interactions between elements belonging to the same subsystems are much more frequent and intensive than interactions between elements belonging to different subsystems. He calls these systems ND systems

and in a more recent article writes that a ND system "*can be thought of as a boxes-within-boxes hierarchy with an arbitrary number of levels. Its special characteristic is that equilibrating interactions within boxes at any level take place much more rapidly than the interactions between boxes at that same level, and similarly all the way to the top of the hierarchy*" (Simon, 2002). Simon's argument is based on a variation-and-selection view of natural (and artificial) evolution: elements are interconnected and combined by natural interactions (or, equivalently, by the trials of a problem-solver), which vary in frequency and intensity, thus creating a variety of assemblies. Of these assemblies, only those which are sufficiently stable will survive, the other assemblies will fall apart before they can undergo any further evolution. The stable assemblies can then again function as building blocks, to be combined into higher level assemblies, and so the process can repeat itself at ever higher levels, forming a set of hierarchically structured complex systems. Simon then uses this "evolutionary process" to explain that ND systems are more probable to emerge because of the existence of intermediate stages. According to Simon, "*if we begin with a population of systems of comparable complexity, some of which are ND and some of which are not, the ND systems will, on average, increase their fitness through evolutionary processes much faster than the remaining systems, and will soon come to dominate the entire population*."

Simon's ND systems theory has been used by many scholars in the design and analysis of complex systems in many fields. A recent check on Google Scholar shows that Simon's 1962 paper has been directly quoted by over 3,800 papers, some of which have been quoted by thousands of other papers. A main reason is that the analysis of the behaviour of a ND system can be subdivided, at least approximately, into the analysis of each of its

subsystems, combined with an aggregated analysis of their interactions. This is of profound value from a practical perspective as it offers an efficient approach to managing complexity in analyzing complex systems.

One thing worth mentioning is that ND is not modularity though some scholars have noted that there is a close link between the two concepts (Egidi & Marengo, 2002). Modularity suggests complete decomposability and tends to provide a role for the "*whole*" merely as an aggregation of the elements. In contrast, ND preserves the vital role of overall identity for the system while allowing it to reap many of the benefits of modularity. In other words, ND emphasizes sustained organizational identity through continual changes both in the inner and outer environments.

2.3 The Social Roles of Simulation Models

Simulation models can be applied in different ways. Bayer et al. (2010) call these different applications of simulation models the *social roles* that simulation models can play. To analyze the social roles of simulation models, they develop a 2 x 2 framework as shown in Table 2-1.

Table 2-1 A Framework to Classify the Social Roles of Simulation Models

	Epistemic Object (Create Knowledge)	**Technical Object** (Make Knowledge Available)
Boundary Object (Facilitate Communications across boundaries)	Learn as Group	Express and Experiment
Representative Object (Represent Reality)	Explore	Predict

Along one dimension, simulation models can be classified as boundary objects and representative objects. Boundary objects are artefacts shared between communities of practice, which have their own specific informational codes (Carlile, 2002; Sapsed & Salter, 2004). Boundary objects can be used to address some of the difficulties of communicating and creating knowledge across disciplinary and organizational boundaries or between disparate stakeholders. Representative objects are artefacts to represent a reality whose existence is seen principally as a fact. Along the other dimension, simulation models can be classified as epistemic objects and technical objects. Epistemic objects are fluid and help to create knowledge, while technical objects are static and seen as basic tools to make knowledge available (Ewenstein & Whyte, 2009).

These two dimensions therefore allow a classification of four types of social role for simulation models. As boundary objects, simulation models can be used as epistemic objects by stakeholders to create new knowledge as a group ("Learn as a Group"), or as technical objects to make knowledge available across the group by expressing the knowledge in a form accessible to others and enabling them to experiment with that

knowledge, i.e. showing what would happen under different scenarios ("Express and Experiment"). As representative objects for a reality, simulation models can be explored as management flight simulators (epistemic objects) to allow the users to learn ("Explore"), or as predictive tools (technical objects) allowing the user to draw on the knowledge embodied in the simulation models without the need to understand the relationships within the system ("Predict").

As pointed out by Bayer et al. (2010), these four types are idealized types. In practice, the social roles of simulation models can be a mixture of any of the four types as different stakeholders might have different views of the role of a simulation model. For instance, a client might want to develop a simulation model as a predictive tool while the modeling process might show that what is required would be to learn as a group. Further, over time the social role of a simulation model might change. For example, learning as a group can be followed by expressing the knowledge for others to experiment, which can be followed by the development of a predictive tool for other users or the development of a management flight simulator as a learning environment for other managers to explore.

2.4 SM Methods

2.4.1 *Basic SM Methods*

There are three basic SM methods that have been applied in healthcare: DES, SD (system dynamics) and ABS (agent-based simulation) (Seila & Brailsford, 2009). Each of these basic SM methods represents a particular world view, which is defined as the overall perspective from which the system is seen and interpreted.

DES is the modeling of systems in which the state variables change only at a discrete set of points in time (Zeigler et al., 2000). DES is based on the concepts of entities, resources and block charts describing entity flow and resource sharing. Entities are objects which represent patients, clients, tasks, parts, documents, etc. They move through the blocks of the flowchart where they stay in queue, receive services, and seize and release resources.

SD is "*the study of information-feedback characteristics of industrial activity to show how organizational structure, amplification (in policies), and time delays (in decisions and actions) interact to influence the success of the enterprise*" (Forrester, 1961). In SD, the real-world processes are represented in terms of stocks (e.g. stocks of material, knowledge, people or money), flows between these stocks, and information that determines the value of the flows. SD only works with aggregates, viewing the items in the same stock as indistinguishable (they do not have individuality). The primary assumption in SD is that the internal causal structure of a system determines its dynamic tendencies. Mathematically, an SD model is a system of integral equations (Sterman, 2000). To approach a problem in SD style, one has to think in terms of global structural dependencies and describe the system behaviour as a number of interacting feedback loops, balancing or reinforcing, and delay structures.

ABS is a relatively new approach to modeling complex systems (Macal & North, 2010). In an ABS model, natural entities i.e. human beings, animals, processes or systems are represented as agents that follow local behaviour rules to interact with each other and with the environment. The behaviours of the agents are defined using state charts. A state chart is a state machine adopted by the worldwide community and included as a part of the

standard UML (Unified Modeling Language) (Borshchev & Filippov, 2004). With state charts, one can graphically capture the different states of the agents, the transitions between these states, the events that trigger those transitions, the timing of transitions, and the actions that the agents take during their lifetime.

Many scholars have compared DES, SD and ABS from different perspectives. On the comparison between DES and SD, Randers (1980) uses nine axes to represent the criteria for comparison: relevance (usefulness), transparency, mode reproduction ability, descriptive realism, insight generating capacity, point predictive ability, formal correspondence with data, fertility, and ease of enrichment. Writing from the perspective of an experienced SD modeler, Lane (2000) identifies conceptual differences between DES and SD in terms of the following eight categories: perspective, resolution, data sources, problem studied, model elements, human agents involved, clients' perception about the model, and outputs. Brailsford & Hilton (2000) compare DES and SD in healthcare studies and identify the main technical distinctions using the following criteria: importance of variability, importance of tracking individuals, number of entities included in a typical study, system control mechanism, relative timescale, and purpose of the study. Sweetser (1999) provides a similar comparison focusing on the capacities of the two approaches. The criteria he uses include the role of the system structure in determining the performance, the engagement of mental models, system orientation, the role of computer simulation, and validity. Morecroft & Robinson (2008) provide an empirical comparison between DES and SD by developing an SD and DES model of the same problem situation. Their comparison focuses on the impacts of SD and DES on the representation and interpretation of phenomena from the real world. Chahal & Eldabi (2008a) provide a

comprehensive meta-comparison of DES and SD. They emphasize that it is important to understand from system, problem and methodology perspectives in order to choose a suitable SM method for the system under investigation

On comparing SD and ABS, Parunak et al. (1998) provide a comparison between SD and ABS from both a high-level view and a practical view. The comparison on the high-level focuses on two areas: the fundamental relationships among entities that they model, and the level at which they focus their attention. The comparison from the practical view focuses on three areas: the underlying structure of a model, the naturalness of the representation of a system, and the verisimilitude of a straightforward representation. Scholl (2001) compares the application of SD and ABS in the explanation of the bullwhip effect in supply chain management. Sun & Cheng (2002) review the application of SD and ABS in many contexts such as smallpox transmission after a bioterrorist attack, the spreading dynamics of infectious disease, and compare the two approaches using the following criteria: computing resources required, individual characteristics being modeled, accurateness of the result, visibility of interaction among individuals, network structure for interaction, difficulty of model construction, most appropriate domain of application, expression of relationships, and representation of time. Schieritz & Milling (2003) compare the primary "*conceptual predispositions"* underlying the two approaches using the following criteria: basic building block, unit of analysis, level of modeling, perspective (macro vs. micro), adaptation of structure, handling of time, mathematical formulation, and origin of dynamics. Wakeland et al. (2004) compare the application of SD and ABS in the study of cellular receptor dynamics in terms of overall approach, mathematics, ease of communications, research relevance, and educational potential.

Pugh (2006) and Yu et al. (2007) present a comparison between DES and ABS model characteristics. Pugh (2006) focuses on the use of DES and ABS to represent M/M/1 queuing systems, while Yu et al. (2007) focus on the use of DES and ABS in the field of transportation. A more recent comparison of DES and ABS is presented by Abdul (2011) on the use of DES and ABS to model human behaviour.

A comparison of the three SM methods is presented by Lorenz & Jost (2006) and Owen et al. (2008). Their studies have sought to establish a framework to assist new simulation users in choosing the right SM methods. Lorenz & Jost (2006) focus on developing a framework for multi-method SM within social science, while Owen et al. (2008) focus on developing a framework for supply-chain practitioners.

Table 2-2 shows a high-level summary of the comparison of DES, SD and ABS identified in the literature mentioned above. Detailed explanations are provided in the following paragraphs.

Table 2-2 Comparison of DES, SD and ABS

Items	DES	SD	ABS
Perspective	Macro	Macro	Micro
System Orientation	Focuses more on modeling the operation process of the system to be modeled	Focuses more on system	Focuses more on modeling individual behaviours
System Behaviour	Is the result of randomness and structure (model)	Is the result of structure (model)	Emerges as the result of interaction among agents
System Representation	Queues, activities and processes	Levels, flows and causal feedback loops	Agents, local rules and interactions
Resolution of Models	Flowchart blocks and Entities (passive objects)	Stocks and flows	Agents (Intelligent objects)
Handling of Time	Discrete	Continuous	Discrete
Computer Animation	High	Low	Medium
Computational Demand	High	Low	High
Model Validation	Emphasis on model structure and outputs	Emphasis on model structure and engagement of the clients' mental models	Emphasis on the local rules of each agent and the interactions among agents
Organization Level	Tactical & operational level	Strategic level	Strategic, tactical, & operational level
Application Orientation	Optimization, prediction and comparison	Problem-solving and understanding	Learning

DES takes a macro view and focuses on the modeling the operation process of the system to be modeled (Brailsford & Hilton, 2000; Chadal & Eldabi, 2008a). DES is based on the concept that the system behaviour is determined by randomness and the internal structure of the system (Sweetser, 1999; Morecroft & Robinson, 2006; Chadal & Eldabi, 2008a). In DES, systems are represented as queues, activities and processes with flowchart blocks and entities and their states change at discrete points in time (Morecroft & Robinson, 2006; Chadal & Eldabi, 2008a). DES is inherent with high capabilities of computer animation where entities can be shown moving through the system. This helps in visual understanding of process flow (Chadal & Eldabi, 2008a). DES has a high demand on computing resources and the validation of a DES model focuses on the validity of its internal structure and model outputs (Brailsford & Hilton, 2000; Chadal & Eldabi, 2008a). The applications of DES are mainly focusing on the tactical and operational levels for the purpose of optimization, prediction, and comparison of different alternatives (Brailsford & Hilton, 2000; Lorenz & Jost, 2006; Owen et al., 2008).

Like DES, SD also takes a macro view (Lane, 2000). However, SD focuses more on modeling the system (Chadal & Eldabi, 2008a). In SD, the system behaviour is viewed as the result of the feedback loop structure of the system (Brailsford & Hilton, 2000). In a SD model, the real world is represented as stocks (also known as levels), flows (also known as rates) and causal feedback loops (Morecroft & Robinson, 2008). Time in a SD model is handled by a discretization process where the time-step, $\boldsymbol{d_t}$, is usually chosen such that all the rates can be regarded as constant over the period $\boldsymbol{d_t}$ (Brailsford & Hilton, 2000). The animation of SD is limited to figures and graphs of model variables (Chahal &

Eldabi, 2008a). SD has a lower demand on computing resources (Brailsford & Hilton, 2000). The validation of a SD model focuses on the validity of the structure and the mental models engaged in the model (Chahal & Eldabi, 2008a). The applications of SD models focus mainly on the strategic level, and the purpose of the model is to solve problems and gain understanding of the problem situation (Brailsford & Hilton, 2000; Lorenz & Jost, 2006; Owen et al., 2008).

Unlike DES and SD, ABS takes a micro view and focuses on modeling individual behaviours (Schieritz & Milling, 2003; Abdul, 2011). From the perspective of ABS, the behaviour of the system emerges as the result of interacting agents (Wakeland et al., 2004). In ABS, systems are represented as intelligent agents, with local rules, and interactions among the agents (Borshchev & Filippov, 2004; Pugh, 2006; Yu et al., 2007). In an ABS model, the changes of system state happen at discrete points of time (Sun & Cheng, 2002; Schieritz & Milling, 2003). ABS has a very good animation capability for showing the behaviour of individual agents and the change of system behaviour over time (Wakeland et al., 2004). ABS also has a high demand on computing resources (Sun & Cheng, 2002). The validation of an ABS model focuses on the validity of the local rules and the interactions among the agents. The application of ABS can be at any level, strategic, tactical or operational, and the main purpose of ABS model is to learn about the behaviour of the system (Borshchev & Filippov, 2004; Lorenz & Jost, 2006; Owen et al., 2008).

2.4.2 *Hybrid Simulation and Distributed Simulation*

In order to facilitate the simulation study of MUHSs, several authors have suggested the use of hybrid simulation approach. For example, Chahal & Eldabi (2008b) discuss the combination of simulation methods in the healthcare domain and propose three types of hybrid: (i) hierarchical mode, where there exist two distinct simulation models working off-line, for example, a DES model feeds an SD model; (ii) process environment, where there are again two distinct models but this time one includes the other, for example a DES model resides inside an SD model; and (iii) integrated mode, where there exists a single model in which multiple simulation methods work "*inline"*. Examples of the first and the second types of hybrid simulation exist, such as for DES and SD by Chahal & Eldabi (2008b) and Brailsford et al. (2010), and for DES and ABS by Leonard et al. (2006) and Vieira et al. (2010). There are no examples yet in the literature of the last type (Gunal, 2012).

Distributed simulation (DS) is a technique for the development of large-scale simulation models. It can be defined as the distribution of the execution of a single run of a simulation program across multiple processors (Fujimoto, 1999 and 2003). To provide a standard DS approach, the IEEE 1516 standard, the High Level Architecture (HLA) (IEEE 1516, 2000), was published in 2000 (and updated in 2006). In the HLA, a distributed simulation is called a federation, and each individual simulator is referred to as a *federate*. A HLA Runtime Infrastructure (RTI) provides facilities to enable *federates* to interact with one another, as well as to control and manage the simulation.

The application of DS in healthcare is a very recent emergence, and there are only a couple of published papers on healthcare models built using DS. Brailsford et al. (2006) recommend DS using low cost CSPs (Commercial Simulation Packages) as a viable solution to model large-scale complex healthcare systems. Katsaliaki & Mustafee (2010) also investigate the viability of using DS to execute large and complex health care simulation models which help government take informed decisions. They find that compared with the standalone simulation, the DS approach decreases significantly the run time for large and complex models. However, they also find that the DS approach is only viable when: 1) the model can be divided into logical parts and the exchange of information between these parts occurs at constant time intervals; 2) the model is sufficiently large and complicated, such that executing the model over a single processor is excessively time consuming.

2.5 HSM Applications

2.5.1 *Review of HSM Applications*

HSM is a very active area of research and practice, and its application has spanned over four decades. Many literature reviews on HSM exist already, including early reviews by Valinsky (1975), England & Roberts (1978), Smith-Daniels et al. (1988), and Lehaney & Hlupic (1995). More recent reviews include those by Jun et al. (1999), Fone et al. (2003), Jacobson et al. (2006), Gunal & Pidd (2009), Mustafee et al. (2010), and Mielczarek & Uzialko-Mydlikowska (2012). There are also many specialized topic reviews available, for example, Cooper et al. (2006) review the application of SM methods in the treatment

of coronary heart disease and Vanberkel et al. (2009) review healthcare models that encompass multiple units.

The review conducted by Jun et al. (1999) is one of the recent reviews which takes a more comprehensive and systematic approach than is taken in some of the earlier reviews. The authors concentrate on DES applications to single clinics and to multi-facility health care clinics and create a broad taxonomy of articles describing DES models that address problems encountered by health care clinics during a period of about 30 years ending in 1997. They distinguish between two main areas: (1) patient flow, which includes patient scheduling and admissions, patient routing and flow schemes, and scheduling and availability of resources; and (2) allocation of resources, including bed sizing and planning, room sizing and planning, and staff sizing and planning. Jacobson et al. (2006) build on the work of Jun et al. (1999) and provide new updates that had been reported from 1999 through 2006. Their analysis focuses on the same two areas as used in Jun et al. (1999).

Fone et al. (2003) provide another systematic review of the literature on DES studies in healthcare between 1980 and 1999, most of which are covered in Jun et al. (1999). However, Fone et al. (2003) take a very different approach from the one used in Jun et al. (1999). Their review aims to assess the quality of the published HSM studies and to consider their influence on policy, rather than on operations. They enumerate five topic areas: 1) hospital scheduling and organization; 2) infection and communicable disease; 3) costs of illness and economic evaluation; 4) screening; and 5) miscellaneous.

Gunal & Pidd (2009) conduct a more recent review on DES studies in healthcare built on Jun et al. (1999) and Fone et al. (2003). This review focuses on papers published since 2004 with an interest in performance modeling within hospitals. The authors classify the papers based on the application areas being focused on. The application areas identified include A&E (Accident & Emergency), inpatient facilities, outpatient clinics, other hospital units (operating rooms, critical care unit, laboratory, pharmacy, etc.), and whole hospital simulations.

Mustafee et al. (2010) is one of the few papers reviewing the applications of multiple SM methods in healthcare. In this paper, the authors profile health care simulation literature published between 1970 and 2007 along two dimensions: 1) subject categories including operational research, health economics and pharmacokinetics; and 2) the application of four SM methods, namely, DES, MCS (Monte Carlo Simulation), SD, and ABS. A more recent but similar survey on the applications of computer simulation models in healthcare is conducted by Mielczarek & Uzialko-Mydlikowska (2012). The authors use the same two-dimension classification of SM applications in healthcare: 1) subject categories; and 2) SM methods. Along the dimension of the subject categories, the authors distinguish between five major groups of simulation models applied in health services: 1) Epidemiology, health promotion, health policy (disease prevention); 2) health and care systems operation; 3) health and care systems design; 4) medical decision-making; and 5) extreme events planning. Along the dimension of SM methods, the authors choose to focus on DES, MCS and ABS.

2.5.2 *SM Applications in the Study of MUHSs*

The literature reviewed in the previous section has a broad scope covering the applications of various SM methods in healthcare. In this section, relevant literature is reviewed with a specific focus: the applications of SM in the study of MUHSs. Simulation models of MUHSs are of strategic importance, however, the number of this type of simulation model is very limited (Jun et al., 1999; Gunal & Pidd, 2009).

One very early DES study on a whole hospital was reported by Fetter & Thompson (1965). The objective of this work is to give a decision support tool to hospital administrators to predict the consequences of design changes and alternative policies with a special interest in maternity processes. Three independent models are created for the hospital subsystems: 1) maternity suite; 2) a surgical pavilion; and 3) an outpatient clinic. The maternity model is used to analyze patient load and bed occupancy. The surgical pavilion model is to support experiments with surgical schedules. Priorities are given to unscheduled surgeries, which are generated according to a probability distribution. The outpatient clinic model handles the schedule of doctors and generates detailed reports on a variety of measures including patient waiting times, and physician idle times.

Hancock & Walter (1984) report a DES study which attempts to smooth the daily patient loads of 19 hospital departments by varying the admission days of urgent inpatient and outpatient loads. Given the variation in average load for each of the departments, the authors conclude that no one single policy can provide a stable workload for all departments, since each department has its own unique patient arrival patterns and treatment requirements.

Lowery (1992) develops a DES model to determine critical bed requirements by including all the units in a hospital's critical care areas i.e. operating rooms, recovery units, intensive care units and intermediate care units. Her literature review shows that most models previously reported do not fully consider the interrelationships between different hospital units and few models have been validated using actual hospital performance data. Focusing on these deficiencies, she demonstrates improvements in her methodologies over previous models.

Royston et al. (1999) report several applications of SD modeling to problems in the UK's National Health System (NHS). One of the applications described in detail in the paper uses SD to develop a better understanding of the interactions between the emergency care system and the social care system including residential care, community care, and primary health care. The main benefit of the model is its use as a learning tool, but it does show that changes in resources such as beds or staffing have less impact than changes in behaviours affecting referral patterns, length of stay, or inter-sectoral flows. In this application, several modeling workshops were organized, and the participants found that the solution to a problem in one sector of the system may often lie in another sector of the system.

Lane et al. (2000) develop a SD model to explore the relationships between waiting times in the emergency department (ED) and hospital bed closure. The result of the study shows that immediate impact of bed shortages is not seen in the ED, but is evident first in cancelled elective admissions, so that using ED waiting times to measure the effect of bed shortages is misleading. In fact, the model shows that it does not make sense to look at

any single measure in isolation as changes in one area can have unforeseen effects in other areas, thus a whole system approach is required.

Brailsford et al. (2004) develop a SD model to study patient flows within the Nottingham Health Authority's emergency care system. The key finding of the model is that "*the system is currently operating dangerously close to capacity*." However, relatively small changes in one part of the system have significant impact elsewhere. For example, it is much better to prevent admissions through the use of a community diagnostic facility than it is to reduce length of stay and discharge patients early.

Leonard et al. (2006) develop a combined SD and ABS model to simulate the interactions between the acute and aged care systems in Australia. The modeling approach is unique in its attempt to combine the high-level SD approach with the low-level ABS approach. The SD model is intended to capture the big picture structure of the "*forest*" while the agents act as the "*trees*" and flow through the model based on individualized behaviours. This model can be used to evaluate policies to address the effects of the aging population in Australia. Specifically, the authors propose policies to more quickly move patients from acute-care hospital beds to more appropriate long-term care resources.

In Lane & Husemann (2008), the authors discuss a "*hybrid approach*" which they apply to modeling acute patient flows. The model depicts a map of a general acute care hospital in a qualitative manner only – due to time constraints, the authors do not quantify the model in any way. Rather, they use it to generate discussions surrounding patient flow and to aid in determining steps that could be taken to improve patient flow. It demonstrates

that patient flow through A&E is affected by factors such as ward capacity and staff availability.

2.5.3 *Major Conclusions*

After examining the existing literature we can summarize the major conclusions as follows:

1) DES, SD and ABS are the most commonly used SM methods to model healthcare as a system. MCS is not good at modeling complex and large, multi-level decision healthcare systems but it is rather appropriate to make comparisons between specific health interventions (Barton et al., 2004; Katsaliaki & Mustafee, 2010).

2) The majority of the HSM studies use DES, while applications of SD and ABS are relatively less common. There are two possible reasons to explain why DES is the most commonly preferred SM method in healthcare. The first one is due to the lengthy history of DES applications in healthcare. In comparison, it is only in the past several years that ABS began to be applied in healthcare. The second one is related to the special modeling needs of healthcare systems (Davies & Davies, 1994/1995): the need for an individual patient focus, the importance of resource constraints, the primacy of clinical decision process, the power of animation and visualisation to communicate with the users, and more realistic representation without restrictive mathematical assumptions. In comparison with DES, the SD method is not well-suited to detailed modeling and copes rather badly with stochastic variation (Brailsford & Hilton, 2000).

3) Most of the SM studies in healthcare are unit-specific and there is a lack of simulation models of healthcare systems consisting of multiple units (Jun et al., 1999; Gunal & Pidd, 2009; Mielczarek & Uzialko-Mydlikowska, 2012).

4) The implementation of the results of the HSM studies is very rare. According to Wilson (1981), a possible reason is that simulation models cannot be developed quickly enough to meet the urgent needs of the decision-makers.

5) Most of the healthcare simulation models are developed by academics for research and/or student projects, rather than consultants or management engineers doing routine analysis on the operations of healthcare systems (Seila & Brailsford, 2009). This helps to explain why the implementation of HSM results is rare.

Chapter 3 Nearly Decomposable-Complex Adaptive Systems (ND-CASs)

This chapter presents a conceptual model, ND-CASs, to represent and describe the operations of MUHSs. The ND-CASs model integrates the key concepts in the theories of CASs and ND systems and is developed from a theoretical perspective. However, in this research, the ND-CASs model is not intended to be regarded as a new theory of complex systems, rather, it is more appropriate for it to be treated as a conceptual tool helping smooth and speed up the SM-based decision-making process for managing operations of MUHSs.

In the following section, the definition of a ND-CAS is introduced, and its main characteristics are discussed in Section 3.2. Section 3.3 discusses the applicability of using the ND-CASs model to represent and describe MUHSs. Its potential benefits for supporting SM-based decision-making in MUHSs are identified and described in Section 3.4.

3.1 Definition of ND-CASs

ND-CASs are special cases of complex systems. They are defined in terms of system elements, inter-element interactions, system structure, and system evolution.

System Elements

The elements in a ND-CAS are represented as adaptive agents, a concept adopted from the CASs theory. The agents are not dividable and have to function as single entities. The agents are intelligent and can learn from each other and from the environment, and

through learning, they can adapt to local and global changes. Agents are diverse and have different information about the system, but none of them understands the system as a whole. Each agent has its own agenda and priorities but has the nature to adapt to the needs of other agents when possible.

Inter-element Interactions

In a ND-CAS, each agent is connected to some other agents. Information is exchanged via inter-agent connections, creating patterns of interactions. Information exchanged can be generic or specific. Generic information is the information about all the agents or a subset of agents in the ND-CAS. Specific information is the information only about the pair of connected agents. Inter-agent interactions are typically associated with the presence of feedback mechanisms in the system, through which agents learn from and adapt to each other. Agents are diverse, and so are the inter-agent interactions. An agent may only interact with a certain number of agents and has no interactions with other agents. It can interact with some agents with higher frequency and intensity than with other agents.

System Structure

The diversity of inter-agent interactions leads to the creation and maintenance of hierarchical structures at the whole system level such that the system can be divided into hierarchical layers of subsystems. The existence of hierarchical structures can be conceptual only and the term *subsystem* is used to represent a real or virtual entity existing within the intermediate layers of a ND-CAS. A subsystem consists of lower layer subsystems or agents and is also part of either a higher layer subsystem or of the ND-

CAS. The hierarchical structures of the system have the ND property: interactions between agents belonging to same subsystems are of higher frequency and intensity than interactions between agents belonging to different subsystems.

The inter-agent interactions are multidimensional. A ND-CAS can simultaneously exhibit multiple hierarchical structures viewed along different dimensions. But along a particular dimension, a ND-CAS only exhibits one hierarchical structure, and this structure is relatively stable; it does change over time but at a much slower pace when compared with the changes of interactions between subsystems and agents at the lower layers.

System Evolution

A ND-CAS evolves over time via a *co-evolutionary* process. This *co-evolutionary* process can be witnessed at three levels: the bottom, the top and the intermediate level. At the bottom level, agents coevolve with each other, in other words, they transform and are transformed by each other via inter-agent interactions. At the top level, the ND-CAS as a whole coevolves with other ND-CASs and with the environment. At the intermediate level, agents, inter-agent interactions and the system structure coevolve together. Changes in agents can affect the inter-agent interactions thus affecting the system structure, and in the other direction, changes in system structure can also affect the agents and the inter-agent interactions.

3.2 Characteristics of ND-CASs

In the definition of ND-CASs, a "*pick and choose*" approach is used to adopt concepts from the theories of CASs and ND systems. This is to ensure that the ND-CASs model can provide a more holistic representation and description of certain types of complex systems i.e. MUHSs than the theories of CASs and ND systems. In the following sections, ND-CASs are compared with CASs and ND systems separately to explain their differences.

3.2.1 *ND-CASs vs. CASs*

In the definition of ND-CASs, the following two concepts are adopted from the CASs theory to describe the dynamics and structure of complex systems: 1) adaptive agents; and 2) inter-agent interactions. According to the CASs theory, system elements in a CAS are represented as adaptive agents who interact with each other and with the environment. The inter-agent interactions can lead to self-organization resulting in the emergent structure at the whole system level. However, in the definition of ND-CASs, the concept of self-organization is dropped. Instead, a *co-evolutionary* process is adopted to explain the creation and maintenance of the hierarchical structures at the whole system level. The *co-evolutionary* process is an enhanced version of the evolutionary process used in the ND systems theory. Both self-organization and the evolutionary process focus on the origination of the system structures meaning how the system structures can emerge from within the system. Compared with them, the *co-evolutionary* process focuses on both the origination and the evolution of the system structures. A ND-CAS can start with a system structure by design, and this system structure can affect the inter-agent interactions,

which, in turn, can affect the system structure. This process will proceed until a stable hierarchical structure is formed at the whole system level.

Both the ND-CASs model and the CASs theory can be used to explain the dynamics of complex systems and both recognize the importance of studying complex systems as a whole. However, the use of the ND-CASs model provides an additional solution to managing complexity in the study of complex systems in that with the ND property, the study of a ND-CAS can be simplified because it can be subdivided into separate studies of its subsystems, combined with an aggregated study of the interactions between the subsystems.

3.2.2 *ND-CASs vs. ND Systems*

In the definition of ND-CASs, the concept of the ND property is adopted from the ND systems theory. The benefits of the ND property have been explained in the previous subsection.

The major difference between a ND-CAS and a ND system is the use of adaptive agents in a ND-CAS to represent the system elements. In the ND systems theory, the system elements have no intelligence and inter-element interactions are assumed as natural existence with different frequencies and intensities. In addition, the ND systems theory pays limited attention to the changes of elements, their interactions and the system structure (Agre, 2003). The ND systems theory recognizes the existence of external changes, which can disturb the balance of the system or equilibrium, but its discussion of how the system will react to these external changes stays at a very high-level: the ND

system will move quickly back to equilibrium. In the ND-CASs model, the use of the adaptive agents provides a much richer explanation on the impacts of changes and how the system will react.

Changes in a ND-CAS can be external or internal. External changes come from outside of the system and include addition, deletion or modification of agents and the rules for inter-agent interactions. Internal changes come from inside the system and include the changes of agents and/or their behaviours as they experiment and gain experience. Both external and internal changes can influence the system structure.

The impacts of external and internal changes on the system structure can be analyzed from two perspectives. The first one is related to the property of a ND-CAS' structure. Inter-agent interactions will always exhibit different frequencies and intensities. This is a fact independent of external or internal changes. Thus the system structure of a ND-CAS shares the ND property though the actual arrangements of the agents in the system structure can change. The second one is about the moment-to-moment changes of a ND-CAS' structure. For external changes, it takes time for the system to settle down as agents need to relearn and readjust themselves, and the result can be a quite different system structure. For internal changes, the impact on the system structure can be minimal. The frequency and/or intensity of inter-agent interactions can change due to the internal changes. However, the frequency and/or intensity change of inter-agent interactions will not affect the system structure unless the change is strong and persistent enough.

3.2.3 *Characteristics of ND-CASs*

Based on the discussions in the previous two subsections, the characteristics of ND-CASs can be summarized as follows.

1) ND-CASs are composed of adaptive agents, which are intelligent and interact with each other and with the environment.
2) Inter-agent interactions are diverse and differ in frequency and intensity.
3) At the whole system level, ND-CASs can exhibit multiple hierarchical structures along different dimensions. Viewed along a particular dimension, there is only one hierarchical structure which is relatively stable. Hierarchical structures have the ND property and can be designed or emerge from the inter-agent interactions.
4) The hierarchical structures of ND-CASs can be maintained or adjusted via a *co-evolutionary* process among the agents, inter-agent interactions and the hierarchical structures. Compared with the changes of interactions between subsystems and agents at the lower layers, the changes of the hierarchical structures happen at a much slower pace. External and internal changes in ND-CASs can change the behaviour of agents and the inter-agent interactions, but they will not affect the hierarchical structures unless the changes are significant and persistent enough to change the system structures of the ND-CASs.

3.3 MUHSs as ND-CASs

A review of the characteristics of ND-CASs as outlined above suggests that MUHSs can be represented as ND-CASs.

First, there are different units included in a MUHS. These units are operated independently and can be represented as adaptive agents. For example, an ED is operated with its own procedures and operational goals. The management team of the ED can make their own decisions on staff planning and scheduling, process improvement and supplies management, and coordinate with other units, e.g. lab services and the hospitalist service. They can also evaluate the impacts of other units on the ED and make changes in the ED accordingly to deliver better services. For instance, to minimize the impact on the ED performance of the long boarding time of admitted ED patients to access beds in inpatient units, a *fast track* can be created to reduce the waiting times of those ED patients who are unlikely to be admitted.

Second, the inter-unit interactions in a MUHS are diverse and of different frequency and intensity. The operations of MUHSs are patient-centric and the resources are organized to support the delivery of healthcare services to the patients. Units providing related services to the same group of patients tend to interact more frequently and intensively than units providing less or not related services. Related services can be upstream services, downstream services or supporting services. For example, an ED interacts more frequently and intensively with a hospital ward than with a family physician clinic because the hospital ward provides downstream services to the admitted patients in the ED, who will stay in the ED until inpatient beds in the hospital ward become available.

Third, as explained above, the units in a MUHS do not interact with the same frequency and intensity. Viewed from the whole system level, certain hierarchical structures exist in the MUHS. One obvious example is the existence of hierarchical structure based on

geographical areas, e.g. provincial healthcare systems, regional healthcare systems, municipal healthcare system. Other structures include lines of formal authority, processes of patient handling and transfer, procedures, protocols and informal communication channels. These structures facilitate and direct information exchange between units in the MUHS and play a vital role in shaping the inter-unit interactions in the MUHS.

Fourth, the hierarchical structures of the MUHS do not stay unchanged. Instead, they coevolve with the units and the inter-unit interactions. For example, the introduction of new services or technologies can change the way in which certain units are operated thus affecting the practices and processes of the whole MUHS. Though the changes inside or outside of the MUHS are consistent, the changes to the hierarchical structures of the MUHS happen at a much slower pace. For instance, an ED can adopt new processes to improve its operational efficiency but this change will have relatively little impact on the interactions between the ED and the hospital wards, thus having little impact on the hierarchical structures of the bigger healthcare system, of which the ED is a unit.

3.4 Potential Benefits of the ND-CASs Model for Supporting Decision-making in MUHSs

Representing MUHSs as ND-CASs offers many potential benefits for effectively supporting decision-making in MUHSs. The major ones are described as follows.

1) Many MUHSs can be so complex that it is impossible or impractical to study them as a whole. With the ND-CASs model, granularity can be added to the study of these MUHSs. They can be studied by looking at each unit, subsystem, inter-unit

interaction and inter-subsystem interaction in the context of the system as a whole. This will make the study of these MUHSs possible and manageable.

2) People tend to be biased by their operational domains. When dealing with decision problems concerning multiple units, it can be very challenging to have all the stakeholders involved develop the same level of understanding of the decision problem and work together to deal with this. The ND-CASs model can be used as a conceptual tool to explain how the MUHS works and how the decision problem can be possibly addressed.
3) The hierarchical structures of a MUHS can be used in the design of the architecture of the MUHS simulation model. This model architecture can be further used as guidelines for developing new unit simulation models or reusing existing ones as well as for integrating unit and subsystem simulation models.

Chapter 4 An Agile HSM Approach

In the previous chapter, ND-CAS is introduced as a conceptual model to represent and describe the operations of MUHSs, and its potential benefits for supporting decision-making in MUHSs are also explored. From the perspective of HSM, the major benefit ND-CAS can offer is to divide the task of developing complex MUHS simulation models into smaller tasks. Develop unit simulation models first and then integrate them to build subsystem simulation models, which are further integrated to build higher layer subsystems simulation models or the MUHS simulation model. In this chapter, a HSM approach is introduced to take advantage of this benefit for improving the rapidity and flexibility of SM applications in healthcare.

In the following section, three driving forces for achieving rapidity and flexibility in HSM are investigated. In Section 4.2, a SM lifecycle approach to decision-making in MUHSs is proposed. The AHSM approach is introduced in Section 4.3 and the value proposition of AHSM is provided in Section 4.4. In the last section, the scope and limitation of AHSM are discussed.

4.1 Driving Forces for Achieving Rapidity and Flexibility in HSM

In Chapter 1, we introduced the need for rapidity and flexibility in HSM in order to effectively support decision-making in MUHSs. Rapidity means how quickly a simulation model can be developed to address the decision problem. Flexibility means the reusability of existing simulation models to address new decision problems. Although rapidity and flexibility are required in any decision-making process, there are three driving forces in

healthcare making this requirement a necessity in SM-based decision-making. These three driving forces are: 1) the nature of decision problems in MUHSs; 2) the dynamics in the decision-making processes of MUHSs; and 3) the current practices of SM in healthcare. Each of the three is discussed in the following sections.

4.1.1 *The Nature of Decision Problems in MUHSs*

Given the complexity and dynamics in the operations of MUHSs, many decision problems in healthcare are not well defined, e.g. the problems related to healthcare service delivery or managing patient flows between different units (Eldabi et al., 2002). These problems contain interactions that are not well understood and are not easy to capture analytically, including the involvement of many stakeholders, who often have different interests and backgrounds. In these cases, understanding the decision problem is a problem by itself.

Decision problems in MUHSs consist of numerous interrelated factors, which can influence each other (Liew & Sundaram, 2009). These factors must be represented in as simple a manner as possible so that decision-makers can understand and collaboratively discuss the problem and the MUHS.

Decision problems in MUHSs do not exist in isolation and they are also interrelated. To effectively address a decision problem, decision-makers may need to recall an existing or previously addressed decision problem as reference (Fierbinteanu, 1999; Power & Sharda, 2007).

4.1.2 *Dynamics in the Decision-making Process of MUHSs*

Decision-making is an iterative process including several components, or decision-making functions, which are highly interrelated and interdependent. A typical decision-making process is illustrated in Figure 4-1 (Harrison, 1999) and can be described as follows:

1) **Setting Decision Objectives:** Decision objectives constitute both the foundation and the end for decision-making. Attainment of the decision objectives is the ultimate measure of decision success.
2) **Searching for Alternatives:** Alternatives result from a search, which is limited by time and money.

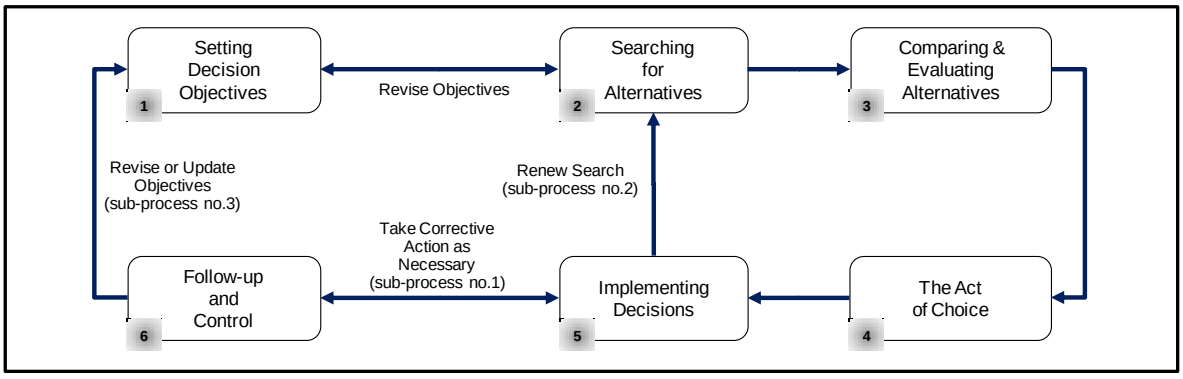


Figure 4-1 Process of Decision-making (Harrison, 1999)

3) **Comparing and Evaluating Alternatives:** There are usually three to five alternatives, and one of them is to do nothing. Alternatives are evaluated using criteria derived from the decision objectives. Evaluation should also include an anticipation of the likely outcome for each alternative and obstacles or difficulties at the time of implementation.

4) **The Act of Choice:** The choice of the final decision alternative is the culmination of the decision-making process, not all of it. The best choice is likely to result from the right approach and it should be the alternative most likely to result in the attainment of the decision objectives.
5) **Implementing Decisions:** Implementation is an important step in the decision-making process. A successful decision must be an implemented decision.
6) **Follow-up and Control:** Follow-up and control are essential to ensure that an implemented decision meets the objectives. Performance is measured by observing the implemented decision in relation to its standard derived from the decision objectives. Unacceptable variance from standard performance should trigger timely an appropriate corrective action (sub-process no.1). Corrective action may result in the implementation of another alternative (sub-process no. 2), which, if not successful, may result in a revision of the original decision objectives (sub-process no.3).

SM can be used to play different roles in the decision-making process. At the very early stage of the decision-making process, SM can be used to support all the decision-makers involved to learn as a group to identify and understand the decision problem and set the decision objectives. Subsequently, SM can be used in the search for different decision alternatives and the evaluation of these decision alternatives. SM can also be used to predict the likely outcomes of the decision and communicate the decision and its likely outcomes to people involved in the implementation of the decision. During the implementation of the decision, SM can again be used to support learning as a group to identify any unacceptable variance from standard performance and trigger timely design and implementation of appropriate corrective actions.

Depending on the role SM is going to play, the requirements for any simulation models to be developed may differ. To support learning as a group, simulation models need to be easily changeable so that suggestions and experiments can be readily implemented and interactively explored. To evaluate different decision alternatives, simulation models need to focus on the precision of the model outputs and have to be developed to include the operational details of the units and their interactions. These simulation models tend to have limited capability for changes.

In addition, as mentioned earlier, the decision-making process in MUHSs is highly dynamic, so SM cannot be used to play different roles in a predefined sequential order. Instead, SM must play variable roles in a rapidly changing manner.

4.1.3 *Current SM Practices in Healthcare*

As discussed in Section 2.5 of Chapter 2, most of the SM studies in healthcare are driven by specific decision problems and are usually one-time only. Also, there is a lack of simulation models of healthcare systems consisting of multiple units (Jun et al., 1999; Gunal & Pidd, 2009; Mielczarek & Uzialko-Mydlikowska, 2012). In their paper, Jun et al. (1999) conclude that "*despite the upward trend of healthcare simulation studies … there is still a void in the literature focusing on complex integrated systems*". Cochran & Bharti (2006) concur with Jun et al. (1999) that the literature on healthcare simulation modeling (HSM) includes a great deal of "*simulation models which vary enormously in complexity but are often unit-specific*".

Commenting on the unit-specific applications of HSM, Gunal & Pidd (2009) point out two potential problems. The first one is that "*there is no general sense of the literature moving forwards, because many papers tend to be reports of rather similar work on rather similar problems*." The second one is that unit-specific models usually assume rather tight boundaries around the system elements being modeled. The authors argue that "*there is a danger, when modeling a single unit, of ignoring what happens over the other side of the wall. That is, useful though they are, unit specific models and simulations may miss the big picture*."

4.2 A SM Lifecycle Approach to Decision-making in MUHSs

As a decision support tool, SM is intended to assist decision-makers in making decisions. There are several types of decision-makers involved in using SM and these decision-makers progress as they develop more confidence (Dreyfus & Dreyfus, 1986), from inexperienced/naïve decision-makers, to average decision-makers/analysts, to experienced decision-makers/modelers (Bhrammanee & Wuwongse, 2008; Iyer et al., 2005; Sprague, 1980). Each type of decision-maker has different needs. Some decision-makers may need more decisional and/or system usage guidance (Silver, 1991), while others may prefer to have minimal guidance. Some may want SM to take care of the entire decision-making process. Others may want to intervene to a greater extent in designing the entire decision-making process and the execution order to suit, or to a lesser extent in specifying a particular SM method. There are a variety of reasons as to why a human intervention is warranted and needed from the perspective of an experienced decision-maker (Beynon et al., 2002). However, it is interesting to note that a decision-maker's interventions may

have an adverse effect on decision model selection and ultimately the decision outcome (Jiang & Klein, 2000).

It is unreasonable and impractical to expect decision-makers to use a different SM method or a different simulation model for each decision. A decision-making process is not necessarily about concentrating on the decision itself, but should emphasise the ways in which decisions are made (Golub, 1997). Therefore, decision-makers should be able to choose an appropriate simulation model or SM method, and not be limited to only one simulation model or SM method (Draman et al., 2002; Geoffrion, 1987).

Simon (1977) proposes a very decision-oriented approach to the decision-making-process in terms of intelligence, design, and choice. Overall, Simon sought a science of administrative behaviour and decision making. He abandoned the notion that decision-makers seek to *optimize*, replacing it with the idea of *satisficing*: the idea that decision-makers search for solutions that are *good enough* in the perceived circumstances, rather than *optimal*. Simon claimed that the search is "*motivated by the existence of problems as indicated by gaps between performance and goals*". However, as Golub (1997) has suggested, decision-making is about the way in which we model the decision. Liew & Sundaram (2009) propose an approach to decision-making by integrating Simon's (1977) proposal with proposals from MS/OR modeling (Krishnan & Chari, 1993; Golub, 1997; Mathur & Solow, 1994). This approach attempts to support every phase and aspect of decisions and the SM lifecycle (Geoffrion, 1989b), and ensure that non-predetermined decision-making processes and interrelated decisions characteristics can be modeled. In this research, the approach proposed by Liew & Sundaram (2009) is modified to make it

more applicable to be used in the healthcare environment. The modified version is shown in Figure 4-2.

As summarized in Figure 4-2, the SM lifecycle is cyclical and iterative, and enables continuous adjustment and refinement especially in terms of storing and retrieving simulation models. Despite the fact that the SM lifecycle progresses step-by-step in a cycle, it can return to any earlier step (not just the previous one), and can skip some steps in the later iteration if it has already gone through that particular step earlier on. Only the major pattern of steps is illustrated in Figure 4-2.

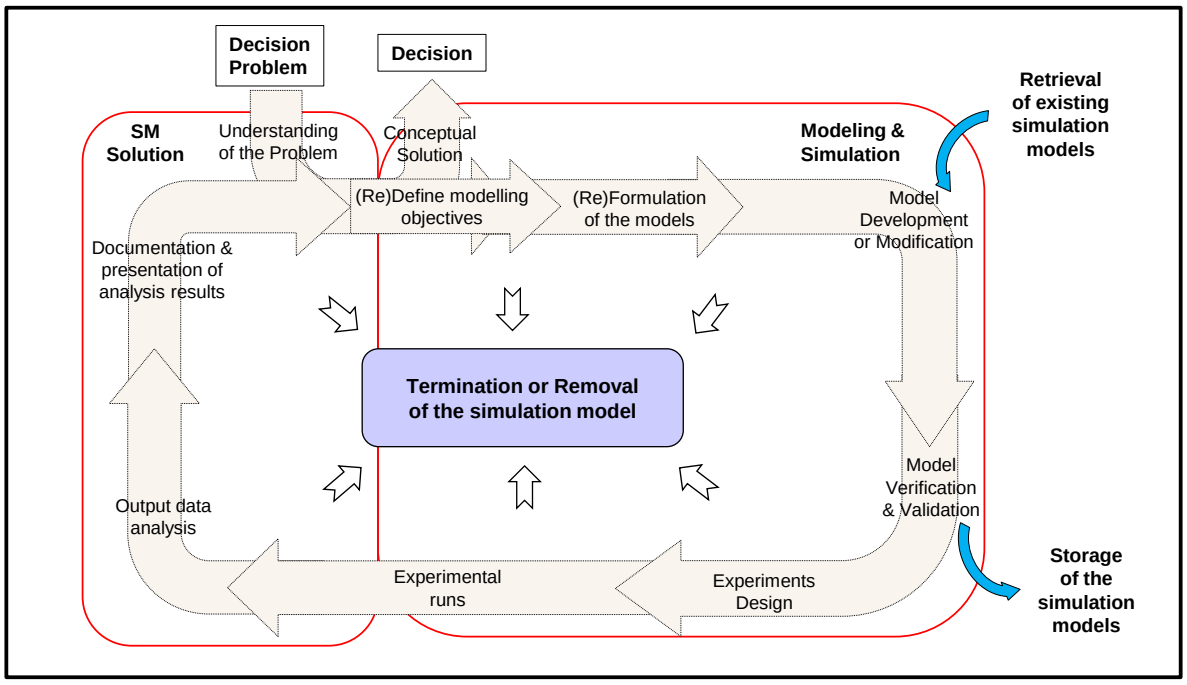


Figure 4-2 Cyclical SM Lifecycle in Supporting Decision-making in MUHSs

The SM lifecycle starts with understanding the decision problem and defining the modeling objectives. If a decision problem is clearly defined and understood, the modeling objective is to model that problem and seek for solutions. Otherwise, if the

decision problem is not well defined and understood, then the modeling objective is to model the operational environment of the decision problem for improved understanding. Once the modeling objective has been defined, a conceptual model is formulated to represent the decision problem or its operational environment. After the formulation of the conceptual model, relevant data are collected and a computerized simulation model is developed. It will be especially beneficial if this simulation model can be developed by reusing existing simulation models and this simulation model itself is storable and retrievable for later use and comparison. Once a simulation model is verified and validated, it can be run to generate different experimental results. A solution can be derived through analysing and investigating the experimental results. The derived solution is then reviewed and validated, and if it is considered unsatisfactory such information can be used to modify and reformulate the simulation model.

4.3 The AHSM Approach

The SM lifecycle approach provides a sound basis for the decision support and modeling framework so that SM can be used to play different roles in the decision-making process in a rapidly changing manner. In other words, in order to effectively support decision-making in MUHSs, SM has to be implemented with both rapidity and flexibility. AHSM is developed to achieve this objective.

AHSM is an SM approach focusing on the high level process for the use of SM in the decision-making process for managing the operations of MUHSs. The primary goal of

AHSM is to improve the rapidity and flexibility in the use of SM to effectively support decision-making in MUHSs.

AHSM consists of two interleaved stages. The first stage is "**Planning**", and its main objective is to identify the hierarchical structures of a MUHS and perform an architectural design of the overall SM studies. The major activities to be covered at this stage are the following ones shown in Figure 4-2: understanding of the (decision) problem, defining modeling objectives, (re)formulation of the (conceptual) model and the experimental design. The second stage is "**Executing**", and its main objective is to develop unit, subsystem and the MUHS simulation models. The major activities to be covered at this stage are the remaining activities shown in Figure 4-2.

Planning

One major activity at this stage is to identify the hierarchical structure of the MUHS to be studied. This activity has two objectives. The first one is to help decision-makers involved in the decision-making process develop a global view of the decision problem and the surrounding environment so that they can share a common understanding on the nature of the decision problem and reach a high-level agreement on how to deal with the decision problem. The second objective is to assist in the architectural design of the overall SM study.

For MUHSs with a large number of units, it can be a nontrivial task to identify their hierarchical structures and decompose them into hierarchical layers of subsystems. To deal with the decomposition problem in these cases, this research proposes a system

decomposition technique. The technical details of this approach will be provided in the next chapter.

Once the hierarchical structure has been identified and the MUHS has been decomposed into hierarchical layers of subsystems, an architectural design of the overall SM study is performed. Results from this architectural design include 1) the objectives of all the simulation models to be developed; 2) the interfaces between different simulation models i.e. the ranges of the input/output parameters; 3) data required for the development and validation of each simulation models; and 4)the experimental design of the MUHS simulation model.

Executing

Following the architectural design of the overall SM study, detailed SM studies are conducted with two steps. The first one is the development of simulation models for each unit. One principle used at this step is to reuse any existing simulation models as much as possible. The second step is to integrate the unit simulation models to form new simulation models to represent the subsystems and the MUHS in a sequential order: unit simulation models are integrated first to build subsystem simulation models. Then subsystem simulation models are further integrated to build higher-level subsystem simulation models or the MUHS simulation model. The reason for this sequential development of simulation models is to eliminate the need to develop a complex MUHS simulation model with a single step as this can be too complex to be done within a realistic time window.

In order to further improve the rapidity and flexibility, a simulation metamodel-based integration approach is proposed in AHSM. With this approach, each unit simulation model is run to generate experimental results, with the design of the experiments determined in the architectural design of the overall SM study. The experimental results are used to develop simulation metamodels represented using mathematical formulae, which can be implemented independently of the SM software package used to develop the original simulation model. These simulation metamodels are fast to run and can be used as building blocks in the development of the simulation models to represent the subsystems and the MUHS. Another advantage for using simulation metamodels is that it is easy to reuse the existing simulation models, which may have been developed using different SM methods or SM packages.

4.4 The Value Proposition of AHSM

The key values AHSM can bring to the SM-based decision-making process in MUHSs can be summarized as follows: 1) low complexity; 2) rapidity, 3) flexibility; and 4) improved engagement of stakeholders in the decision-making process in MUHSs.

Low Complexity: low complexity in conducting SM studies of MUHSs can be achieved by subdividing the MUHS into subsystems with smaller size, which can be modeled and studied separately and independently. Simulation metamodels, which are derived from the experimental results of the simulation models of the units in the MUHS, can be implemented without the requirement of a special software package. This can further

reduce the complexity of the development of the integrated simulation model to represent the MUHS to be studied.

Rapidity: The unit simulation models can be developed in parallel. Changes in one unit of the MUHS can be dealt with separately without affecting the development of the simulation models of other units. The use of simulation metamodels to represent the unit simulation models can improve the speed for the run of the integrated simulation models representing the MUHS.

Flexibility: the SM study of each unit can use separate SM methods and SM software packages. Depending on the requirements in the decision-making process, different simulation metamodels can be developed to represent the units for different purposes. This makes it possible to reuse the same unit simulation models for the development of different MUHS simulation models.

Improved Stakeholder Engagement: the independent developments of the unit simulation models encourage the involvement of the people with unit-specific knowledge in the decision-making process of the MUHS. The development of unit simulation metamodels and the integration of these simulation metamodels to represent subsystems and the MUHS also encourage people from different units to interact with each other. This will also encourage the people involved to share ownership of the final decision and participate more actively in the implementation of the decision.

4.5 AHSM Scope and Limitations

In order to fully realize the potential of AHSM, it is important to become aware of its scope and limitations.

1) AHSM is not an approach designed only for the HSM community, rather, it is an approach developed to facilitate the communication, coordination and interaction among healthcare modelers, decision-makers and other stakeholders in the decision-making process of MUHSs.
2) In order to realize its full potential, AHSM needs to be used in conjunction with many existing methods and techniques for SM studies of MUHSs. These methods and techniques include those for data collection and processing, model verification and validation, etc.
3) AHSM is not a cookbook approach for developing simulation models of MUHSs; the application of AHSM does not need to follow the "all-or-none" principle. The ultimate goal of AHSM is to improve the rapidity and flexibility in the study of MUHSs, and if the use of one or more of the techniques supporting AHSM cannot serve this ultimate goal, their use should be re-evaluated.
4) AHSM is not a silver bullet that will solve all decision problems in MUHSs. AHSM is developed to target those OM problems which involve multiple units and multiple decision-makers. An in-depth understanding of the inter-unit interactions is crucial for making the right and informed decisions for these problems.

5) AHSM is not developed for specific SM methods, since AHSM should work with any SM method including those not discussed explicitly in this research (e.g. MCS).
6) AHSM is primarily developed for SM applications in MUHSs to deal with OM decision problems which are decomposable. Namely, the overall decision problems can be decomposed into smaller problems which can be dealt with by analyzing the operations of each unit and inter-unit/subsystem interactions. If the decision problems are not decomposable, then the applicability of the AHSM approach can be compromised.

Chapter 5 AHSM Supporting Techniques/Approaches

This chapter presents three types of technique or approach to support the implementation of AHSM. The first is a system decomposition technique whose objective is to identify the hierarchical structure of the MUHS to be studied and decompose the MUHS into hierarchical layers of subsystems. The second is a NN-based simulation metamodeling technique whose objective is to develop NN-based simulation metamodels to represent the simulation models of units or the subsystems of the MUHS. The third is a model integration approach for integrating the simulation models or simulation metamodels of the units to represent the subsystems and the overall MUHS. Details on each technique or approach are provided in the following sections.

5.1 System Decomposition by Factor Analysis

5.1.1 *Overview*

As discussed in Chapter 3, a MUHS can be represented as a ND-CAS and decomposed into hierarchical layers of subsystems with interactions between units belonging to the same subsystems much more frequent and intense than those between units belonging to different subsystems. This is very beneficial for modeling a MUHS with large number of units, where the development of a single simulation model to represent the interactions between all the units in the MUHS can be impractical. With the ND-CASs model, either new simulation models are developed or previously developed simulation models are reused to represent each of the individual units. These unit simulation models are integrated together for the development of subsystem simulation models, which can be

further integrated for building simulation models of higher-level subsystems or the overall MUHS.

However, the work in Chapter 3 only shows that a MUHS can be represented as ND-CAS, it does not provide a solution to finding the exact hierarchical structure of a MUHS. When the number of units in a MUHS is small, e.g. 3 or less, the hierarchical structure of that MUHS can be very obvious, but with a large number of units, it can be a nontrivial task to identify the hierarchical structure. To deal with the decomposition problem for these MUHSs, a heuristic approach is proposed in this section.

The proposed approach was first used by Bauer et al. (1985/1991) and Matthes (1988) for decomposing a DES model into smaller segments to ease model implementation. Based on graph theory (West, 2001), the authors developed a network representation (or graph) to represent the relationships between the components of the DES model. This graph, in turn, is systematically decomposed into sub-graphs by means of factor analytic methods, or principal component analysis (PCA) in particular. Components of the DES model belonging to the same sub-graphs are first interconnected together to form segments. These segments are further interconnected to form the whole DES model. From the implementation's viewpoint, this approach is indeed an aggregation approach as it starts from the components at the lowest level and aggregates them together to form segments, each including multiple components. However, from a retrospective viewpoint, it can be regarded as a decomposition approach as the final structure has middle layers while the original structure has no middle layers.

Compared with those in Bauer et al. (1985/1991) and Matthes (1988), the way in which the decomposition is applied in this research differs in two ways. First, it is used to decompose a MUHS instead the structure of a DES model. Second, in Bauer et al. (1985/1991) and Matthes (1988), the decomposition is only performed one step and the overall resulting structure only has three layers: components, segments and the whole. But in this research, the decomposition will be repeated until no additional layer in the structure can be identified. The MUHS will start with only one layer: the MUHS as a whole consisting of multiple units. After the first stage of decomposition, there are two possibilities: 1) no new middle layer is identified and the decomposition process stops, or 2) A middle layer is identified and the MUHS becomes a three-layer system: the MUHS at the top layer (layer 3), the units at the bottom layer (layer 1) and the subsystems at the middle layer. For the second possibility, a further stage of decomposition is performed considering the subsystems as the basic elements of the MUHS. This process proceeds until no new layers can be identified.

5.1.2 *The Process of the Decomposition Approach*

The decomposition process includes the following steps (Bauer et al., 1985/1991; Matthes, 1988):

1) Construct the graph of the system to be decomposed.
2) Construct the matrices representing the graph.
3) Calculate the pseudo-covariance (C) and pseudo-correlation (R) matrices.
4) Perform PCA on the R matrix and determine the number of sub graphs.
5) Determine the structure of each sub-graph.

6) Name each sub-graph.

In the following paragraphs, each step is explained in detail using a relatively simple graph as an example.

Step 1: Construct the graph of the system to be decomposed

Based on the textbook definition by West (2001), the following is a definition of a graph:

Definition: A graph G is a triple consisting of a vertex set V(G), an edge set E(G), and a relation that associates with each edge and two vertices (not necessarily distinct) which are called its endpoints (West, 2001: Definition 1.1.2, p_2).

Additionally, a graph is drawn by setting each vertex at a point and signifying each edge by a line connecting the locations of its endpoints. The values assigned to the edges are the amount of information (e.g., number of inputs, attributes, etc.) being passed between vertices. A graph may be undirected, which means that the flow between the vertices can go both ways, or directed, where there is a distinct flow between vertices and it only goes one way. The arrow heads on the edges show the direction of information flow, specifically for directed graphs.

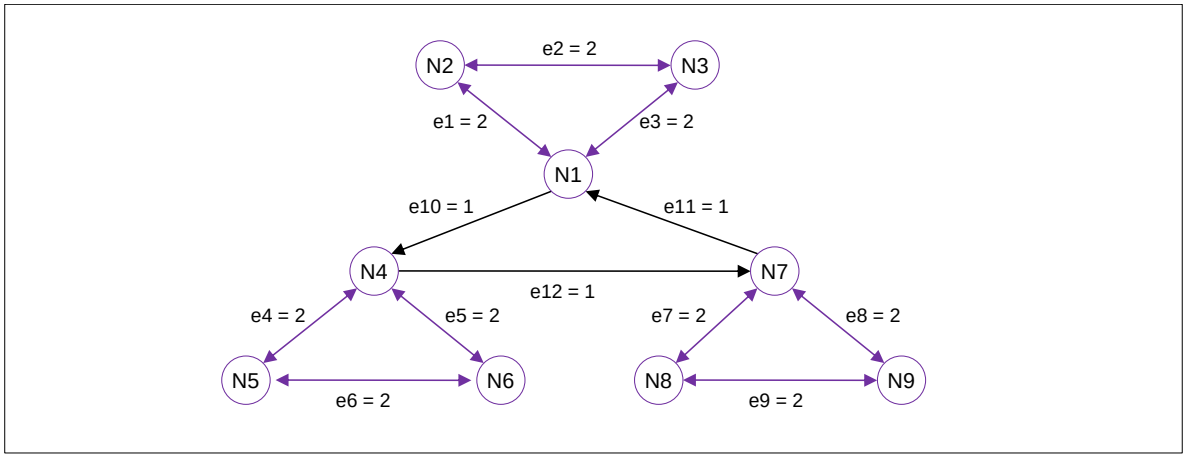


Figure 5-1 A Simple Graph (Bauer et al. 1985)

Figure 5-1 depicts a simple graph used in Bauer et al. (1985). This graph can be represented as follows:

$$G = \{V(G), E(G), R(G)\} \qquad \text{(Eq. 5-1)}$$

Where

V(G) = {N1, N2, N3, N4, N5, N6, N7, N8, N9}, is the vertex set,

E(G) = {e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12}, is the edge set, the value of each edge represents the directions of the information flow: 1 for unidirectional; 2 for bidirectional.

R(G) = {$e_{N1\leftrightarrow N2}$, $e_{N1\leftrightarrow N3}$, $e_{N2\leftrightarrow N3}$, $e_{N1\leftrightarrow N4}$, $e_{N4\leftrightarrow N7}$, $e_{N7\leftrightarrow N1}$, $e_{N4\leftrightarrow N5}$, $e_{N4\leftrightarrow N6}$, $e_{N5\leftrightarrow N6}$, $e_{N7\leftrightarrow N8}$, $e_{N7\leftrightarrow N9}$, $e_{N8\leftrightarrow N9}$}, is the set of relations.

Step 2: Construct the matrices representing the graph

A matrix is typically a clear and efficient manner of representing a graph for use in analysis. A graph can be represented in terms of its adjacency and/or incidence matrix. Fundamentally, the incidence matrix captures the vertex-to-edge relationships while the adjacency matrix captures the vertex-to-vertex relationships. Below is the formal definition:

Definition: Let G be a loopless (multiple edges are allowed but loops are not) graph with vertex set V(G) = { $\upsilon_1,\ldots, \upsilon_n$} and edge set E(G) = {$e_1,..., e_m$}. The **adjacency matrix** of G, written A(G), is the n-by-n matrix in which entry $a_{i,j}$ is the number of edges in G with endpoints { υ_i, υ_j}. The **incidence matrix** M(G) is the n-by-m matrix in which entry $m_{i,j}$ is 1 if it is an endpoint of e_j and otherwise is 0 [West, 2001: Definition 1.1.7, p5-6].

The adjacency matrix A(G) = ($a_{i,j}$) is therefore given by

$$a_{i,j} = \begin{cases} 1 & (v_i, v_j) \in E(G) \\ 0 & Otherwise \end{cases} \qquad \text{(Eq.5-2)}$$

and the incidence matrix M(G) = ($m_{i,j}$) of a graph is given by

$$m_{i,j} = \begin{cases} 1 & if\ v_i\ \ is\ an\ endpoint\ of\ e_j \\ 0 & Otherwise \end{cases} \qquad \text{(Eq.5-3)}$$

The adjacency and edge incidence matrices of the simple graph in Figure 5-1 are shown in Figure 5-2.

A(G)=

	N1	N2	N3	N4	N5	N6	N7	N8	N9
N1	0	1	1	1	0	0	0	0	0
N2	1	0	1	0	0	0	0	0	0
N3	1	1	0	0	0	0	0	0	0
N4	0	0	0	0	1	1	1	0	0
N5	0	0	0	1	0	1	0	0	0
N6	0	0	0	1	1	0	0	0	0
N7	1	0	0	0	0	0	0	1	1
N8	0	0	0	0	0	0	1	0	1
N9	0	0	0	0	0	0	1	1	0

M(G)=

	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12
N1	1	0	1	0	0	0	0	0	0	1	1	0
N2	1	1	0	0	0	0	0	0	0	0	0	0
N3	0	1	1	0	0	0	0	0	0	0	0	0
N4	0	0	0	1	0	1	0	0	0	1	0	1
N5	0	0	0	1	0	1	0	0	0	0	0	0
N6	0	0	0	0	1	1	0	0	0	0	0	0
N7	0	0	0	0	0	0	1	1	0	0	1	1
N8	0	0	0	0	0	0	1	0	1	0	0	0
N9	0	0	0	0	0	0	0	1	1	0	0	0

Figure 5-2 Adjacency and Incidence Matrices for the Simple Graph

In addition, there is also an edge weighting matrix (W), which is an m-by-m matrix representing the edge values. The W matrix of the simple graph is shown in Figure 5-3.

W(G) =

	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12
e1	2	0	0	0	0	0	0	0	0	0	0	0
e2	0	2	0	0	0	0	0	0	0	0	0	0
e3	0	0	2	0	0	0	0	0	0	0	0	0
e4	0	0	0	2	0	0	0	0	0	0	0	0
e5	0	0	0	0	2	0	0	0	0	0	0	0
e6	0	0	0	0	0	2	0	0	0	0	0	0
e7	0	0	0	0	0	0	2	0	0	0	0	0
e8	0	0	0	0	0	0	0	2	0	0	0	0
e9	0	0	0	0	0	0	0	0	2	0	0	0
e10	0	0	0	0	0	0	0	0	0	1	0	0
e11	0	0	0	0	0	0	0	0	0	0	1	0
e12	0	0	0	0	0	0	0	0	0	0	0	1

Figure 5-3 Edge Weighting Matrix for the Simple Graph

Step 3: Calculate the pseudo-covariance (C) and pseudo-correlation (R) matrices

Next in the process is to calculate the pseudo-covariance matrix $C = MWM^T$, where M is the edge incidence matrix, W is the edge weighting matrix and M^T is the transpose of M. The command used in **MATLAB® 2006** to calculate C matrix is: >> *C = M*W*M'*. The calculated C matrix for the simple graph is shown in Figure 5-4.

$$C(G) = \begin{bmatrix} 6 & 2 & 2 & 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 6 & 2 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 4 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 6 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 4 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 \end{bmatrix}$$

Figure 5-4 Pseudo-covariance (C) Matrix for the Simple Graph

Since C in Figure 5-4 is symmetric (i.e., a matrix is equal to its transpose) and positive semi-definite (i.e., all of its principal minors ≥ 0), it can be converted to a pseudo-correlation matrix $R = D^T CD$, where D is the inverse square root of the diagonal matrix of C and D^T is the transpose of D. The D calculation is shown in Figure 5-5 with the corresponding simple graph D matrix.

$$D = \begin{bmatrix} \ddots & & & \\ & \frac{1}{\sqrt{c_{ii}}} & & 0 \\ & & \ddots & \\ 0 & & & \ddots \end{bmatrix} \qquad D(G) = \begin{bmatrix} 0.408 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.408 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.408 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}$$

Figure 5-5 D Matrix for the Simple Graph

The commands used in **MATLAB® 2006** for the calculation of R matrix are as follows:

1) >> DINV = sqrt(diag(C))
2) >>R = C./(DINV*DINV')

The derived R matrix is shown in Figure 5-6.

$$R(G) = \begin{bmatrix} 1 & 0.408 & 0.408 & 0.167 & 0 & 0 & 0.167 & 0 & 0 \\ 0.408 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.408 & 0.5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.167 & 0 & 0 & 1 & 0.408 & 0.408 & 0.167 & 0 & 0 \\ 0 & 0 & 0 & 0.408 & 1 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.408 & 0.5 & 1 & 0 & 0 & 0 \\ 0.167 & 0 & 0 & 0.167 & 0 & 0 & 1 & 0.408 & 0.408 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.408 & 1 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.408 & 0.5 & 1 \end{bmatrix}$$

Figure 5-6 Pseudo-correlation (R) Matrix for the Simple Graph

Step 4: Perform PCA on the R matrix and determine the number of sub-graphs

Now that the association matrices have been derived, the next step is to assess the dimensionality of the graph. The dimensionality assessment is basically determining how many sub-graphs are present in the graph using PCA.

PCA (also known as the Karhunen-Loève transformation) is a data reduction technique used to reduce a complex dataset to a lower dimension to expose the sometimes hidden, underlying structure of a high dimensional data (e.g., data with several features). The basic premise is to transform the original set of variables (features) into some smaller set of linear combinations that explain the most variance in the original dataset. The general PCA algorithm can be found in Dillon et al. (1984). The command used in **MATLAB® 2006** to perform PCA on R matrix is: *>> [LPR, ER, PR] = pcacov(R)* and

- LPR is a p-by-p matrix, with each column containing coefficients for one principal component.
- ER is a vector containing the principal component variances, which are the eigenvalues of R.

- PR is a vector containing the percentage of the total variance explained by each principal component.

Table 5-1 depicts the results of performing PCA on the pseudo-correlation matrix R using **MATLAB® 2006**.

Table 5-1 Extracted Factors for the Simple Graph

Factor	Eigenvalue	% of Variance	Cumulative % of Variance
1	2.000	22.222	22.222
2	1.833	20.370	42.593
3	1.833	20.370	62.963
4	0.833	9.259	72.222
5	0.500	5.556	77.778
6	0.500	5.556	83.334
7	0.500	5.556	88.889
8	0.500	5.556	94.445
9	0.500	5.556	100.000

Once the eigenvalues are extracted from the R matrix, several decision criteria can be used to determine the number of factors (principal components) to be retained (Hair et al., 2006: p119):

- ***Latent Root Criterion*:** Factors with eigenvalues greater than 1.0.
- ***Percentage of Variance Criterion*:** Enough factors to meet a specified percentage of variance explained.
- ***Scree Test Criterion*:** Factors shown by the Scree test to have substantial amounts of common variance (i.e., factors before the inflection point).
- ***A Priori Criterion*:** The researcher already knows how many factors to extract before undertaking the factor analysis.

As an important note, multiple decision criteria are often required to decide upon the number of factors to be retained. Based on the recommendations in Hair et al. (2006), the following procedure is used to determine the number of factors to be retained:

Step a: use the ***Latent Root Criterion*** to decide the initial number of factors to be retained.

Step b: calculate the total percentage of variance explained by the initial number of factors.

Step c: verify whether the ***Percentage of Variance Criterion*** can be met. No absolute threshold has been adopted for all applications. In natural science, the threshold is usually 95%, but in social science, where information is less precise, 60% is a commonly used number (Hair et al., 2006: page 120). The 60% threshold will be used in this research as the inter-unit and inter-subsystem interactions in MUHSs can measured along different dimensions and we usually only consider some of the most important ones related to the decision problem.

Step d: If the ***Percentage of Variance Criterion*** is met, use the factors selected in step 1. Otherwise, keep adding more factors until the ***Percentage of Variance Criterion*** is met. In cases where there are many equally important factors with the same eigenvalue, the procedure will stop at the first factor where the ***Percentage of Variance Criterion*** is met.

Based on the extracted eigenvalues in Table 5-1, three factors are retained because the eigenvalues of all the three factors are greater than 1.0, and the cumulative percent of variance, which can be explained by the three factors is 62.96% (>60%).

Step 5: Determine the structure of each sub-graph

After determining the number of factors to be retained, the next step is to interpret the factors (sub-graphs) or determine the structure of each sub-graph. First, a PCA will be performed on the C matrix using the following **MATLAB® 2006** command:

>> [LPC] = pcacov(C)

Like LPR in the PCA of R matrix, LPC is the loading matrix. Each column of LPC contains coefficients for one factor. Each coefficient (or called factor loading) illustrates the relation of the corresponding node to that factor: the greater the absolute value of the coefficient, the greater the linear correlation of the node to the factor. Squared factor loadings indicate what percentage of the variance in an original variable is explained by a factor. A node is said to *load* on a factor when the node has the highest loading on that factor. After the factor loading has been obtained, the initial results will be evaluated and a number of judgments are made in viewing and refining these results. The initial factor solutions achieve the objective of data reduction, but in most instances they will not provide information that offers the most adequate interpretation of the factors or features under examination. So an orthogonal factor rotation is often needed to simplify the factor structure. The objective of the orthogonal factor rotation is to simplify the rows and columns of the factor matrix to facilitate interpretation. In a factor matrix, columns represent factors, with each row corresponding to a node's loading across the factors. By simplifying the rows, it means making as many values in each row as close to zero as possible (i.e., maximizing a variable's loading on a single factor). By simplifying the columns, it means making as many values in each column as close to zero as possible

(i.e., making the number of high loadings as few as possible). The most popular rotation method is the **VARIMAX** rotation (Hair et al., 2006: p126), which centers on simplifying the columns of the factor matrix by maximizing the variance of a column of the factor matrix of the retained factors (Kaiser, 1958). The logic is that interpretation is easiest when the node-factor correlations are (1) close to either +1 or -1, thus indicating a clear positive or negative association between the nodes and the factor; or (2) close to 0, indicating a clear lack of association. This structure is fundamentally simple.

Table 5-2 shows the unrotated factor loadings for the simple graph.

Table 5-2 Unrotated Factor Loadings – C for the Simple Graph

Node	Factor 1	Factor 2	Factor 3
N1	-0.471	0.000	0.525
N2	-0.236	0.000	0.442
N3	-0.236	0.000	0.442
N4	-0.471	-0.454	-0.262
N5	-0.236	-0.383	-0.221
N6	-0.236	-0.383	-0.221
N7	-0.471	0.454	-0.262
N8	-0.236	0.383	-0.221
N9	-0.236	0.383	-0.221

Not much can be interpreted from the unrotated factor loading. Table 5-3 shows the rotated factor loadings using the ***VARIMAX*** rotation method for the simple graph. The command for performing **VARIMAX** rotation (with three retained factors) to the factor loading matrix, LPC, is show as follows:

>> Bvar= rotatefactors(LPC(:,1:3),'Method','Varimax')

Bvar is the rotated factor matrix shown in Table 5-3.

Table 5-3 *VARIMAX* Rotated Factor Matrix – C for the Simple Graph

Node	Factor 1	Factor 2	Factor 3
N1	-0.058	-0.058	0.701
N2	0.045	0.045	0.497
N3	0.045	0.045	0.497
N4	-0.058	-0.701	0.058
N5	0.045	-0.497	-0.045
N6	0.045	-0.497	-0.045
N7	-0.701	-0.058	0.058
N8	-0.497	0.045	-0.045
N9	-0.497	0.045	-0.045

After examining Tables 5-3, we see nodes 1, 2 and 3 load on Factor 3, nodes 4, 5 and 6 load on Factor 2, and nodes 7, 8 and 9 load on Factor 1. So we can say that there are three (3) principal graphs here (3 retained factors) and each is composed of the three nodes.

Step 6: Name each sub-graph

Typically, part of the PCA assessment is to evaluate how the nodes in the factors are related and thus producing a "*naming*" convention for the grouping, also known as the interpretability criterion. In this case, we only need to assess which nodes belong to what factor, since the graph is fairly generic and no specific names are assigned to the nodes.

5.2 NN-based Simulation Metamodeling

In AHSM, simulation models are recommended to be developed and maintained at the unit level so that they can be kept current to reflect changes in the real systems. It also improves the flexibility and reusability of simulation models to address different decision problems. However, there are two major issues in the implementation of the AHSM

approach. The first one is that many unit simulation models may exist already and they were developed using various SM methods and/or SM software packages, simply due to historical reasons. The second one is that the unit simulation models can be very complex already with a lot of details included. Although certain SM methods and SM software packages can be recommended for future SM studies to improve the interoperability and standardization, it can be costly and impractical to update the existing simulation models to use new SM methods or SM software packages. In order to make it easy to reuse these existing simulation models, simulation metamodeling is proposed in this section as a useful solution.

5.2.1 *Overview of Simulation Metamodeling*

Simulation metamodeling was first proposed by Blanning (1975) to obtain useful sensitivity information with a significant reduction in the computation time. Kleijnen (1975) suggested some statistical tools to lead metamodels into common use in stochastic simulation, and later (Kleijnen, 1979) proposed a definition of a simulation metamodel, which has been commonly used since then. This definition can be summarized as follows:

Let X_j denote a factor j influencing the outputs of the real-world system (j =1, 2, ... , s), and let Y denote the response of the system. The relationship between the response variable Y and the inputs X_j of the system can be represented as:

$$Y = f_1 (X_1, X_2 , \dots , X_s) \qquad \text{(Eq. 5-4)}$$

A simulation model is then an abstraction of the real system, in which we consider only a selected subset of the input variables $\{X_j,\ j = 1,2, \dots r\}$ where r is significantly smaller

than the unknown s. The response of the simulation Y' is then defined as a function f_2 of this subset and a vector of random numbers v representing the effect of the excluded inputs:

$$Y' = f_2 (X_1, X_2, \dots , X_r, v) \qquad \text{(Eq. 5-5)}$$

A metamodel is now a further abstraction, in which we select a subset of the simulation input variables $\{X_j, j = 1,2, \dots m, m <= r\}$ and describe the system as

$$Y'' = f_3 (X_1, X_2, \dots , X_m) + \varepsilon \qquad \text{(Eq. 5-6)}$$

where ε denotes a fitting error, which has an expected value of zero. Figure 5-7 illustrates these levels of abstraction.

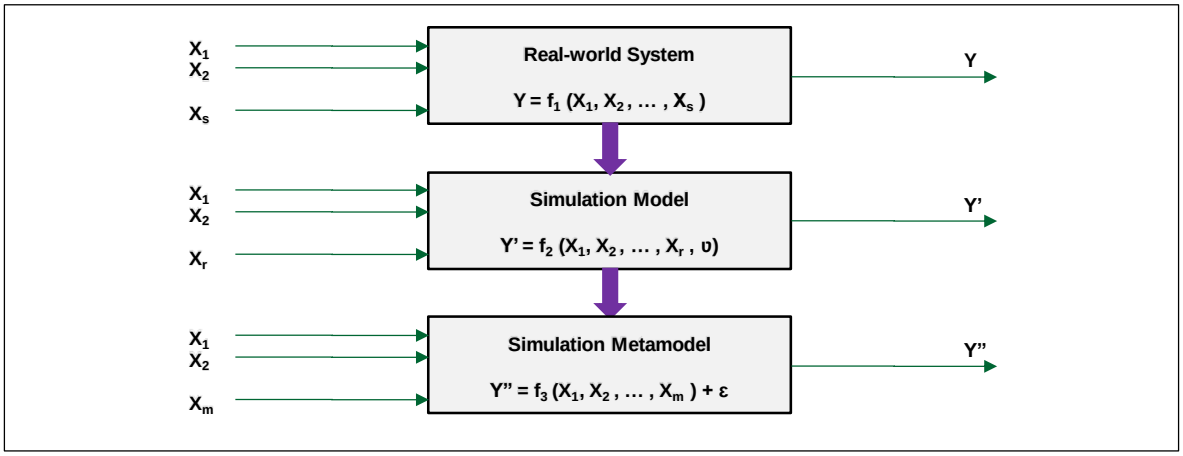


Figure 5-7 The Concept of a Simulation Metamodel

Simulation metamodeling has been used extensively as a post-analysis tool in manufacturing, agriculture, chemical engineering and other areas (Yu & Popplewell, 1994). The advantages of metamodeling in post-simulation analysis are explored by Friedman & Pressman (1988). Among these are model simplification, enhanced

exploration and interpretation of the model, generalization to other models of the same type, sensitivity analysis, optimization, answering of inverse questions and better understanding of the studied system and the inter-relationships between system variables.

A simulation metamodel provides a statistical summarization of simulation results, allowing some extrapolation from the simulated range of systems conditions, therefore its potential use in many other applications areas has been explored by many authors. Starr (1991) reports a use of hybrid metamodels and simulation techniques for decision support. Similarly, Santos & Santos (2007) also recommend the reuse of a simulation metamodel as a building block in a large simulation model. In military applications, simulation metamodeling has been used as a technique to aggregate hierarchically lower-level models into the next higher-level model (Rodriguez, 2008).

The most popular methods for constructing simulation metamodels have been based on response surface methodologies (RSM) using parametric regression model approximations. These methods are the topic of entire texts (Myers, 1976; Box & Draper, 1987; Khuri & Cornc, 1987) and have been used successfully for performing sensitivity analysis within a limited region of the parameter space and determining solutions satisfying various constraints (Box & Draper, 1987). However, they have not been used to perform global approximation of a simulation model because of their inability to provide a globally accurate fit to the response function (Myers et al., 1989).

One alternative approach is to use an artificial neural network (ANN), usually called a "*neural network*" (NN). Pierreval & Huntsinger (1992) explore the potential use of NNs as a metamodeling tool for discrete-event and continuous simulation models. They

describe NN as a suitable metamodeling technique to represent a discrete-event job simulation model. In his early study on NNs, Hurrion (1992) develops a NN metamodel from a stochastic simulation of a coal depot. The NN is trained to predict the mean and the 99% confidence interval of the time the depot remains open. Badiru & Seiger (1993) use a NN as a simulation metamodel for economic analysis of risky projects. They develop a NN model to predict the potential returns from investment projects with stochastic parameters, e.g. initial investment, the rate of return, and the investment period. The metamodel developed can analyze the performance of potential future projects without re-running the time-consuming simulation. The authors carried out a similar study in 1998 (Badiru & Seiger, 1998). Experimenting with a simple investment project, they show that good predictive capability can be achieved by simplifying a simulation with a NN. Kilmer & Smith (1993) and Kilmer et al. (1994) experiment with an inventory problem, and also suggest NNs as metamodels for DES analysis. NNs perform better than first-order and second-order linear regression models when compared with these two traditional approaches (Kilmer & Smith, 1993). Hurrion (1998) experiments with three example problems: an M/M/s queue, a coal depot system and a flow shop system, and shows that NN metamodels produce more accurate responses than regression metamodels.

Compared with RSM, NN has many unique characteristics but its use as a simulation metamodeling technique is a relatively recent development (Kilmer et al., 1994; Badiru & Seiger, 1998). Given the focus of this research, it is not practical to provide a comprehensive comparison between NN and RSM approaches for all possible applications related to the implementation of the AHSM approach. Instead, this research chooses to only use NN as the simulation metamodeling technique. However, this choice

does not intend to recommend NN as the only technique or the best technique for developing simulation metamodels. Instead, the purpose is just to promote NN as another good candidate technique of simulation metamodeling and is a viable simulation metamodeling technique to support the implementation of the AHSM approach.

5.2.2 *NN Overview*

NN is a mathematical system that mimics the way in which the brain works. It consists of several highly interconnected computational elements, known as nodes, neurons, or perceptrons. There are various NN models and the most popular and commonly employed NN is the multilayered feed-forward NN (MFNN) (Kate & Jatinder, 2000). Other names for the MFNN are FANN (Feed-forward Artificial Neural Network), MLP (Multilayer perceptron) network, and back propagation (due to its learning algorithms) network. NNs are based on the generalized delta algorithm, which provides a method of updating the weights so that the errors are minimized (Bishop, 1995). An example of three-layer NN model architectures is shown in Figure 5-8.

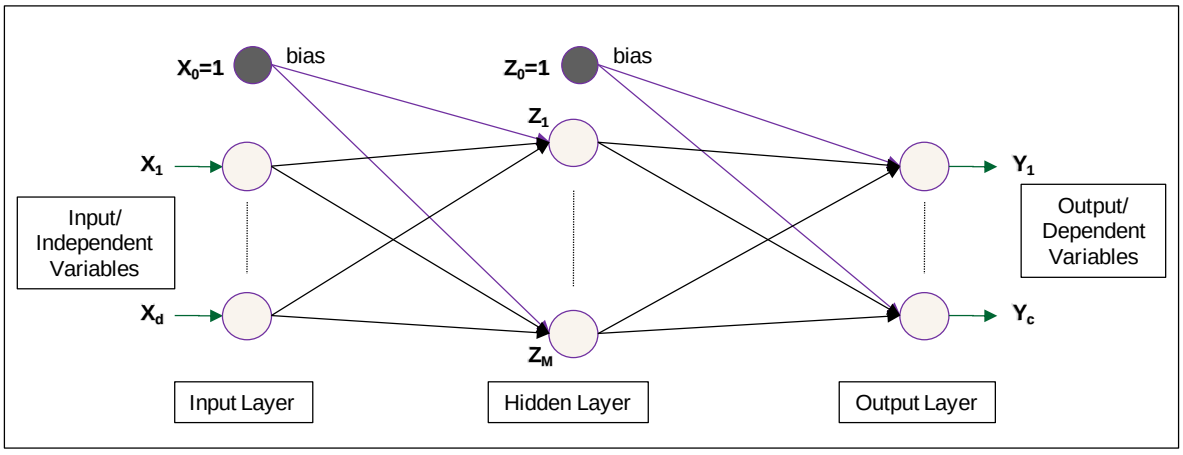


Figure 5-8 Three-layer NN Model Architecture

As shown in Figure 5-8, the first layer is called the input layer, and the last layer is known as the output layer. The number of nodes in the input layer corresponds to the number of independent variables, and the number of nodes in the output layer corresponds to the number of dependent variables. The layers of nodes in between the input and output layers are called hidden layers. Biases are weight values associated with individual nodes. Biases are determined by the iterative flow of training data through the network (i.e., biases are established during a training phase in which the network learns how to identify particular classes by their typical input data characteristics).

Each node in a NN performs the following tasks:

1) Signals are received from other nodes (X_0, X_1, …,X_d).
2) Signals are multiplied by their corresponding weights (W_0X_0, W_1X_1, …,W_dX_d).
3) Weighted signals are summed (Sum = $W_0X_0 + W_1X_1 +…+ W_dX_d$).
4) The calculated sum is transformed by an activation function [F(Sum)].
5) The transformed sum is sent to other nodes (repeat 1-4 above).

Some of the common activation functions are as follows (Sarle, 1994; Stern, 1996; Haykin, 1999):

- Linear or identify: $F(x) = x$
- Hyperbolic tangent: $F(x) = \tanh(x)$
- Threshold: $F(x) = 0$ if $x<0$, 1 otherwise
- Gaussian: $F(x) = x^{(-x2/2)}$
- Logistic: $F(x)=(1+e^{-x})^{-1}$

Typical activation functions used in the hidden layers of NNs are the hyperbolic tangent function or the logistic functions. This is typically followed by an output of linear activation nodes for prediction problems (Haykin, 1999). NNs often have one or more hidden layers, but generally, three-layer NNs (given adequate number of M nodes in the single hidden layer) are universal approximators (Hornik et al., 1989).

Compared with conventional statistical procedures, NNs usually use more parameters, and are thus more susceptible to overtraining. The overtraining phenomenon is observed when the mean squared error of the NN continues to increase while the network performance is still improving in learning the training set. This is highly undesirable as it signifies that the NN cannot recognize unknown patterns and its generalization ability is hampered (Tzafestaset et al., 1996). A commonly used method in dealing with the overtraining phenomenon is to divide the data into a training set and a test set. The training set is used to fit the NN model, and the test set is used to evaluate the model's performance. Alam et al. (2004) recommend the use of the RMSE (Root Mean Square Deviation) as the measure of performance for the NN due to its ability to incorporate a measure of both the variance and the square of the bias of the prediction errors.

The NN is trained by repeatedly giving it examples from its training set. A training set consists of an input vector paired with a corresponding output vector. Each example from the training set is offered to the NN and its output is calculated. The error between the calculated result and the real result is used to modify the matrix of weights between each layer in the network. The error adjustment is termed back propagation, which is a conceptually simple iterative gradient descent algorithm (Rumelhart et al., 1986).

In training NNs, the following three training parameters are required: learning rate, momentum, and training tolerance. Learning rate limits or multiplies the extent of weight changes in any given iteration. A high learning rate that reacts very quickly to input changes tends to make the network unstable. However, if the learning rate is lower than optimum, the NN will take a substantially longer time to learn. A consensus among researchers is that adaptive learning rates are preferred as they can stabilize and accelerate convergence to a desired solution (Looney, 1996). The momentum factor determines the proportion of the last weight change that is added to the new weight change. Low momentum often causes oscillation of weights and renders the network unstable, and learning is never completed. High momentum corresponds to a lack of flexibility and adaptability on the part of the NN. In general, the momentum factor should be less than one (unity) to stabilize back-propagation (Yu & Chen, 1997). The training tolerance factor specifies the margin of error allowable when NN outputs are compared to real outputs. A training tolerance of zero indicates that the NN outputs must exactly match the real outputs. A training tolerance close to zero can adversely affect the ability of the model to generalize, as a high degree of accuracy in the model is desired relative to training data. However, a high training tolerance factor is also not recommended, as it will result in inaccurate results because the specified accuracy is low (Kuo & Reitsch, 1995). In summary, balance must be achieved in specifying the training parameters for a NN. The training parameters are application specific and are usually determined by trial and error.

5.2.3 *The Steps in NN-based Simulation Metamodeling*

Figure 5-9 shows a set of steps in the development of NN-based simulation metamodels.

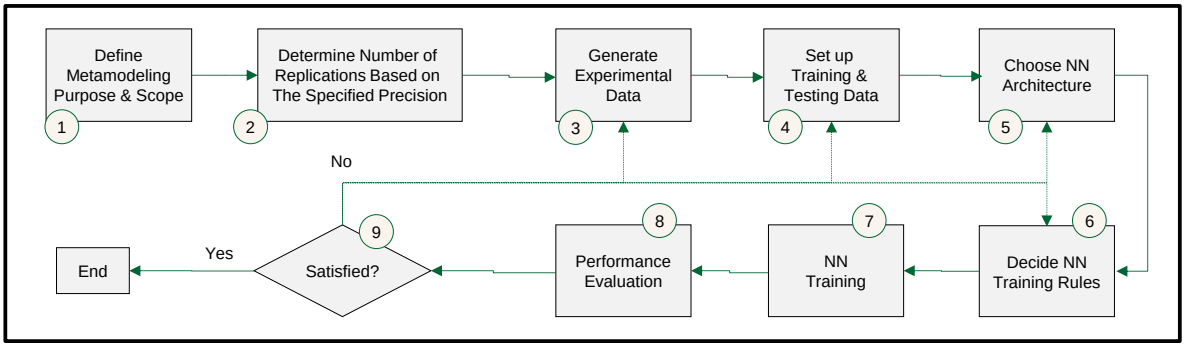


Figure 5-9 Steps in NN-based Simulation Metamodeling

In this research, the Multilayer Perceptron (MLP) module of IBM SPSS® Neural Networks 19 is used for the training and testing of the NN models and will be used as reference in the following discussions.

Step 1: Determine metamodeling purpose and scope: The purpose and scope of all the subsystem simulation models and metamodels should be decided in the *Planning* step of the AHSM approach, which involved decomposition of the MUHS into hierarchical layers of subsystems and the architectural design of the overall SM study of the MUHS. The purpose of the simulation metamodels of the units in the MUHS is to capture the behaviours of the units in response to inputs from other units or outside of the MUHS. The scope of the metamodel includes the input-output relationships to be included in the metamodel, the range of the metamodel inputs and the specified precision of the metamodel outputs compared with those of the original simulation models. The range of the metamodel inputs is determined with the consideration of the following two main factors: 1) the purpose of the simulation metamodel, and 2) the limitation of the original simulation model. An initial range of the metamodel inputs can be determined based on the purpose of the simulation metamodel. Then the original simulation model is run for

inputs at the lower and upper limits of the range to verify that it can properly represent the operations of the real systems within this range for the purpose of analysis. If the verification result is positive, the range of inputs is used for the development of the simulation metamodel. Otherwise, the original simulation model is modified to serve the purpose of the simulation metamodel.

Step 2: Determine number of replications based on specified precision: Obtain the proper number of replications for the simulation model based on the specified precision for the outputs of the simulation model. The objective to be achieved at this step is to find the minimum number of replications so that the half length of the 95% confidence interval (CI) of the experimental results is less than the specified precision. There are two main factors to be considered when deciding the specified precision, which include: 1) the purpose of the simulation metamodel, and 2) the time required to run the simulation model.

Step 3: Generate experimental data: Once the proper number of replications for the simulation model has been decided, the simulation model is run to generate experimental results. In the simulation metamodeling process, this step may be repeated several times. At the beginning, only a small number of data points can be chosen for each input. For example, if there are three inputs of the simulation metamodel, we can only use three data points for each input: one data point at each end (lower and upper limit) and one in the middle. Then we need to run the original simulation model 3x3x3 = 27 times. If at later steps, we find that the simulation metamodel cannot accurately represent the original

simulation model with the performance target, which will be discussed later, we can come back to this step and generate additional experimental results.

Step 4: Set up training and testing data: Partition the experimental data into two groups: one group is used to train the NN models and the other group is used to test the trained NN models. Both the training and testing data points should be evenly distributed across the input ranges of the simulation metamodel to maintain performance consistency of the simulation metamodel at any data point. As a rule of thumb, 20% to 30% of the data points are used as testing data. One important note to make here is that the test data set is designed to validate the trained NN models on interpolation rather than extrapolation. In other words, the trained NN models are used as a special "look-up" table to provide results of the given inputs within the domain of the experimental results generated from running the original simulation models.

Step 5: Choose NN architectures: Choose the architectures of the NN models, which includes the following parameters: 1) number of hidden layers; 2) numbers of nodes on each hidden layer; 3) activation function on hidden layers; 4) activation function on output layer; and 5) method for rescaling input and output variables. In order to make it easy to train NN models and implement them once trained, the following default NN architectures in **IBM SPSS® Neural Networks 19** are used in this research:

- **Number of hidden layers:** 1
- **Number of nodes on each hidden layer:** same as or dividable by the number of inputs

- **Activation function on hidden layer:** hyperbolic tangent
- **Activation function on output layer:** identity
- **Method for rescaling input and output variables:** standardized.

Changes to the default NN architectures will only be considered when the trained NN models cannot meet the performance targets.

Step 6: Decide NN training rules: Decide on the rules used to train the NN model. The following default training settings in **IBM SPSS® Neural Networks 19** are used in this research:

- **Type of training:**
  - Batch mode: Update the synaptic weights only after passing all training data records; that is, batch training uses information from all records in the training dataset.
- **Optimization algorithm:**
  - Gradient descent
- **Training options:**
  - Initial learning rate: 0.4
  - Momentum: 0.5
  - Interval center: 0
  - Interval offset: +/-0.5
- **Stopping rules:**
  - Maximum steps without a decrease in error: 1000

- Maximum training time: 15 minutes
- Maximum training epochs: 1,000
- Minimum relative change in training error: 1.0e-4
- Minimum relative change in training error ratio: 1.0e-3

Step 7: NN training: Train the NN model using the training data sets and following the training rules described in Step 6.

Step 8: Performance evaluation: Evaluate the performance of the NN model using the test data sets.

Step 9: Satisfied? If the performance of the trained NN-based simulation metamodels meets expectations, then the simulation metamodeling process will end. Otherwise, choose different training rule following the steps below:

a) Run the MLP module three times and get three trained NN models. Compare the *sum of squares error* and *relative error* for the training datasets of the three NN models in the model summaries.

b) If the difference is over 10% between the three NN models, reduce the initial learning rate by 0.05 and repeat step a). Otherwise, go to next step.

c) Compare the *sum of squares error* and *relative error* between the training datasets and the testing datasets for each NN model, if the difference is over 10%, increase momentum by 0.05 and go back to step a). Otherwise, go to next step.

d) Rerun the MLP module once and get a new trained NN model. If the performance of the NN model meets expectations, then stop the simulation metamodeling process. Otherwise, go to next step.

e) Check the *stopping rule used* in the model summary, if the stopping rule used is maximum training time of 60 minutes, then end the process of choosing different training rules.

f) Make the change to relax that rule. For example, if the *stopping rule used* in the model summary is *1000 consecutive steps* with no decrease in error, then change the maximum steps without a decrease in error from 1000 to 10,000 and go back to step d).

After we have tried different training rules, we need to go back to the previous steps to choose different NN architectures, different ways of partitioning the data into training/test groups, or collect more experimental data if the performances of the trained NN models are still not satisfactory.

5.3 A Flexible Approach to Integrating Simulation Models

In AHSM, a simulation model is developed to represent each unit in the MUHS to be studied. These unit simulation models are used to analyze the behaviour of the units in response to the inputs from other units or from outside of the MUHS. Unit simulation models are integrated to represent the subsystems and the subsystem simulation models are further integrated to represent the higher-level subsystems or the overall MUHS.

Conceptually, model integration sounds like a very simple and straightforward task; however, there are many challenges at the implementation level. First, unit simulation models may have been developed using different SM methods and SM software packages. These can be very complex models already and typically have not been developed to be integrated with other simulation models. Second, simply integrating the programming code of unit simulation models does not automatically create a simulation model to represent the combined operations of all the units. Much verification and validation work is required in the model integration process, introducing extra complexity into the development process of the MUHS simulation model. Third, the integrated simulation model representing the MUHS must not become too complex, otherwise it may defeat the purpose of the AHSM approach especially in terms of the flexibility necessary to address future decision problems.

To address these challenges, different model integration approaches are reviewed in the following section, after which a FMI (flexible model integration) approach is proposed to address the different model integration requirements in the application of AHSM.

5.3.1 *Model Integration*

Model integration means different things to different people due to their different perspectives on what constitutes a model. Historically, there have been two main viewpoints on model and model integration, one from computer and information science and the other from operations research/management science, though the two have been gradually blending (Tang, 2002).

From the viewpoint of computer and information science, a model is defined as "*a schematic description of a system, theory, or phenomenon that accounts for its known or inferred properties and may be used for further study of its characteristics*" (Addanki et al., 1990). The main focus of model integration research from the computer and information science perspective is on addressing the technical issues that are relevant to model integration, i.e. *conflict resolution* and *structural validity* of the integrated models (Krishnan & Chari, 1993). Conflict resolution is a major part of model integration as the components of the models being integrated may have *naming conflicts* (an attribute named *price* means the price of the material in a manufacturing model while the same attribute means the sale price in a demand estimation model), *granularity conflicts* (a variable in a revenue forecasting model forecasts revenues by year while the budgeting model uses revenue forecasts by the quarter), or *dimensional/units of measurement conflicts* (one of the variables which measures sales volume does it in dollars, while the other measures it in terms of the number of units sold). Bradley & Clemence (1987) propose a type calculus for modeling languages with the specific objective of identifying conflicts. Krishnan et al. (1993) describe the use of types and type inferences in a language called ASCEND (Piela et al., 1992) to facilitate conflict resolution. Bhargava et al. (1991) introduce a concept called *quiddity* which works in concert with dimensional information to detect conflicts in names, type, and dimension.

Given the variety of changes to model structure during integration, how can one be certain that the structure that emerges upon integration is a valid model? The most promising development in answer to this question is the work of Geoffrion (1987) on structured modeling. Structured modeling is a formally specified notational framework for modeling

that was developed to address a variety of model development problems. A key feature of structured modeling is the ability to test if a given structure is a valid structured model. Another useful feature of structured modeling is its explicit representation of the inter-relationships between model elements. This representation enables the impact of a change to a model object to be identified by tracing the dependencies between model elements. Geoffrion (1989a) describes a detailed manual procedure for integrating models specified in the SML language (Geoffrion, 1992a/b).

From the viewpoint of OR/MS, a model is defined as "*a computational representation having a specific real-world interpretation and providing the user with inferences about that world*" (Elam & Lee, 1986). Model integration is used as a practical approach to thinking about "*modeling in the large*" to support model-based decision-making. Model integration research from this perspective focuses on operation and manipulation of models once they have been developed (Krishnan & Chari, 1993). Specifically, model integration has attracted significant considerations in the study and management of complex systems such as cancer biology (Patel & Nagl, 2011), and environmental decision support systems (Rizzoli et al., 1998; Argent, 2004). In complex systems research, the need for model integration is vital to understanding the very systems under study. It usually involves a large modeling community, each individual group within that community having their respective subsystem specialty, modeling methodology preferences and technological limitations, and each contributing to the literature a mass of knowledge in the form of models and simulations. These fields establish their own version of a modeling lifecycle, however, too often the majority of efforts are shelved or

forgotten, to the detriment of the research community: whole-system simulations fall short due to the lack of an integrating principle between modeling efforts.

5.3.2 *Model Integration Approaches*

There are three approaches to model integration: direct model integration (DMI); metamodel-based integration (MMI); and hybrid model integration (HMI).

The Direct Model Integration (DMI) Approach

Direct Model Integration (DMI) is an approach that directly combines the individual models (Meckesheimer, 2001). The DMI approach can be further distinguished between "*deep*" integration and "*functional*" integration (Geoffrion, 1989a). Others have made this same distinction using different terms; for example, Dolk & Kottemann (1993) call it *definitional* vs. *procedura*l integration. *Deep* integration produces a single new model that combines two or more models. *Functional* integration, in distinction, does not yield a single new model. Instead, it leaves the individual models as they are and superimposes a computational agenda for coordinating calculations amongst them, typically directing certain models' outputs to other models' inputs while specifying the order of computations (portions of which may be left to automatic resolution at run-time). The computational agenda here, which of course must be defined formally, serves as the (only) definition of the functional integration. The HLA defined by IEEE is one example of the functional integration approach.

The Metamodel-based Integration (MMI) Approach

In the MMI approach, simulation metamodels are developed to represent each simulation model using the techniques discussed in Section 5.2. These simulation metamodels, instead of the original models, are integrated.

The Hybrid Model Integration (HMI) Approach

The HMI approach is a mixture of the DMI and MMI approaches. Using this approach, simulation metamodels are only developed to represent some of the simulation models, then these simulation metamodels, together with some of the original simulation models are integrated to construct the integrated simulation model. Multiple criteria can be used to decide whether the original simulation model or the simulation metamodel should be used to represent a particular unit or subsystem. These criteria include: 1) the chosen SM method and/or SM software package for the MUHS simulation model; 2) the complexity of the simulation model; and 3) the decision problem.

5.3.3 *Unit Simulation Model Integration in AHSM*

Any of the three model integration approaches can be applied in the integration of unit and subsystem simulation models, but each approach has certain advantages and disadvantages. On the positive side, the DMI approach is very straightforward and can retain the operational details of the subsystems which are captured in the original simulation models, however, it has several disadvantages. First, the interfaces of the subsystem simulation models have to be designed to a common standard. Once the objective of the system model changes, the interfaces of the subsystem simulation models

may need to change and this will require the redesign of the interfaces of the subsystem simulation models. Second, the DMI approach usually requires the same computer/operating system platform or the same SM method and software package. Third, if some subsystem simulation models are already computationally expensive to run, the system model can be computationally prohibitive to run. Fourth, the DMI approach may result in overbuilt system simulation models as not all the features of the subsystem simulation models are needed, but removing the extra features of the subsystem simulation models can be very costly and time-consuming.

The MMI approach has two advantages over the DMI approach: rapidity and flexibility. First, the simulation metamodels take much less time to run than the original simulation models, reducing the computational expense and allowing modelers to gain insight into the decision problems by evaluating and viewing decision alternatives much more quickly. This also encourages interactive sessions, in which modelers of different subsystem models can participate in the system level analysis. Second, with the MMI approach, a set of simulation metamodels can be generated and implemented. This will better manage the task of creating and managing different interface designs, because it reduces both the amount of custom coding as well as the complexity of the subsystem simulation models. This permits a greater variety of analysis to be implemented. However, the MMI approach also has some disadvantages. First, a simulation metamodel is required for each output variable, thus if there are many output variables in the subsystem simulation models, it will require many simulation metamodels to be developed. Second, the simulation metamodeling process will introduce errors for each individual metamodel which can accumulate and affect the accuracy of the system

simulation model. Third, simulation metamodels are only valid within the domain of experimental results generated from the run of the original simulation models. Finally, once the original simulation models change, the simulation metamodels will become invalid and new experimental results will have to be generated for the development of new simulation metamodels.

The HMI approach is a trade-off solution between the DMI approach and the MMI approach, only applicable and efficient when some subsystem simulation models are simple enough to be integrated directly and/or when they are developed using the same SM method and SM software package as those for the development of the integrated simulation models.

5.3.4 *A FMI Approach for AHSM*

In order to maximize the advantages of the three model integration approaches while mitigating their disadvantages, a Flexible Model Integration (FMI) approach is proposed to support the implementation of AHSM. FMI is a procedural approach which outlines a framework for the use of different model integration approaches in AHSM. It includes the following five major steps:

1) <u>Make a list of the unit, subsystem and MUHS simulation models to be developed.</u>

To address a particular decision problem concerning a MUHS, the MUHS can be decomposed into hierarchical layers of subsystems, and subsequently the interactions among subsystems can be studied to address that decision problem. For MUHSs with a large number of units, the decomposition can be done using the technique proposed in

Section 5.1. Following the decomposition of the MUHS, a simulation model needs to be developed of each unit, after which these unit simulation models need to be integrated to represent subsystems. The integrated subsystem simulation models need to be further integrated to represent higher-level subsystems or the overall MUHS. Let's assume a MUHS consisting of n units can be decomposed into an m-layer system, and we use K_i (for i=1 to m), to represent the number of units or subsystems at each layer. Then

K_1 = n, which is the total number of units at layer 1,

K_i, (for i=2 to m-1) = the number of subsystems at each layer between 2 and m-1, and

K_m = 1, the MUHS at the top layer (layer m).

This means that we will need to first develop n unit simulation models, then $\sum_{i=2}^{m-2} K_i$ integrated subsystem simulation models, and finally one integrated MUHS simulation model. Once the n unit simulation models have been developed, the development of the MUHS simulation model can be divided into m-1 steps, each step representing the development of subsystem simulation models or the overall MUHS simulation model by integrating a number of simulation models at the immediately lower layer. For example, the 1st step after the development of the K_1 unit simulation models is to develop the K_2 subsystem models at layer 2 of the hierarchical structure of the MUHS by integrating different unit simulation models. The development of the subsystem simulation models within a particular layer can be done in parallel, but their integration to create the subsystem simulation models at the layer above has to be done in a sequential order. That

means a subsystem simulation model at a certain layer cannot be done unless all the required simulation models at the lower layer have been developed.

2) Define the purpose and interface specification of each simulation model

The purpose of each simulation model should be determined by the decision problem and criteria used to decompose the MUHS. For example, if the decision problem concerns patient flow within the MUHS, the unit simulation models should focus on simulating the operations of the units and assessing how the patient flows between the units can be affected by changes happening in the same or different units. Similarly, a subsystem simulation model should focus on simulating the interactions amongst units or subsystems at the lower layer, which represent the internal operations of that subsystem, and assessing how the patient flow via this subsystem can be affected by changes happening in the same or different subsystems. The interface specification of each simulation model should be in alignment with the criteria used to decompose the MUHS. Using the same example for the patient flow analysis, the major interface parameters of the simulation models will be the arrival/exit rates and waiting times of patients.

3) Perform an inventory check to assess the reusability of each previously developed simulation model

After the purposes of the simulation models have been decided, an inventory check should be done for all the previously developed simulation models, with the goal being to use or modify previously developed simulation models provided that this will take less time and effort than developing new simulation models from scratch.

The previously developed simulation models can fall into one of the following two categories: 1) simulation models representing individual units; and 2) simulation models representing two or more units (i.e. subsystem or MUHS models). For a previously developed unit simulation model, consider the following:

- Use it if it can serve the purpose of the planned study,
- Modify it if it takes less time and effort than developing a new simulation model from scratch, or
- Drop it from consideration for the current decision problem.

For a previously developed multi-unit simulation model, consider the following:

- Use it represent the MUHS if it can serve the purpose of the planned study,
- Use it to represent a subsystem if it can serve the purpose of the planned study,
- Use it for the development and calibration of some unit simulation models if possible, or
- Drop it from consideration for the current decision problem.

For units with no previous simulation models developed, check what SM methods and software packages were used for the development of the existing simulation models of other units and try to use the same SM method or SM software package as long as there are no obvious disadvantages for using them.

4) <u>Decide the model integration approaches to be used for the development of the subsystem or MUHS simulation models</u>

Use the following principles for choosing the integration approach:

- **For the integration of unit simulation models**

  Use the DMI approach only when the unit simulation models are simple, and developed using the same SM method and SM software package, and do not require frequent updating. Otherwise, the MMI or HMI approaches should be used.

- **For the integration of subsystem simulation models**

  Use MMI approach only when the integrated simulation model becomes too complex. Otherwise, the DMI or HMI approaches should be used.

5) Develop or reuse unit simulation models for developing subsystem simulation models and the overall MUHS simulation model

After all the preparation work mentioned in the previous steps has been finished, the process for developing and integrating simulation models will start. Major tasks at this step include:

- Develop and validate new unit simulation models.
- Modify and validate previously developed unit simulation models.
- Develop and validate subsystem simulation models by integrating simulation models representing units or subsystems at the immediately lower layer.
- Develop and validate the MUHS simulation model by integrating simulation models representing subsystems at the immediately lower layer.

Chapter 6 CHR Healthcare System Decomposition

6.1 Introduction

This chapter presents an application example of the system decomposition technique introduced in Section 5.1 of Chapter 5, using the healthcare system within the Calgary Health Region (CHR) in Alberta, Canada as the target MUHS to be decomposed. In order to avoid unnecessary details while maintaining the representativeness of the application example, a simplified version of the CHR healthcare system studied in a research project funded by the HQCA (Health Quality Council of Alberta) in 2009 is used (Rohleder et al., 2009). The objective of that research project was to investigate patient flow problems within the CHR with a focus on the services affecting the operations of EDs within the CHR. The author was one of the researchers involved in this project. The CHR healthcare system studied in the HQCA research project consisted of the following eight (8) units:

- Primary Care,
- Urgent Care,
- Emergency Departments (ED),
- Outpatient Clinics and Consultants Services,
- Surgery & Recovery,
- Hospital Wards (HW),
- Patient waiting services for outpatient clinics and consultants services, and
- Patient waiting services for surgery and hospital wards.

The patient flows between the eight (8) units, which correspond to the CHR operations in 2009, are shown in Figure 6-1.

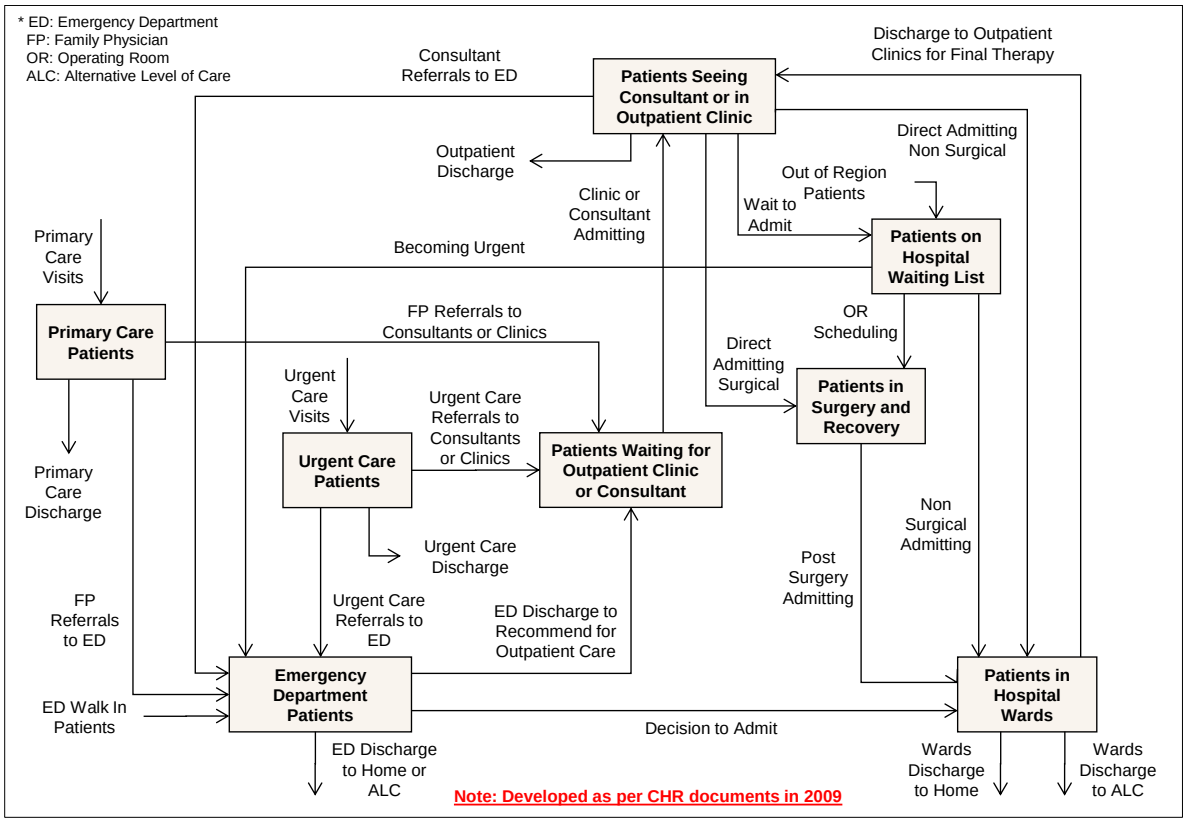


Figure 6-1 Patient Flow Diagram for the CHR Healthcare System

There are two main reasons why the CHR healthcare system studied in the HQCA research project was chosen in this application example. The first one is because the CHR healthcare system is a typical MUHS with many service units interconnected with each other. The proposed system decomposition technique can be used to make the study of the CHR healthcare system more manageable. The second reason is because the HQCA research project can be used as a practical example to demonstrate how the system

decomposition technique might be able to help make some modeling and research tasks easier from a retrospective point of view.

In the following sections, the system decomposition technique is applied in decomposing the CHR healthcare system into hierarchical layers of subsystems. Then the potential applications of the decomposition result in the HQCA project are explored.

6.2 CHR Healthcare System Decomposition

In this section, the steps described in Section 5.1 of Chapter 5 are followed in the decomposition of the CHR healthcare system.

Step 1: Construct the graph of the system to be decomposed

The first step when applying the system decomposition technique in this case is to construct a graph representation of the CHR healthcare system. This can be done via a direct translation from the patient flow diagram shown in Figure 6-1. The sources of the arriving patients and destinations of the discharged patients are not part of the CHR healthcare system, but they need to be included when constructing the graph representation of the CHR healthcare system. For the purpose of easy formatting, the following acronyms (Table 6-1), instead of the full names of the healthcare services or units, are used in the graph representation of the CHR healthcare system as shown in Figure 6-2.

Table 6-1 Acronyms used in the Graph Representation of the CHR Healthcare System

Acronym	Full Name or Explanation
S1-4	Sources of arriving patients
D1-5	Destinations of discharged patients
P	Primary Care
U	Urgent Care
ED	Emergency Departments
OC	Outpatient Clinics and Consultants Services
SR	Surgery & Recovery
HW	Hospital Wards
CWL	Patient Waiting Services for outpatient clinics and consultants services
HWL	Patient Waiting Services for surgery and hospital wards

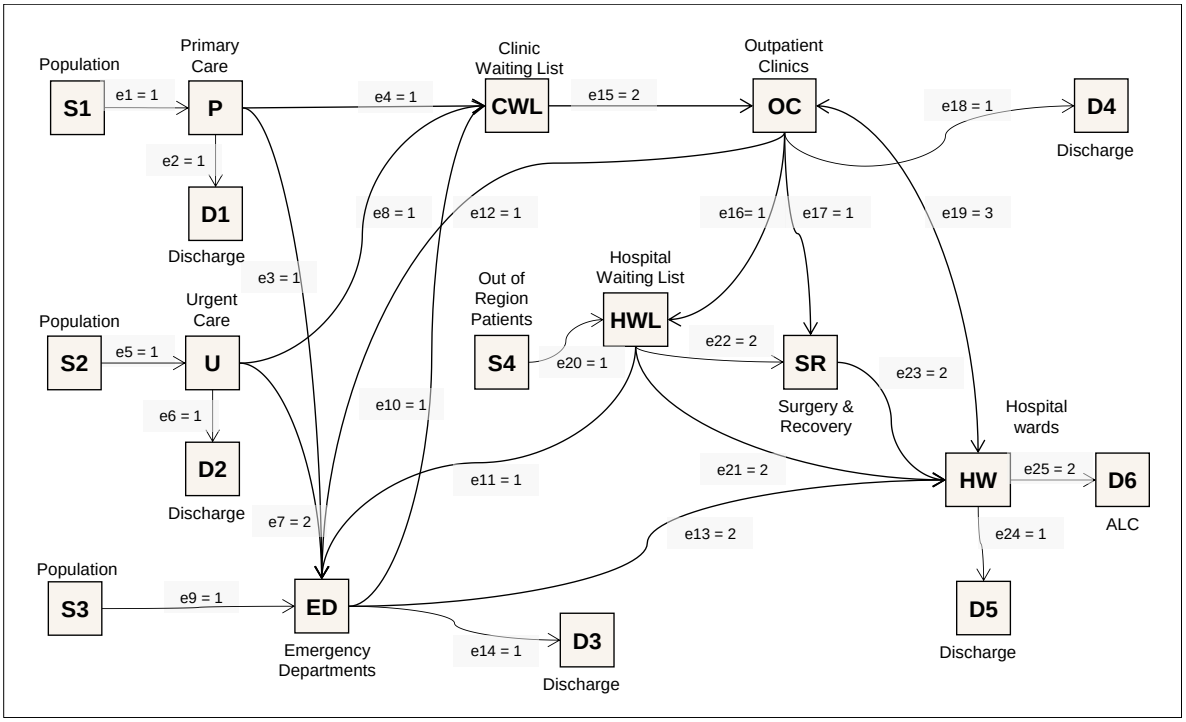


Figure 6-2 Graph Representation of the CHR Healthcare System (G1)

One important step in constructing the graph representation is to determine the value of the edges which represent the relationships between the nodes in the graph. In reality, there can be many different types of relationships between two nodes. For example, the relationship between the node ED (Emergency Departments) and HW (Hospital Wards) can be measured by directions of patient flow, patient volumes, waiting times, shared resources etc. Different measures can be chosen for analyzing a particular type of decision problem. In this chapter, the focus is on the patient flow analysis, so the values of the edges in the graph are determined with the consideration of the following two factors:

1) The directions of the patient flow, and
2) The dependency of the patient release from the upstream node on the availability of resources within the downstream node.

Table 6-2 lists all the possible edge values and the corresponding inter-node relationships.

Table 6-2 Edge Values and Corresponding Inter-node Relationships

Edge Value	Corresponding Inter-node Relationships
1	1) Unidirectional patient flow, and 2) Patient release is not dependent on the availability of downstream resources
2	1) Unidirectional patient flow, and 2) Patient release is dependent on the availability of downstream resources
2	1) Bidirectional patient flow, and 2) Patient release is not dependent on the availability of downstream resources
3	1) Bidirectional patient flow, and 2) Patient releases from one or both nodes are dependent on the availability of downstream resources

Here are some examples:

1) The value of the edge e1 (between node S1 and node P) is 1 because the patient arrival is a unidirectional flow, and a patient's seeking for primary care (leaving S1) is not dependent on whether the physicians in the primary care are available (busy) or not.

2) The value of the edge e13 (between node ED and node HW) is 2 because the patient flow between ED and HW is unidirectional (from ED to HW only) and the admitted ED patients will stay in the ED until there are inpatient beds available in the HW.

3) The value of the edge e19 (between node OC and node HW) is 3 because patients visiting outpatient clinics (OC) can be admitted to hospital wards (HW), patients in hospital wards can also be discharged to outpatient clinics for final therapy, and patients will continue to stay in OC or HW if there are no spaces available in HW or OC.

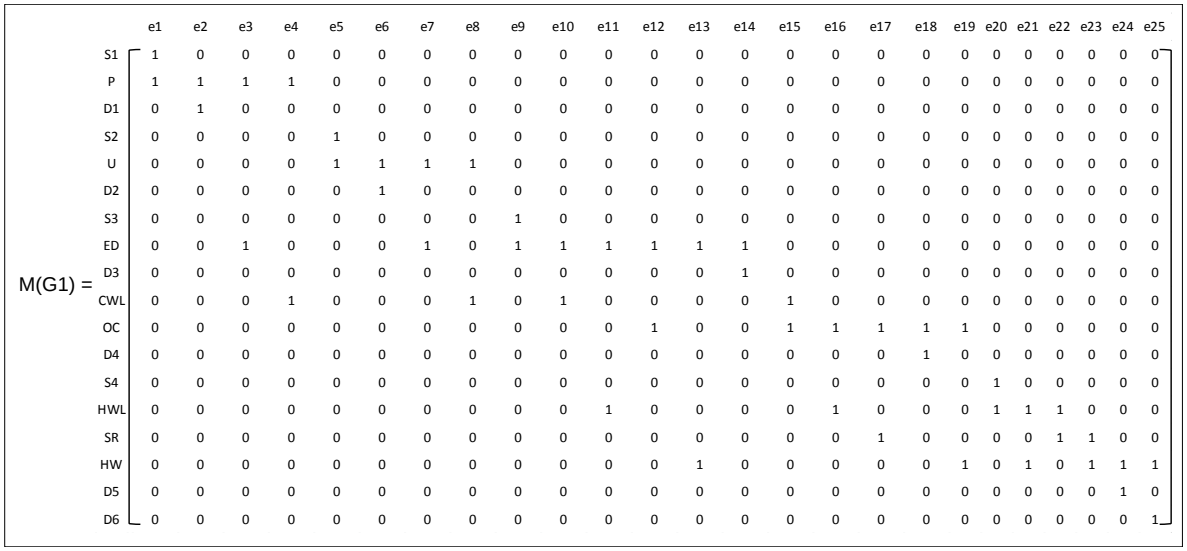
Step 2: Construct the matrices representing the graph

After the graph representation has been constructed for the CHR healthcare system, the next step is to construct the matrices representing the graph. These matrices include the adjacency, incidence and edge weighting matrices, and they are shown in Figure 6-3 to Figure 6-5.

A(G1) =

	S1	P	D1	S2	U	D2	S3	ED	D3	CWL	OC	D4	S4	HWL	SR	HW	D5	D6
S1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
U	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
ED	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CWL	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
OC	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
HWL	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0
SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
HW	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

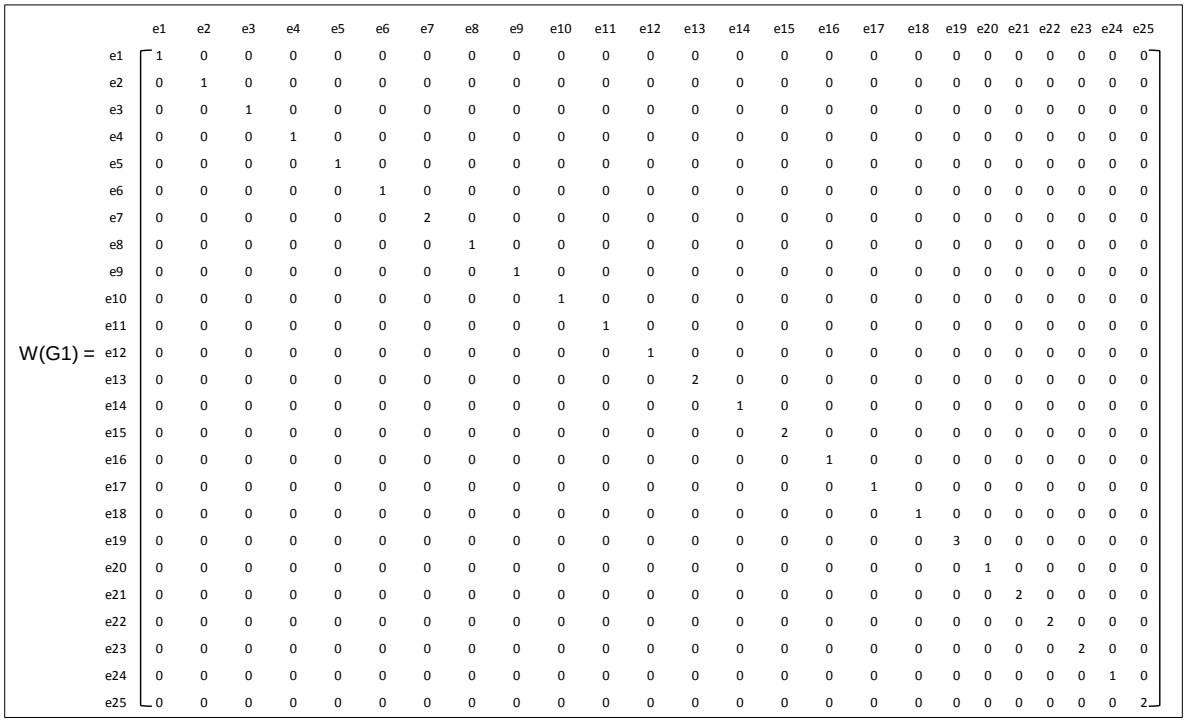
Figure 6-3 Adjacency Matrix for the CHR Healthcare System



M(G1) =

	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12	e13	e14	e15	e16	e17	e18	e19	e20	e21	e22	e23	e24	e25
S1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
U	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ED	0	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
CWL	0	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
OC	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
S4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
HWL	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	1	0	0	0
SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0
HW	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	1	1	1
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 6-4 Incidence Matrix for the CHR Healthcare System



W(G1) =

	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12	e13	e14	e15	e16	e17	e18	e19	e20	e21	e22	e23	e24	e25
e1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e7	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
e13	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
e14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
e15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
e16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
e17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
e18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
e19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
e20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
e21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
e22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
e23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
e24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
e25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2

Figure 6-5 Edge Weighting Matrix for the CHR Healthcare System

Step 3: Calculate the pseudo-covariance (C) and pseudo-correlation (R) matrices

After constructing the three matrices in step 2, the next step is to calculate the pseudo-covariance (C) and pseudo-correlation (R) matrices. The results are shown in Figure 6-6 and Figure 6-7, respectively.

C(G1) =

1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	4	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	5	1	0	2	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	2	0	1	10	1	1	1	0	0	1	0	2	0	0	
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	
0	1	0	0	1	0	0	1	0	5	2	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	1	0	2	9	1	0	1	1	3	0	0	
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	
0	0	0	0	0	0	0	1	0	0	1	0	1	7	2	2	0	0	
0	0	0	0	0	0	0	0	0	0	1	0	0	2	5	2	0	0	
0	0	0	0	0	0	0	2	0	0	3	0	0	2	2	12	1	2	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2	

Figure 6-6 Pseudo-covariance Matrix (C) for the CHR Healthcare System

R(G1) =

1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.500	1.000	0.500	0.000	0.000	0.000	0.000	0.158	0.000	0.224	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.500	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	1.000	0.447	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.447	1.000	0.447	0.000	0.283	0.000	0.200	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.447	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.316	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.158	0.000	0.000	0.283	0.000	0.316	1.000	0.316	0.141	0.105	0.000	0.000	0.120	0.000	0.183	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.316	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.224	0.000	0.000	0.200	0.000	0.000	0.141	0.000	1.000	0.298	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.105	0.000	0.298	1.000	0.333	0.000	0.126	0.149	0.289	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.333	1.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.378	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.120	0.000	0.000	0.126	0.000	0.378	1.000	0.338	0.218	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.149	0.000	0.000	0.338	1.000	0.258	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.183	0.000	0.000	0.289	0.000	0.000	0.218	0.258	1.000	0.289	0.408
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.289	1.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.408	0.000	1.000

Figure 6-7 Pseudo-correlation Matrix (R) for the CHR Healthcare System

Step 4: Perform PCA on the R matrix and determine the number of sub-graphs

After the C and R matrices have been calculated, PCA is performed on the R matrix to extract factors in **MATLAB® 2006**. The extracted factors are shown in Table 6-3.

Table 6-3 Extracted Factors (G1)

Factor	Eigenvalue	% of Variance	Cumulative % of Variance
1	2.000	11.111	11.111
2	1.775	9.860	20.971
3	1.676	9.312	30.282
4	1.372	7.621	37.904
5	1.355	7.526	45.430
6	1.332	7.399	52.829
7	1.000	5.556	58.385
8	1.000	5.556	63.940
9	1.000	5.556	69.496
10	1.000	5.556	75.051
11	1.000	5.556	80.607
12	1.000	5.556	86.163
13	0.616	3.423	89.585
14	0.591	3.281	92.866
15	0.469	2.603	95.469
16	0.338	1.877	97.346
17	0.299	1.659	99.005
18	0.179	0.995	100.000

Following the procedure specified in Subsection 5.1.2, six factors are initially extracted using the ***Latent Root Criterion***. Then we check whether the selection of six factors can meet the ***Percentage of Variance Criterion***. It fails as the total amount of variance explained by the first six factors is only 52.829%, which is less than 60%. We continue to add more factors. Factors 7 to 12 are equally important as they have the same eigenvalue

of 1, but we stop at factor 8 as it is the first factor to have a cumulative of variance explained over 60%. So the number of factors to be retained is eight (8).

Step 5: Determine the structure of each sub-graph

The next step is to perform PCA on matrix C and apply ***VARIMAX*** rotation to the first eight (8) factors. The rotated factor loadings are shown in Table 6-4.

Table 6-4 *VARIMAX* Rotated Factor Matrix – C

Node	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6	Factor 7	Factor 8
S1	-0.003	-0.030	0.019	0.001	0.023	0.282	-0.002	-0.034
P	0.002	-0.011	-0.019	-0.001	-0.016	0.911	0.005	0.111
D1	-0.003	-0.030	0.019	0.001	0.023	0.282	-0.002	-0.034
S2	0.002	-0.052	0.017	0.003	0.241	0.021	-0.002	-0.030
U	0.003	-0.014	-0.019	-0.004	0.933	-0.020	0.007	0.100
D2	0.002	-0.052	0.017	0.003	0.241	0.021	-0.002	-0.030
S3	-0.021	0.128	-0.005	-0.016	-0.041	-0.022	0.007	-0.010
ED	-0.001	0.975	0.010	0.033	0.047	0.031	-0.033	0.057
D3	-0.021	0.128	-0.005	-0.016	-0.041	-0.022	0.007	-0.010
CWL	0.050	-0.053	-0.082	0.008	-0.079	-0.080	0.001	0.975
OC	-0.061	-0.015	0.975	-0.003	-0.003	-0.007	0.032	0.105
D4	-0.044	-0.006	0.159	-0.007	0.025	0.026	-0.020	-0.058
S4	-0.008	-0.026	-0.007	0.201	0.005	0.001	-0.063	0.004
HWL	-0.034	-0.022	-0.003	0.973	-0.001	-0.001	0.117	-0.007
SR	-0.126	0.025	-0.046	-0.107	-0.002	-0.002	0.976	0.007
HW	0.947	0.025	0.089	0.025	-0.001	0.005	0.153	-0.057
D5	0.107	-0.017	-0.022	-0.009	0.002	0.000	-0.017	0.005
D6	0.255	-0.048	-0.065	-0.011	0.002	-0.008	-0.063	0.033

From Table 6-4, we can see that:

- Nodes HW, D5 and D6 load on Factor 1,
- Nodes S3, ED and D3 load on Factor 2,
- Nodes OC and D4 load on Factor 3,
- Nodes S4 and HWL load on Factor 4,

- Nodes S2, U and D2 load on Factor 5,
- Nodes S1, P and D1 load on Factor 6,
- Node SR loads on Factor 7, and
- Node CWL loads on Factor 8.

This result makes sense as the nodes representing the patient sources and destinations are highly related to their associated services. A graphic representation of the eight (8) sub-graphs is provided in Figure 6-8.

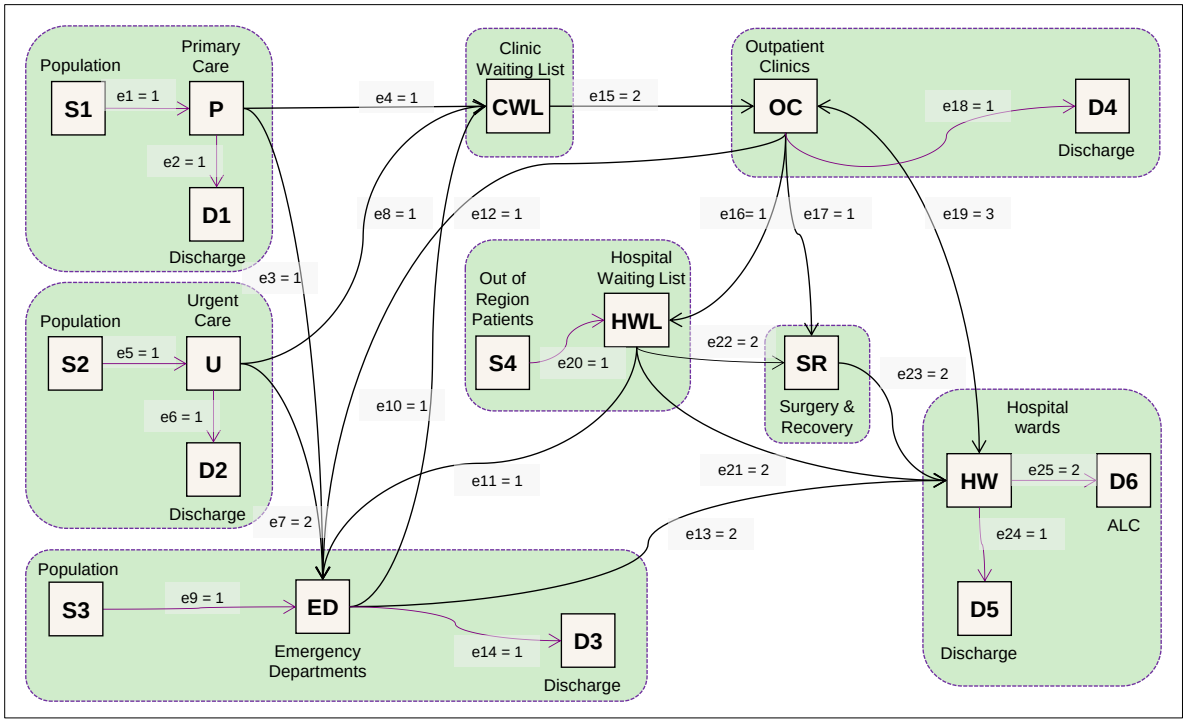


Figure 6-8 8-Subgraph Decomposition of the CHR Healthcare System

Step 6: Name each sub-graph

The naming of the eight (8) sub-graphs is very straightforward. It can be done by using the same ones initially assigned (Table 6-5) as their loadings on the corresponding factors are very high (>90%).

Table 6-5 Names of the 8 Sub-graphs

Name	Full Name or Explanation
P	Primary Care
U	Urgent Care
ED	Emergency Departments
OC	Outpatient Clinics and Consultants Services
SR	Surgery & Recovery
HW	Hospital Wards
CWL	Patient Waiting Services for outpatient clinics and consultants services
HWL	Patient Waiting Services for surgery and hospital wards

6.3 CHR Healthcare System Further Decomposition

In the previous section, the CHR healthcare system is decomposed into eight (8) subsystems (sub-graphs) using the system decomposition technique. In this section, the system decomposition is applied again to further decompose the CHR healthcare system into higher level subsystems, each consisting of some of the eight (8) subsystems identified in the previous section. In order to apply the same process as in the previous section, all the elements of each of the 8 subsystems are consolidated to a single node by

ignoring all the inter-element interactions within the subsystem and only keeping those interactions between this subsystem and other subsystems. The result is shown in Figure 6-9 as the first step of the decomposition process.

Step 1: Construct the graph of the system to be decomposed

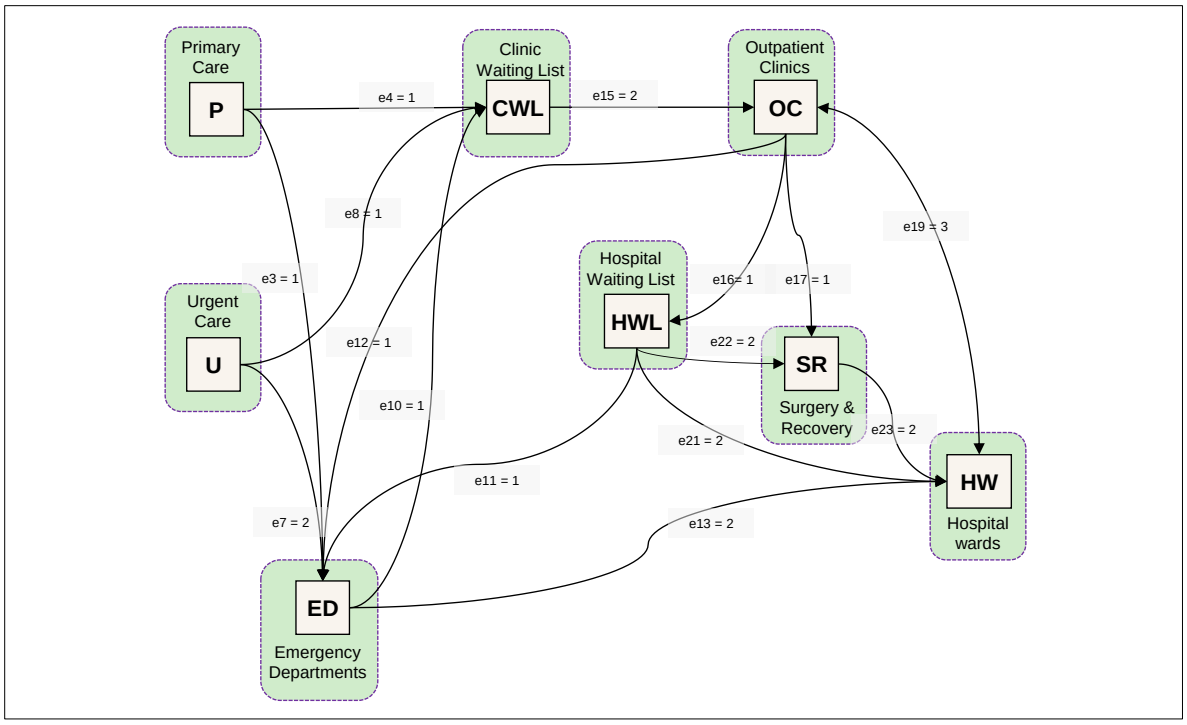
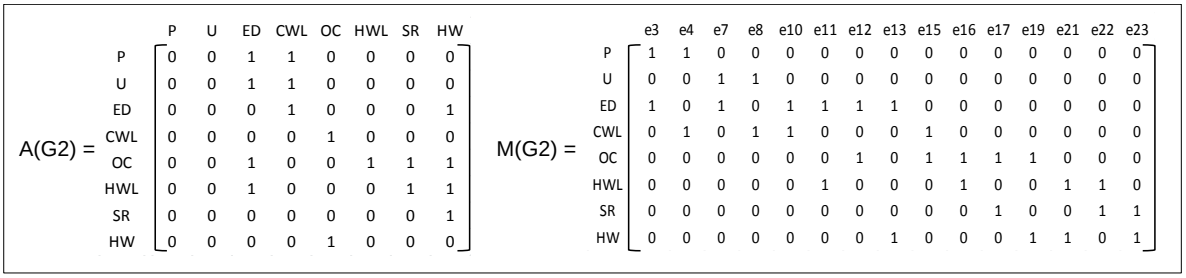


Figure 6-9 Graph of the CHR Healthcare System [Consolidated] (G2)

Step 2: Construct the matrices representing the graph

The adjacency, incidence and edge weighting matrices of the graph shown in Figure 6-9 are presented in Figure 6-10.



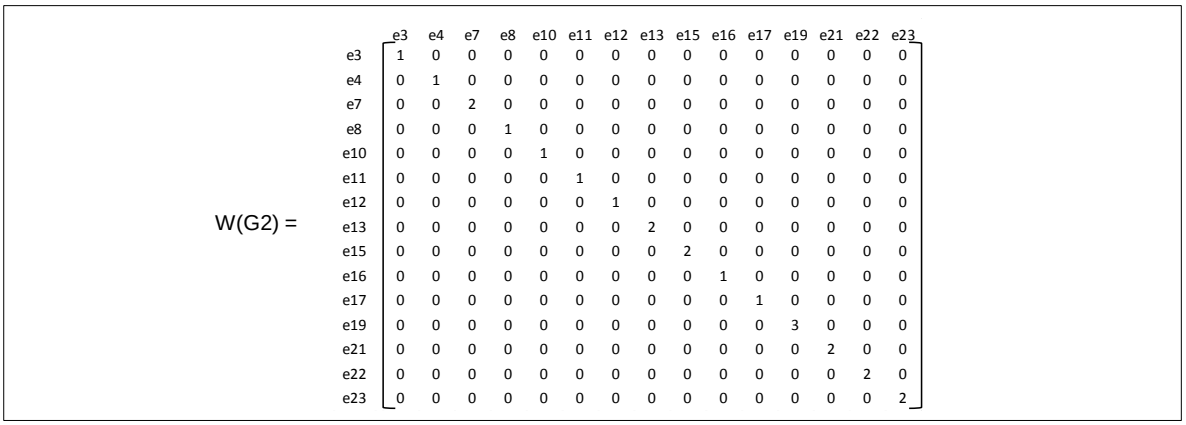


Figure 6-10 Adjacency, Incidence and Edge Weighting Matrices [Consolidated]

Step 3: Calculate the pseudo-covariance (C) and pseudo-correlation (R) matrices

The calculated pseudo-covariance (C) and pseudo-correlation (R) matrices are shown in Figure 6-11.

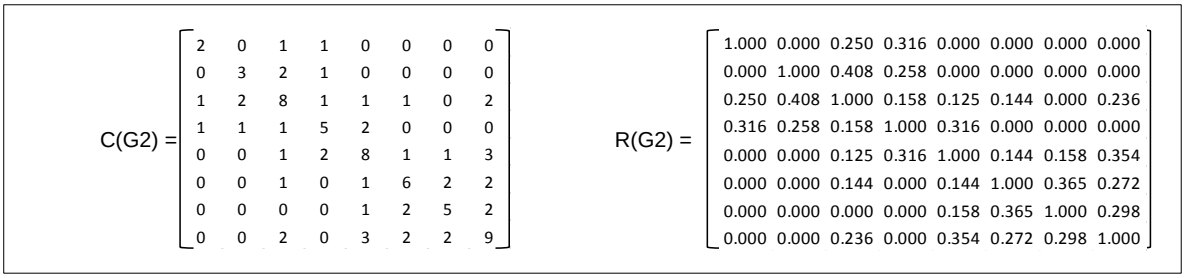


Figure 6-11 Pseudo-covariance and Pseudo-correlation Matrices [Consolidated]

Step 4: Perform PCA on the R matrix and determine the number of sub graphs

After performing PCA on the R matrix, the extracted factors are shown in Table 6-6.

Table 6-6 Extracted Factors (G2)

Factor	Eigenvalue	% of Variance	Cumulative % of Variance
1	2.000	25.000	25.000
2	1.578	19.728	45.065
3	1.102	13.773	57.565
4	0.978	12.229	68.905
5	0.842	10.522	79.542
6	0.650	8.122	88.221
7	0.479	5.985	95.254
8	0.371	4.640	100.000

Following the same procedure specified in Subsection 5.1.2, four factors are retained to meet the ***Percentage of Variance Criterion***.

Step 5: Determine the structure of each sub graph

After performing a PCA on the C matrix, the factor loadings are calculated for the three retained factors using ***VARIMAX*** rotation method. The result is shown in Table 6-7.

Table 6-7 Factor Loadings with *VARIMAX* Rotation

Node	Factor 1	Factor 2	Factor 3	Factor 4
P	-0.065	0.168	0.058	0.008
U	-0.065	0.351	0.016	-0.004
ED	0.217	0.867	-0.095	-0.042
CWL	-0.290	0.264	0.533	0.026
OC	0.178	-0.079	0.837	-0.012
HWL	-0.042	0.100	-0.035	0.810
SR	0.047	-0.086	0.032	0.584
HW	0.909	-0.061	0.031	0.028

From Table 6-7, we can see that:

- Node HW loads on Factor 1,
- Nodes P, U and ED load on Factor 2,
- Nodes CWL and OC load on Factor 3, and
- Nodes HWL and SR load on Factor 4

This means that the CHR healthcare system can be represented at the next layer as comprised of four subsystems. Figure 6-12 provides a graphic presentation of the decomposition result.

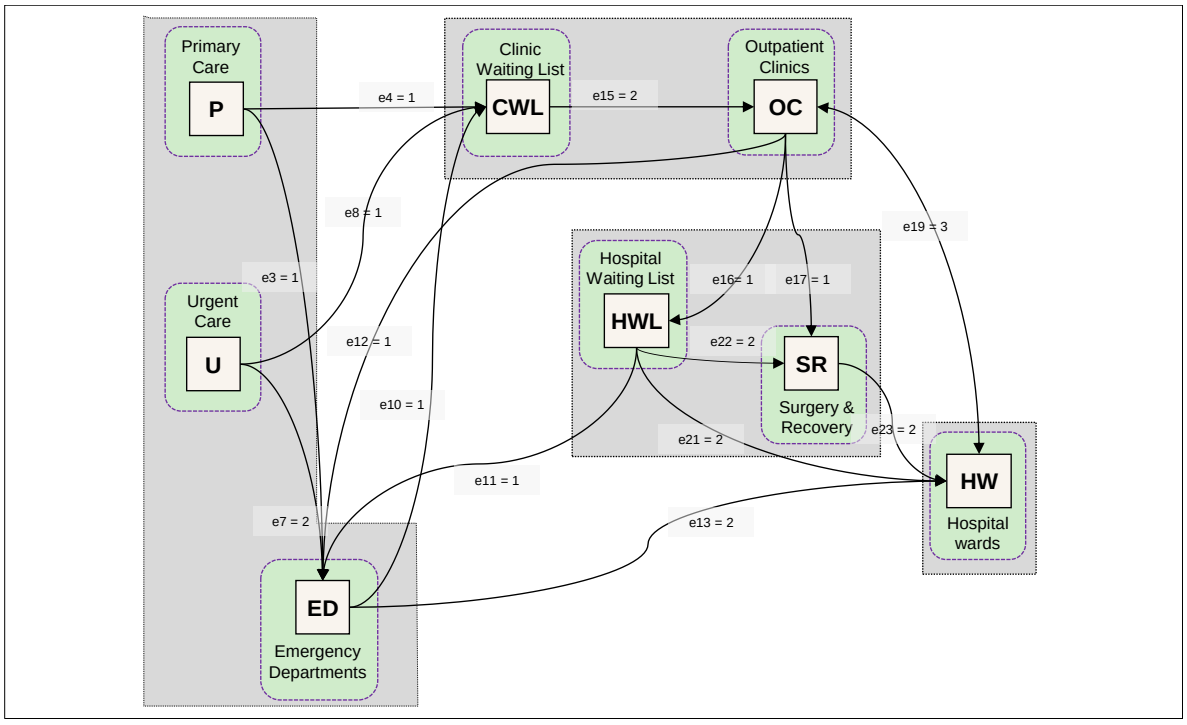


Figure 6-12 4-Subsystem Decomposition of the CHR Healthcare System

Step 6: Name each sub-graph

Now, we need to produce new names to interpret the decomposed CHR healthcare subsystems.

- The first subsystem includes only the hospital wards, so we call it *"Hospital Wards Services"* and it offers inpatient services to referred patients only.
- The second subsystem includes primary care, urgent care and the emergency departments. We call this subsystem "*Primary, Urgent and Emergency Care Services*" and it opens to the public.
- The third subsystem includes the clinic waiting list and the outpatient clinics. We call it "*Outpatient Services*" and it offers outpatient services to referred patients only.
- The fourth subsystem includes surgery & recovery and the hospital waiting services. We call this subsystem "*Hospital Waiting and Surgery Services*", and it provides services to referred patients only.

After the decomposition is finished, an assessment needs to be done to decide whether further decomposition is necessary. In this example, there are only 4 subsystems left so there is no need to continue the decomposition process.

From the above decomposition results, we can see that the CHR healthcare system can be represented as a three-layer system with details summarized in Table 6-8.

Table 6-8 Hierarchical Structure of the CHR Healthcare System

Layer 3 (The Whole System)	**Layer 2 (Subsystems)**	**Layer 1 (Units)**
CHR Healthcare System	Hospital Wards Services	1) Hospital wards
	Primary, Urgent and Emergency Care Services	1) Primary care 2) Urgent care 3) Emergency departments
	Outpatient Services	1) Outpatient waiting service 2) Outpatient clinics
	Hospital Waiting and Surgery Services	1) Hospital waiting service 2) Surgery and recovery

At the top layer (layer 3) sits the CHR healthcare system. At layer 2, there are four subsystems: 1) ***Hospital Wards Services***, which are for referred patients only; 2) ***Primary, Urgent and Emergency Care Services***, which open to the public; 3) ***Outpatient Services***, which are offering outpatient services and only open to referred patients; and 4) ***Hospital Waiting and Surgery Services***, which are only open to referred patients as well. Each subsystem at layer 2 consists of one or more units at layer 1: 1) ***Hospital Wards Services***: Hospital wards; 2) ***Primary, Urgent and Emergency Care Services:*** Primary care, Urgent care and Emergency departments; 3) ***Outpatient Services***: Outpatient waiting list and outpatient clinics; and 4) ***Hospital Waiting and Surgery Services***: Hospital waiting services and surgery & recovery. Figure 6-13 shows a graphic representation of the hierarchical structure of the CHR healthcare system.

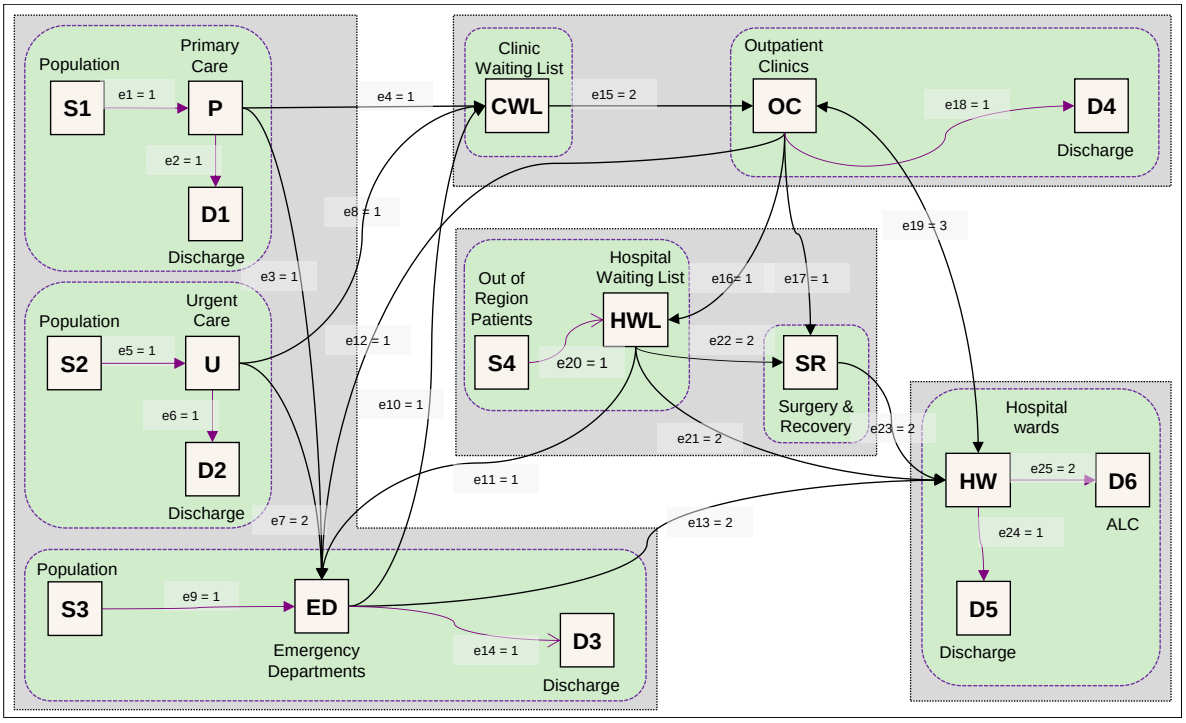


Figure 6-13 Hierarchical Structure of the CHR Healthcare system

6.4 Benefits of the Identified Hierarchical Structure

In the previous sections, we have shown how to use the system decomposition technique to identify the hierarchical structure of the CHR healthcare system. The identified hierarchical structure can benefit the SM study of patient flows within CHR healthcare system in a number of ways. The most important one is the use of this hierarchical structure as a guiding map in conducting various SM studies of the CHR healthcare system. This can be further elaborated from the following two perspectives: 1) developing new SM studies from scratch; and 2) enhancing reusability of existing simulation models.

6.4.1 *Developing New SM Studies From Scratch*

For new SM studies of the CHR healthcare system, the identified hierarchical structure can be used to define the required tasks and their sequences. For example, simulation models can first be built to represent each of the eight healthcare service units shown in Figure 6-13. These simulation models can be developed, verified and validated independently as long as the inputs and outputs of the simulation models have captured all the inter-unit interactions. Moreover, the independent development of simulation models for individual units also increases the possibility for the reuse of existing simulation models. This can speed up the development process of the simulation models.

Once the simulation models of the eight units have been developed, they can be integrated using DMI, MMI or HMI approaches to represent the subsystems at layer 2. These layer 2 subsystem simulation models can again be verified and validated independently as long as they have captured all the interactions between the units at layer 1 of the CHR healthcare system. The layer 2 subsystem simulation models can be further integrated to represent the whole CHR healthcare system.

This approach is very clear and logical making it easy to coordinate SM efforts for the study of MUHSs with a large number of units. With this approach, multiple unit and integrated simulation models instead of only one complex simulation model will be developed and maintained at the unit level for the study of a particular MUHS. This will make it easy to keep these simulation models current and increase their reusability in the future studies of the MUHS.

6.4.2 *Enhancing Reusability of Existing Simulation Models*

The identified hierarchical structure of a MUHS can also provide a cost-effective solution to enhancing the reusability of existing simulation models of units of the MUHS. Very often, a single complex simulation model is developed for a MUHS with large number of units. This simulation model can be very powerful in the study of the MUHS with particular purposes but the model itself can be very difficult for people not involved in the development of the simulation model to understand. In many cases, these simulation models are used as black boxes once they are developed. It is usually very difficult to modify them for other purposes of study.

With the identified hierarchical structure of a MUHS, the simulation model representing the MUHS can be more easily modified to improve its understandability. Here, we will use the CHR SD model as an example. This model was developed by Dr. David Cooke and is a very complex model populated with much data. The model can be used to test various management scenarios for improving patient flows within the CHR (Cooke et al. 2008). With permission from Dr. Cooke, the author was allowed to use and modify the CHR SD model in this doctoral research.

The author was involved in the analysis of the input data used in the CHR SD model, but was not involved in the development of the model itself. It took the author quite a while to become familiar with the details of the CHR SD model. Then the author realized that it could take others much longer time to understand this model if they do not have any exposure to the development of the CHR SD model. So the author decided to rebuild the

CHR SD model to make it easier to understand, and the identified hierarchical structure of the CHR healthcare system became very useful in that process.

Below is the process the author followed in the rebuilding of the CHR SD model:

1) Develop an SD model for each service/unit. The inflows and outflows of each SD model are determined following the hierarchical structure of the CHR healthcare system shown in Figure 6-13. For example, the EDs in the CHR have the following inflows and outflows:
    a. **Inflows:** walk in patients, elective patients from primary care, urgent care, outpatient clinics and hospital waiting list.
    b. **Outflows:** discharged to home and admitted to hospital wards.
2) Run the original CHR SD model and get the stabilized inflow and outflow rates. These inflow rates are used to drive the individual SD models and the outflow rates are used to calibrate each SD model so that its outflow rates match the results from the running of the original CHR SD model.
3) Integrate the individual SD models to build subsystem SD models representing the subsystems at layer 2. For example, integrate the SD models of primary care, urgent care and emergency care to represent the *Primary, Urgent and Emergency Care Services* subsystem SD model. Each subsystem SD model is run and calibrated using the inflow rates and outflow rates generated from running the original SD model.

4) Further integrate the subsystem SD models to develop the CHR SD model. The results from the running of the original CHR SD model are used to calibrate the integrated CHR SD model.

The above process is very efficient as it makes it very easy to trouble-shoot problems in the process of model development. Following this process, the author only spent about a day rebuilding the CHR SD model, which has exactly the same features as the original CHR SD model. This example does show that the identified hierarchical structure of the CHR healthcare system helps to increase the possibility of reusing part of or the entire CHR SD model. The rebuilt CHR SD model has a modular structure with each module representing the operations of an individual service unit. In addition, there are also 11 more SD models developed in the process of rebuilding the CHR SD model. These 11 SD models include:

- Three layer 2 subsystem SD models, and
- Eight unit SD models.

The SD models at each layer can be run and validated independently within the content of the whole CHR healthcare system, but they can also be used to construct or update SD models representing subsystems at higher layer. This will speed up the development and updating of the integrated CHR SD model thus improving the rapidity of the SM study of the CHR healthcare system. More importantly, these SD models representing subsystems and units at different layers can be easily reused in future SM studies if only some of the units are involved.

6.5 Discussion and Conclusions

In this chapter, the CHR healthcare system is used as an example for the application of the proposed system decomposition technique in Section 5.1 of Chapter 5. A graph is first constructed to represent the eight units and their interactions in the CHR healthcare system. This graph is converted into association matrices, which are factored using standard PCA routines in **MATLAB® 2006**. Each factor is equivalent to a subsystem and factor loading analysis is performed for each factor to determine the units included in each subsystem. This process will repeat until no further decomposition is needed. Based on the analysis result, the CHR healthcare system can be decomposed into a three-layer system. At the top layer (layer 3) is the CHR healthcare system. At layer 2, there are four subsystems: 1) ***Hospital Wards Services***, which are for referred patients only; 2) ***Primary, Urgent and Emergency Care Services***, which open to the public; 3) ***Outpatient Services***, which are offering outpatient services and only open to referred patients; and 4) ***Hospital Waiting and Surgery Services***, which are only open to referred patients. Each subsystem at layer 2 consists of one or more units at layer 1: 1) ***Hospital Wards Services***: Hospital wards; 2) ***Primary, Urgent and Emergency Care Services:*** Primary care, Urgent care, and Emergency departments; 3) ***Outpatient Services***: Outpatient waiting list and outpatient clinics; and 4) ***Hospital Waiting and Surgery Services***: Hospital waiting services and surgery & recovery.

This application example shows that the proposed system decomposition technique follows a logical procedure and is easy to use. The only subjective portions in this approach are the definition of units, the measures of the inter-unit interactions, and the

decision on how many factors to retain. This helps to minimize the impact of human error and personal bias from different stakeholders involved in SM studies of a MUHS.

The identified hierarchical structure of a MUHS can be used as a guiding map for conducting various SM studies of that MUHS. For new SM studies starting from scratch, the hierarchical structure can be used as a common ground to bring all the stakeholders on the same page and determine the research tasks and their sequences so that simulation models can be developed independently within the content of the scope of the overall MUHS SM study. These independently developed SM model can be integrated to construct simulation models representing subsystems at the higher layer or the MUHS at the top. For existing SM studies of a MUHS, the hierarchical structure can be used to add modularity to the single complex simulation model to make it easier to understand. In addition, it can also be used to develop simulation models representing units or subsystems at lower layers, thus improving the reusability of the simulation models in future SM studies.

Chapter 7 Metamodeling of an ED DES Model

7.1 Introduction

In AHSM, simulation metamodeling is proposed as a technique to represent and simplify the unit simulation models of a MUHS. In this chapter, an application example is presented using the procedure introduced in Section 5.2 of Chapter 5, applied to a DES model of patient flow within the ED at the Foothills Medical Center (FMC) in the CHR that was developed in 2008 (Xu et al., 2008).

In the following section, a brief overview is provided of the features of the original FMC ED DES model. The development of the NN-based simulation metamodels to represent the FMC ED DES model is introduced in Section 7.3. Major findings and conclusions from this application example are discussed in Section 7.4.

7.2 The Original FMC ED DES Model

The objectives of the FMC ED DES model are to identify patient flow problems in the FMC ED and provide a test environment for suggested improvement alternatives (Xu et al., 2008). The FMC ED in the Alberta, Canada is an emergency care facility handling over 70,000 patient visits annually. The FMC ED is staffed with 10 emergency physician (EP) shifts per day. It has a total of 53 ED treatment beds (ED beds), and 21 to 23 treatment nurses on duty depending on the time of the day. The flow of patients through the FMC ED, represented in the FMC ED DES model is shown in Figure 7-1 and can be described as follows:

- Patients arrive and wait for triage.
- Patients are triaged to determine their CTAS (Canadian Triage and Acuity Scale) score and the type of ED bed required.
- Patients wait for ED beds or LWBS (Leave Without Being Seen).
- Patients getting ED beds wait for ED nurse assessment.
- ED nurse assessment determines a patient's need for DI (Diagnostic Imaging) or lab tests.
- Patients who need DI or lab tests take those tests and wait for the results to be made available.
- Patients who do not need DI or lab tests and those with their DI or lab test results back wait for a 1st EP assessment. While waiting for 1st EP assessment, some patients may choose LWBS.
- First EP assessment resulting in a disposition decision of one of three types: discharge, take more lab tests, or recommended for admission. Some patients may LAMA (Leave Against Medical Advice) after the 1st EP assessment.
- Patients who need more DI or lab tests take those tests and wait for the results to be made available. After their test results are back, they wait for EP reassessment.
- EP reassessment resulting in a disposition decision of one of the three types noted earlier. Patients who need more DI or lab tests will repeat the previous step.

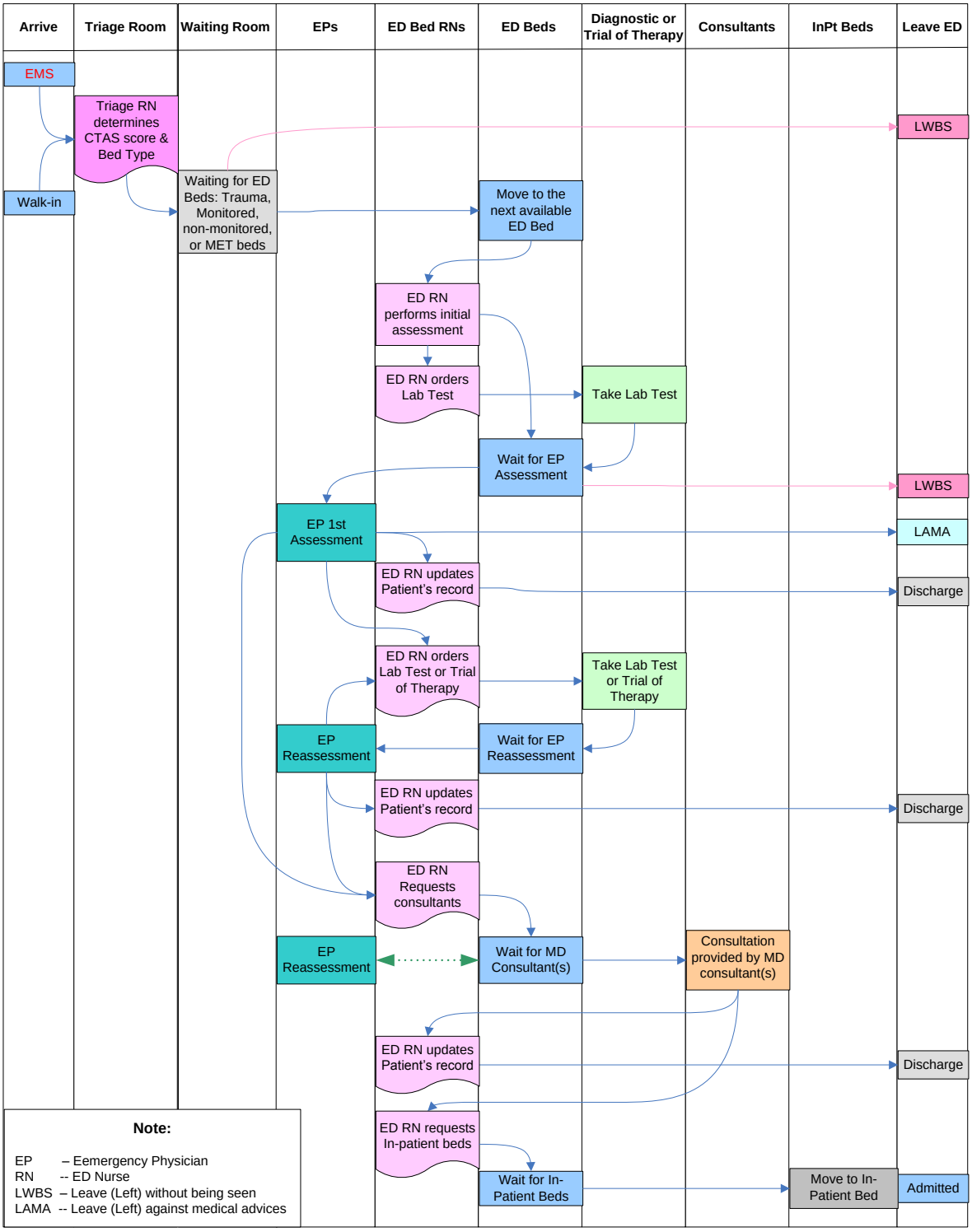


Figure 7-1 Diagram of Patient Flow within the FMC ED

- Patients with admission recommended wait for assessments from a consulting physician.
- Consulting physician assessment results in a decision to either discharge or admit the patient.
- Patients admitted wait in the ED until inpatient beds are available in the main hospital wards.

Data from the following sources are incorporated in the FMC ED DES model:

- Over 30,000 patient records that included time stamps, acuity score, and other key pieces of information.
- Information on EP activities from 20 shifts where EPs were "shadowed" by data collectors.
- More than 37,000 DI records and 140,000 CLS (Calgary Lab Services) records were analyzed to estimate patient need for such testing and the test response times.

The FMC ED DES model was developed in an iterative manner, through a number of intermediate versions, given the fact that the data highlighted above did not become available all at once. Interim model results were reviewed by the research team, and presented to some FMC ED staff for comments.

Some of the key features of the FMC ED DES model include:

- Explicit modeling of EP capacity: includes details on the shift pattern that EPs work and on the tasks that they must deal with during a shift.

- Modeling of the contribution of DI and lab test delays to the length of stay (LoS) in the ED: includes details on patient need for these tests and on the test response time, so that their impact on LoS is explicitly modeled.
- Modeling of LWBS patients using a NN model: recognizes that this is not a fixed percentage but rather it depends upon the wait times being experienced by patients.

The FMC ED DES model was built using ARENA® 12.0 and was validated by comparing the simulation model's output values to those collected from the operations of the FMC ED. Details on the model data and validation results can be found in Appendix B.

The validated model can be used to identify bottlenecks in the FMC ED processes and test the impact of a variety of changes on the performance of the FMC ED, including (but not limited to) changes in the patient demand pattern, the EP staffing schedule, and the number of ED beds. Managerial questions such as the following can be answered by the model:

- How will ED performance improve by expanding or changing accelerated treatment areas (e.g., sub-waiting rooms, rapid access zones)?
- How will changes to EP hours or shifts affect patient waiting times?
- If operational changes are made that improve ED treatment times or reduce delays, what is the overall impact on ED patient flow?

7.3 Development of NN-based Simulation Metamodels

In this section, the steps mentioned in Chapter 5 (Figure 5-9) are followed to develop the NN-based simulation metamodels for the FMC ED DES model.

Step 1: Determine metamodeling purpose and scope

The metamodeling purpose and scope are determined based on the decision problems to be addressed. In this application example, let's assume the purpose of metamodeling is to analyze the impact of the patient arrival volume and boarding time of admitted patients on the performance of the ED. The chosen performance indicators are as follows:

- Time patients spend in the waiting room,
- Percentage of patients who LWBS,
- Time patients spend waiting for 1st EP assessment after getting an ED bed (EP time), and
- Work-up time, which is the duration between the 1st EP assessment and the disposition decision made by the EP. The work-up time can be further differentiated between work-up time for patients, who are discharged by the EPs (Work-up time for non-consulted patients) and work-up time for patients, who are recommended for admission consult (Work-up time for consulted patients).

Figure 7-2 shows the scope of the metamodeling.

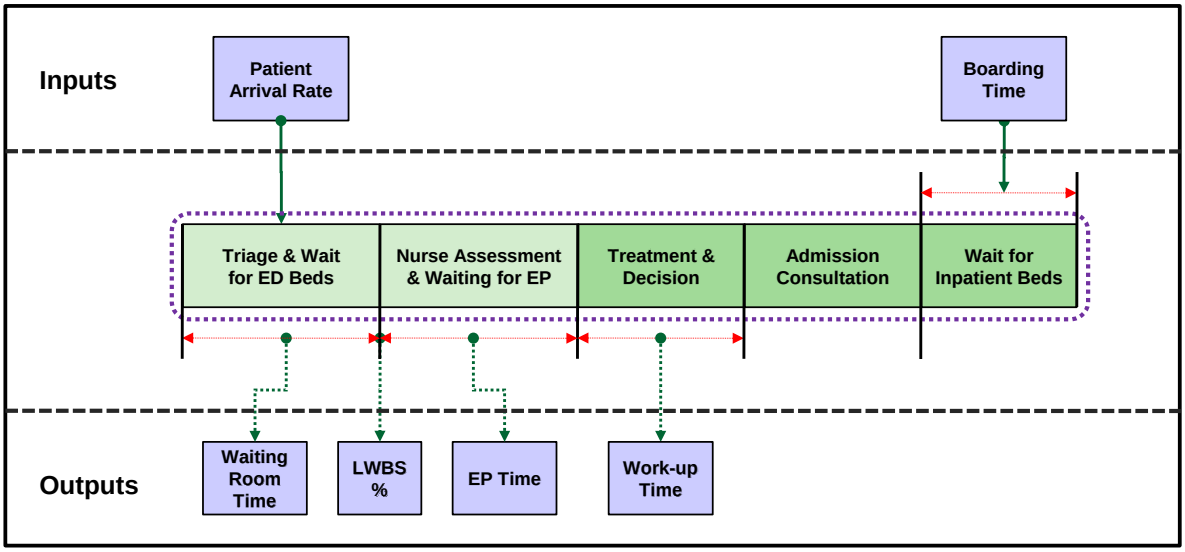


Figure 7-2 Scope of the FMC ED DES Metamodel

There are two inputs: 1) patient arrival rate, and 2) boarding time of admitted ED patients. For simplicity purpose, the two inputs of the metamodels are represented as a percentage of their value used in the baseline scenario of the FMC ED DES model (as determined when the FMC ED DES model was developed). For example, if the patient arrival rate is 110%, this means that the patient arrival rate will be increased by 10% over the baseline value.

As explained in Subsection 5.2.3, the ranges of the metamodel inputs are determined with the consideration of the following two main factors: 1) the purpose of the simulation metamodel, and 2) the limitation of the original simulation model. Let's assume the FMC ED management team is concerned about two performances of the ED operations: 1) average time patients spent in the waiting room (waiting room time); and 2) average boarding time of the admitted patients (boarding time), and they are 1.7 hours and 3.8

hours respectively when the FMC ED DES model was developed. The ED management team is now interested in the following ranges:

1) Waiting room time: 0.5 to 5 hours
2) Average boarding time: 3 to 6 hours

As the average boarding time is a direct input to the simulation metamodel, it is easy to determine its range, which is 66% to 158%. To leave some margin, we extend the range of the average boarding time, and use the range of 60% to 160%.

For the waiting room time, we need to run the original FMC ED DES model for different patient arrival rates to determine its range by starting with very high (180%) and very low (50%) then reducing/increasing them gradually. Each experimental run is with a replication length of 485 days including 120 days as the warm-up period. The replication length and the warm-up period are determined at the validation stage of the original FMC ED DES model to ensure the model has reached the steady state after the warm-up period. After several rounds of trials, we find the following range for the patient arrival rate:

- 80%: 0.45 hours waiting room time
- 150%: 5.20 hours waiting room time

Then we check other outputs at the upper and lower limit of each input, and see whether the input ranges make senses. After we have obtained confirmation from this check, the ranges of the two inputs are decided as follows:

1) Patient arrival rate: 80% to 150%

2) Average boarding time: 60% to 160%

For the simulation metamodel, there are five types of outputs, and each of them is further divided into five outputs, one for a different CTAS score ranging from 1 to 5. As shown in Table 7-1, there will be 5 x 5 =25 outputs for each pair of inputs.

Table 7-1 25 Outputs of the FMC ED DES Model

Simulation Model Outputs	**CTAS Score**				
	1	2	3	4	5
Waiting Room Time	WR CTAS1	WR CTAS2	WR CTAS3	WR CTAS4	WR CTAS5
LWBS Patients %	LWBS WR CTAS1	LWBS WR CTAS2	LWBS WR CTAS3	LWBS WR CTAS4	LWBS WR CTAS5
EP Time	EP Time CTAS1	EP Time CTAS2	EP Time CTAS3	EP Time CTAS4	EP Time CTAS5
Workup Time for Consulted Patients	WU CNLT CTAS1	WU CNLT CTAS2	WU CNLT CTAS3	WU CNLT CTAS4	WU CNLT CTAS5
Workup Time for Non-consulted Patients	WU NoCNLT CTAS1	WU NoCNLT CTAS2	WU NoCNLT CTAS3	WU NoCNLT CTAS4	WU NoCNLT CTAS5

Step 2: Determine number of replications based on specified precision

Before running the FMC ED DES model to generate the data needed for building the simulation metamodels, it is necessary to determine the number of replications needed to gather the desired statistics. According to Law (2006:500), fixing the number of replications gives little to no control over the half-length of the CI; therefore, an analytical procedure (Banks et al., 2005:395) is performed to determine the number of replications for estimating the mean with a desired precision. Details on this analytical procedure are discussed in Appendix C.

The following specified precisions for the five performance indicators are used for determining the minimum number of replications:

- Average Waiting Room Time (minutes): 2.5%
- Average LWBS Patients %: 5%
- Average EP Time (Minutes): 2.5%
- Average Work-up Time for Consulted Patients: 2.5%
- Average Work-up Time for Non-consulted Patients: 2.5%

The analytical procedure is performed for each performance indicator to determine the minimum number of replications for that particular performance indicator, then the largest value amongst these minima is used as the minimum number of replications. Table 7-2 shows the analysis result, and nine (9) replications are needed.

Table 7-2 Minimum Number of Replications

Simulation Model Outputs	Unit	Replication, r				$\bar{Y}$	S_0	Target ϵ		$(Z_{0.025}S_0/\epsilon)^2$	Minimum R	$(t_{0.025,R-1}S_0/\epsilon)^2$
		1	2	3	4			% of $\bar{Y}$	Value			
Waiting Room Time	Minute	97	103	100	104	101	3.03	2.5%	2.53	5.51	9	7.6
LWBS Patients %	%	6.5	7.3	6.9	7.3	7.0	0.39	5.0%	0.35	4.62	8	6.7
EP Time	Minute	79	83	78	81	80	2.37	2.5%	2.00	5.37	8	7.8
Workup Time for Consulted Patients	Minute	136	139	136	137	137	1.46	2.5%	3.43	0.70	4	1.8
Workup Time for Non-consulted Patients	Minute	128	129	129	129	129	0.74	2.5%	3.22	0.20	3	1.0

And Table 7-3 shows the specified precision of different performance indicators for nine 485-day replications.

Table 7-3 Specified Precisions with Nine 485-day Replications

Simulation Model Outputs	Unit	Replication, r									$\bar{Y}$	S_0	Target ϵ		H
		1	2	3	4	5	6	7	8	9			% of $\bar{Y}$	Value	R=9
Waiting Room Time	Minute	97	103	100	104	107	101	101	105	100	102.1	2.92	2.5%	2.6	2.2
LWBS Patients %	%	6.5	7.3	6.9	7.3	7.5	7.0	7.3	7.0	6.9	7.1	0.30	5.0%	0.4	0.2
EP Time	Minute	79	83	78	81	82	80	84	82	81	81.1	2.01	2.5%	2.0	1.5
Workup Time for Consulted Patients	Minute	136	139	136	137	138	136	139	137	138	137.3	1.13	2.5%	3.4	0.9
Workup Time for Non-consulted Patients	Minute	128	129	129	129	129	129	129	128	129	128.8	0.56	2.5%	3.2	0.4

Steps 3 & 4: Generate experimental data and set up training and testing data

As discussed in Subsection 5.2.3, the generation of the experimental results is an iterative process with several rounds of trial-and-error. We will not describe the details here but only list the final results.

To generate the training data for the NNs, the FMC ED DES model is run at the following values for each of the two inputs:

1) Patient arrival rate: 15 values (80% to 150% with a step of 5%)
2) Average boarding time: 6 values (60% to 160% with a step of 20%)

This generates a total of 15 * 6 = 90 possible sets of input configurations. For each input configuration, the FMC ED DES model is run with nine (9) replications, each with a replication length of 485 days including 120 days as the warm-up period.

For the generation of the test data for the NNs, the FMC ED DES model is run at the following values for each of the two inputs:

1) Patient arrival rate: 6 values (82.5/97.5/107.5/117.5/132.5/147.5%)

2) Average boarding time: 5 values (70% to 150% with a step of 20%)

This creates another total of 6 * 5 = 30 input configurations to generate the outputs of the test data.

Both the training and test datasets are chosen to evenly cover the ranges of the two inputs. Figure 7-3 provides a graphic presentation of the training and test configuration data sets.

Average Boarding Time	Patient Arrival Rate																				
	80.0%	82.5%	85.0%	90.0%	95.0%	97.5%	100.0%	105.0%	107.5%	110.0%	115.0%	117.5%	120.0%	125.0%	130.0%	132.5%	135.0%	140.0%	145.0%	147.5%	150.0%
60%	T		T	T	T		T	T		T	T		T	T	T		T	T	T		T
70%		V				V			V			V				V				V	
80%	T		T	T	T		T	T		T	T		T	T	T		T	T	T		T
90%		V				V			V			V				V				V	
100%	T		T	T	T		T	T		T	T		T	T	T		T	T	T		T
110%		V				V			V			V				V				V	
120%	T		T	T	T		T	T		T	T		T	T	T		T	T	T		T
130%		V				V			V			V				V				V	
140%	T		T	T	T		T	T		T	T		T	T	T		T	T	T		T
150%		V				V			V			V				V				V	
160%	T		T	T	T		T	T		T	T		T	T	T		T	T	T		T

T Training Data Points V Test Data Points Not used

Figure 7-3 Training/Test Data Set for NN-based Simulation Metamodeling

Steps 5 & 6: Choose NN architecture and decide training rules

As discussed in Subsection 5.2.3, we choose to start with a very simple NN. This NN has a three-layer feed-forward architecture consisting of (see Figure 7-4):

1) Input layer: 2 nodes

2) Hidden layer: 2 nodes

3) Output layer: 1 node

Other information on the architecture of the NN models includes:

- **Activation function**:
  - Hyperbolic tangent for hidden layer; and identity for output layer
- **Rescaling of scale dependent variables:**
  - standardized to be in the range (0-1)

The numbers of nodes in the hidden layers is chosen to be the same as the number of inputs (Xu, 2010). For simplicity purpose, we call this architecture a 2-2-1 NN model.

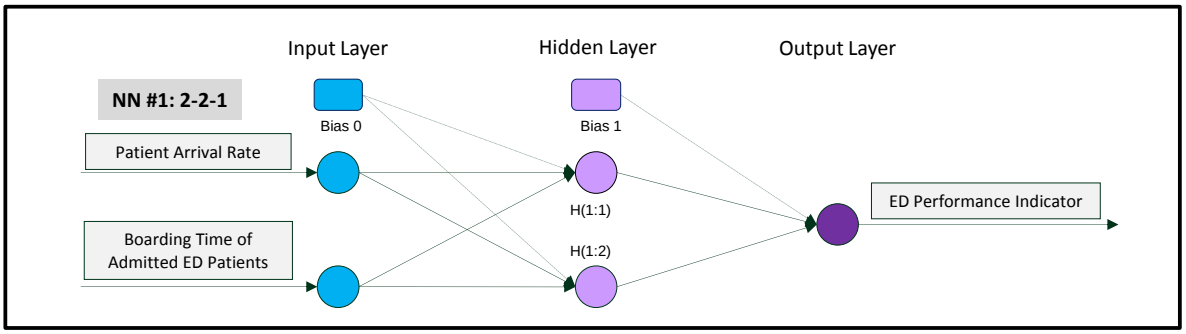


Figure 7-4 The 2-2-1 NN Architecture

Each NN model will be trained for each of the 25 outputs. This means that a total 25NNs need to be trained, one for each output as determined in step 1.

The final training rules used are determined following the steps mentioned in Subsection 5.2.3. These rules are listed as follows:

- **Type of training:**
  - Batch mode
- **Optimization algorithm:**

- Gradient descent

- **Training options:**
  - Initial learning rate: 0.2
  - Momentum: 0.9
  - Interval center: 0
  - Interval offset: +/-0.5

- **Stopping rules**
  - Maximum steps without a decrease in error: 10,000
  - Maximum training time: 1 hour
  - Maximum training epochs: 1,000,000
  - Minimum relative change in training error: 1.0e-8
  - Minimum relative change in training error ratio: 1.0e-8

Steps 7, 8 & 9: NN training, performance evaluation, and check of satisfaction

The training rules specified above are used to train the NN models. After being trained, the measures listed in Table 7-4 are used to evaluate the performances of the 25 trained NNs.

Table 7-4 Performance Measures of the Trained NN Models

Simulation Model Outputs	CTAS Score					Performance Measure	Performance Target
	1	2	3	4	5		
Waiting Room Time	WR CTAS1	WR CTAS2	WR CTAS3	WR CTAS4	WR CTAS5	MAPD	2.5%
LWBS Patients %	LWBS WR CTAS1	LWBS WR CTAS2	LWBS WR CTAS3	LWBS WR CTAS4	LWBS WR CTAS5	MAD	0.25%
EP Time	EP Time CTAS1	EP Time CTAS2	EP Time CTAS3	EP Time CTAS4	EP Time CTAS5	MAPD	2.5%
Work-up Time for Consulted Patients	WU CNLT CTAS1	WU CNLT CTAS2	WU CNLT CTAS3	WU CNLT CTAS4	WU CNLT CTAS5	MAPD	2.5%
Work-up Time for Non-consulted Patients	WU NoCNLT CTAS1	WU NoCNLT CTAS2	WU NoCNLT CTAS3	WU NoCNLT CTAS4	WU NoCNLT CTAS5	MAPD	2.5%

In Table 7-4, MAPD means Mean Absolute Percent Deviation, and MAD means Mean Absolute Deviation. MAPD and MAD are defined as follows:

Let:

V_{sim} = The simulated value from the running of the simulation model

V_{NN} = The value predicted by the trained NN

APD = Absolute Percent Deviation of V_{NN} from V_{sim}

AD = Absolute Deviation of V_{NN} from V_{sim}

Then:

$$APD = |V_{NN} - V_{sim}| / V_{sim} * 100\% \quad \text{(Eq. 7-1)}$$

$$AD = |V_{NN} - V_{sim}| \quad \text{(Eq. 7-2)}$$

And

$$MAPD = \sum_{i=1}^{m} APD_i \, / m \quad \text{(Eq. 7-3)}$$

$$MAD = \sum_{i=1}^{m} AD_i \, / m \quad \text{(Eq. 7-4)}$$

Where:

m = number of data points in the training or test data set.

MAD is used to evaluate the NNs for LWBS % as the LWBS values are very small in certain cases. All the other values are using MAPD.

The five NNs for each performance indicator will use the same architecture, and they will be evaluated together. This is to simplify the implementation of NNs.

Table 7-5 shows the evaluation results of the 25 trained 2-2-1 MM models.

Table 7-5 Performance Evaluation of the 25 Trained 2-2-1 NN Models

Simulation Model Outputs	Data Sets	CTAS Score					Perfromance Target	Evaluation (Individual)	Evaluation (Overall)
		1	2	3	4	5			
Waiting Room Time	Training Set	1.100%	1.400%	1.500%	0.700%	1.100%	2.50%	Pass	Pass
	Test Set	0.900%	0.900%	1.100%	0.700%	0.900%	2.50%	Pass	
	Total	1.000%	1.200%	1.400%	0.600%	1.000%	2.50%	Pass	
LWBS Patients %	Training Set	0.009%	0.187%	0.104%	0.632%	0.211%	0.25%	Pass	Fail
	Test Set	0.005%	0.203%	0.080%	0.747%	0.206%	0.25%	Fail	
	Total	0.008%	0.194%	0.099%	0.657%	0.209%	0.25%	Fail	
EP Time	Training Set	1.200%	2.700%	2.700%	1.400%	1.100%	2.50%	Fail	Fail
	Test Set	0.900%	1.600%	1.700%	1.300%	1.100%	2.50%	Pass	
	Total	1.100%	2.300%	2.300%	1.300%	1.000%	2.50%	Pass	
Work-up Time for Consulted Patients	Training Set	0.900%	0.300%	0.200%	0.800%	2.000%	2.50%	Pass	Pass
	Test Set	0.900%	0.400%	0.300%	1.000%	2.000%	2.50%	Pass	
	Total	0.900%	0.300%	0.200%	0.800%	1.900%	2.50%	Pass	
Work-up Time for Non-consulted Patients	Training Set	1.500%	0.300%	0.200%	0.200%	0.600%	2.50%	Pass	Pass
	Test Set	1.000%	0.200%	0.100%	0.200%	0.500%	2.50%	Pass	
	Total	1.300%	0.200%	0.100%	0.200%	0.600%	2.50%	Pass	

From Table 7-5, we can see that trained 2-2-1 NN models for the prediction of LWBS Patient % and EP Time fail to meet the performance target. So a more complex NN model is chosen to be trained for these two types of outputs. This means a total of 10 NNs need

to be retrained using the 25 training data sets. The architecture of this new NN, called a 2-4-1 architecture, is shown in Figure 7-5.

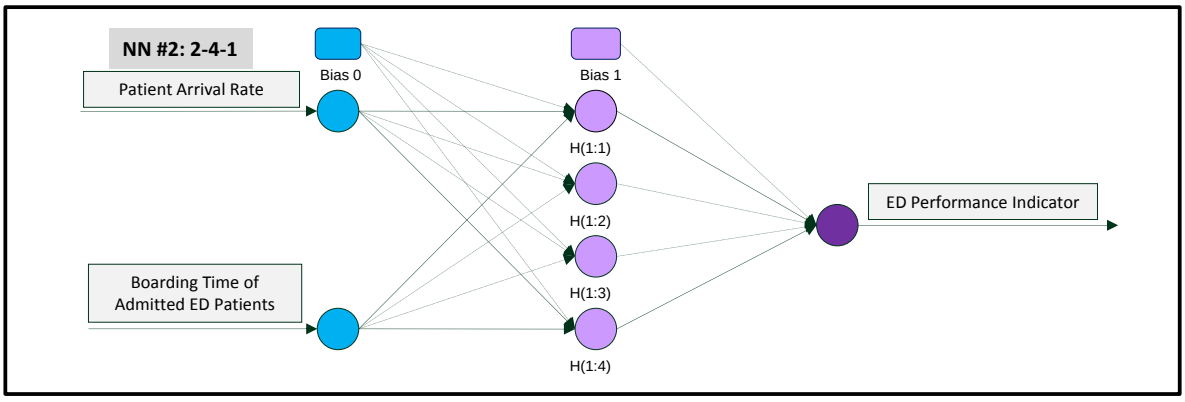


Figure 7-5 The 2-4-1 NN Architecture

As shown in Figure 7-5, the new NN has a three-layer feed-forward structure consisting of:

1) Input layer: 2 nodes

2) Hidden layer: 4 nodes

3) Output layer: 1 node

The numbers of nodes in the hidden layers is chosen to be an integer times of the number of inputs (Xu, 2010). The performance evaluation of the ten (10) trained 2-4-1 NN models is shown in Table 7-6.

Table 7-6 Performance Evaluation of the 10 Trained 2-4-1 NN Models

Simulation Model Outputs	Data Sets	CTAS Score					Perfromance Target	Evaluation (Individual)	Evaluation (Overall)
		1	2	3	4	5			
LWBS Patients %	Training Set	0.007%	0.084%	0.071%	0.159%	0.201%	0.25%	Pass	Pass
	Test Set	0.008%	0.091%	0.061%	0.169%	0.205%	0.25%	Pass	
	Total	0.007%	0.086%	0.068%	0.164%	0.201%	0.25%	Pass	
EP Time	Training Set	1.252%	0.834%	0.582%	0.435%	0.753%	2.50%	Pass	Pass
	Test Set	0.919%	0.449%	0.661%	0.507%	0.962%	2.50%	Pass	
	Total	1.164%	0.710%	0.589%	0.437%	0.776%	2.50%	Pass	

The parameters of the 25 trained NNs can be found in Appendix D.

In any case where a 2-4-1 NNs still cannot meet the performance target, more complex NN architecture, i.e. 6 or more nodes on the hidden layer, can be used to continue the training until the performance targets can be met.

7.4 Discussion and Conclusions

This chapter provides an application example of using the NN-based simulation metamodeling technique for the development of simulation metamodels of the FMC ED DES model. In this application example, there are two inputs for the NN-based simulation metamodels: 1) average volume of arriving ED patients; and 2) boarding time of admitted ED patients. There are five types of output: 1) Time patients spend in the waiting room; 2) Percentage of LWBS patients; 3) EP time; 4) Work-up time for consulted patients; and 5) Work-up time for non-consulted patients. Each type of output is further divided into five outputs, one for each different CTAS score (1-5). The following three-layer feed-forward NN architecture is used for the initial training: 1) input layer: 2 nodes; 2) hidden layer: 2 nodes; and 3) output layer: 1 node. 2.5% MAPD is used as the performance target for all

the outputs except for the Percentage of LWBS patients, for which 0.25% MAD is used as the performance target.

After being developed, the 15 NNs for the following three types of output can satisfy their performance targets using the 2-2-1 NN architecture: 1) Time patients spend in the waiting room; 2) Work-up time for consulted patients; and 3) Work-up time for non-consulted patients. The other 10 NNs for the following two types of outputs can satisfy their performance targets using the 2-4-1 NN architecture: 1) Percentage of LWBS patients; and 2) EP time.

The NN-based simulation metamodels are deterministic models meaning a fixed output is calculated given fixed inputs without the need to run the original simulation model. These metamodels can be implemented using mathematical formulae thus eliminating the need for the simulation software package used to develop and run the original simulation models. In addition, it takes much less time to run the simulation metamodels than the original simulation model to generate outputs with comparable experimental precision.

There are two key findings from this application example. First, it shows that NN is a good candidate for developing simulation metamodels. Second, the NN-based metamodeling process is a trial-and-error process.

Chapter 8 An Integration Example of Simulation Models

8.1 Introduction

In AHSM, it is recommended that simulation models are developed and maintained at the unit level to keep them current. When decision problems occur in a MUHS, unit simulation models can be integrated to develop the MUHS simulation model. Depending on the complexity of the MUHS, the system decomposition technique demonstrated in Chapter 6 may be required to decompose the MUHS into hierarchical layers of subsystems. In this case, unit simulation models are first integrated to develop simulation models representing the subsystems at the lowest layer. Subsequently, these subsystem simulation models are integrated to represent subsystems at higher layers or the overall MUHS. As discussed in Section 5.3 of Chapter 5, there are three model integration approaches, each with its own advantages and disadvantages: DMI, MMI and HMI. In order to assist in choosing the appropriate approach for integrating unit and subsystem simulation models, a procedural FMI approach is proposed including five steps. The first four steps focus on the preparation work, including architectural design of the overall SM study, an inventory check of all the previously developed simulation models, and selection of the integration approach to be used. The last step focuses on the development, integration and validation of the unit, subsystem, and overall MUHS simulation models.

In this chapter, we will use an example to demonstrate the application of the FMI approach with an emphasis on the last step, and a further focus on the application of the MMI approach. This is because the DMI approach has already been well covered by

many scholars (Geoffrion, 1989a; Dolk & Kottemann, 1993; Meckesheimer, 2001) and the HMI approach is a mixture of DMI and MMI.

In this application example, we will first start with a simple MUHS consisting of only two units, for each of which a DES model has already been developed. We will demonstrate how the MMI approach can be used to integrate them to address decision problems concerning the two units. For comparison purpose, two integrated simulation models developed using the alternative DMI and HMI approaches are also presented. After the two-unit MUHS example, this chapter discusses the possibility of scaling up this example to integrate additional unit simulation models, e.g. CHR surgery and primary care simulation models.

8.2 Background Information

The MUHS chosen for this application example includes a portion of the CHR healthcare system described in Chapter 6, this being the following two units: 1) the FMC ED; and 2) the CHR HW. The high-level patient flows within the FMC ED and CHR HW are shown in Figure 8-1.

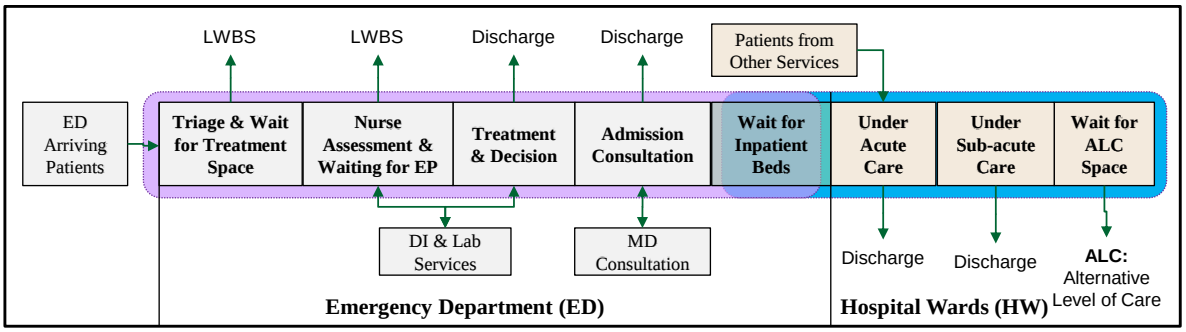


Figure 8-1 High-level Patient Flows Within the FMC ED and the CHR HW

The patient-flow within the FMC ED has already been discussed in Section 7.2 of Chapter 7, so only the patient flow within the CHR HW needs to be described here:

- Admitted patients from the FMC ED stay in the FMC ED until inpatient beds become available in the CHR HW.
- Patients transferred from other services to the CHR HW will wait for inpatient beds as well.
- Each patient in the CHR HW will go through acute care once an inpatient bed is accessed.
- After acute care, some patients are discharged and the occupied inpatient beds become available for new patients, while others move on to sub-acute care.
- After the sub-acute care, some patients are discharged releasing an inpatient bed for new patients, while others begin to wait for an alternative level of care (ALC) space (frequently long-term care).
- When an ALC space becomes available, the patient waiting for ALC leaves the CHR HW and the occupied inpatient bed becomes available for new patients.

As shown in Figure 8-1, the CHR HW provides downstream services for admitted patients from the FMC ED, who will stay in the FMC ED until inpatient beds become available in the CHR HW, though the FMC ED and the CHR HW are operated independently. In addition to accepting patients admitted from the FMC ED, the CHR HW also accepts patients from other services, e.g. surgery or outpatient clinics. The operations of the FMC ED and the CHR HW are interconnected since the boarding time of the admitted patients from the FMC ED depends on the operation of the CHR HW, and the operations of the

FMC ED affect the total number of patients arriving to the CHR HW. The longer the boarding time of the admitted patients in the FMC ED, the slower the patient flow via the FMC ED as the beds in the FMC ED cannot be freed up to accept new patients. In the meantime, the slower the patient flow via the FMC ED, the smaller the number of patients admitted to the CHR HW and the shorter the boarding time for admitted patients from the FMC ED. In other words, the patient arrival rate from the FMC ED to the CHR HW and the boarding time of admitted patients from the FMC ED are unit performance measures which impact each other.

There is a DES model, as introduced in Chapter 7, developed by the author in 2009 as part of a research project in which the author participated (Rohleder et al., 2009), to simulate the operations of the FMC ED. In the same research project, Drs. Paul Rogers and Tom Rohleder developed a DES model to simulate the operations of the hospitalist service in the FMC. Based on that DES model, a DES model was developed to simulate the operations of the CHR HW.

In the FMC ED DES model, the boarding time of admitted ED patients is modeled via probability distributions derived from historical data. In the CHR HW DES model, the arrival rate of the ED patients to the CHR HW is also modeled as a probability distribution derived from historical data. By integrating these two DES models, it can provide a holistic view of the operations of both units, explicitly accounting for the dependency between these two parameters of the two-unit MUHS.

8.3 Development of the Integrated FMC ED & CHR HW Simulation Model Using the MMI Approach

This section describes the process for developing the integrated FMC ED and CHR HW simulation model using the MMI approach. The process is composed of the following four steps:

1) Perform architectural and detailed design of the integrated FMC ED & CHR HW simulation model.
2) Develop simulation metamodels to represent the FMC ED and the CHR HW.
3) Develop the integrated FMC ED & CHR HW simulation model.
4) Verify the valid range of the integrated FMC ED & CHR HW simulation model.

Details on each step are explained in the following subsections.

8.3.1 *Architectural and Detailed Design of the MUHS Simulation Model*

As in the development of any simulation models, the first step in the development of the integrated FMC ED & CHR HW simulation model is to determine the modeling objective.

In this application example, we are particularly interested in understanding the impact of the arrival rate of patients to the FMC ED on the LoS of the admitted patients from the FMC ED to the CHR HW considering the interactions between the FMC ED and the CHR HW. Accordingly, the modeling objective is to explicitly model the interactions between the FMC ED and the CHR HW in addition to the operations of FMC ED.

After the modeling objective has been determined, we can start the architectural design of the integrated FMC ED & CHR HW simulation model. The objective of the architectural design is to design the architecture of the integrated FMC ED & CHR simulation model, outlining the major modules, and the inputs, outputs, and interfaces between the modules. The designed architecture should best support the achieving of the modeling objectives. The designed architecture of the integrated FMC ED & CHR HW simulation model is shown in Figure 8-2. It includes two modules:

- The FMC ED module representing the operations of the FMC ED, and
- The CHR HW module representing the operations of the CHR HW.

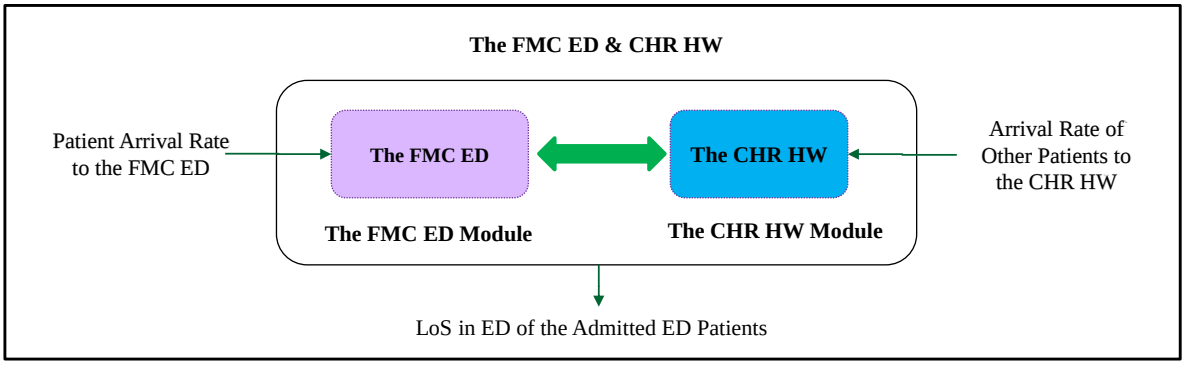


Figure 8-2 Architecture of the Integrated FMC ED & CHR HW Simulation Model

The integrated simulation model has two external inputs:

- Patient arrival rate to the FMC ED, and
- Arrival rate of other patients to the CHR HW.

and one output:

- LoS in ED of admitted ED patients.

The integrated model has one inter-module interface. It is bidirectional with arrows at both ends of the connection meaning there is information exchanged in both directions. The information exchanged over this interface includes:

- Arrival rate of admitted ED patients to the CHR HW (from the FMC ED module to the CHR HW module), and
- Boarding time of admitted ED patients (from the CHR HW module to the FMC ED module).

After the architectural design, a detailed design needs to be performed for the integrated FMC ED & CHR HW simulation model. The detailed design defines:

- The scope of the NN-based simulation metamodels to be developed, including the inputs and outputs of each simulation metamodel, and
- The interconnections between the simulation metamodels.

The detailed design of the integrated FMC ED & CHR HW simulation model is shown in Figure 8-3.

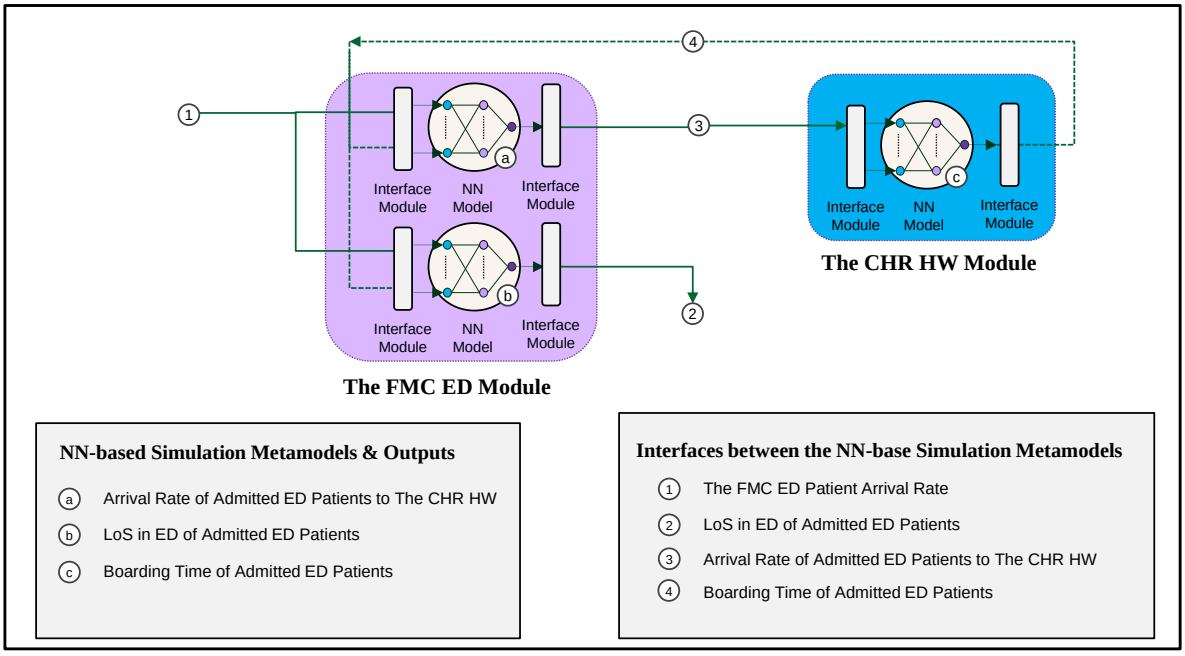


Figure 8-3 Detailed Design of the Integrated FMC ED & CHR HW Simulation Model

As shown in Figure 8-3, we need one NN-based simulation metamodel in each module to handle the inter-module interactions. In addition, we also need another NN-based simulation metamodel in the FMC ED module to calculate the LoS of admitted ED patients. In the architectural design, the arrival rate of other patients to the CHR HW is listed as an input. But in this application example, we assume that this arrival rate will stay the same so it is eliminated in the detailed design. As a result, only four interfaces are listed in Figure 8-3, including:

- Two external interfaces
    - The FMC ED patient arrival rate, and
    - LoS in ED of admitted ED patients.
- Two internal interfaces:

    - Arrival rate of admitted ED patients to the CHR HW, and
    - Boarding time of admitted ED patients.

In Figure 8-3, there is a pair of interface modules associated with each NN model to handle the translation of data format to/from the NN model. Details on the three NN-based simulation metamodels and their interfaces are provided in next subsection.

8.3.2 *Development of the NN-based Simulation Metamodels*

In the detailed design of the integrated FMC ED & CHR HW simulation model, two NN-based simulation metamodels are employed in the FMC ED module. One NN-based simulation metamodel is to handle the interactions with the CHR HW module and it has the following interfaces:

- Two for inputs:
    - Patient arrival rate to the FMC ED, and
    - Boarding time of admitted ED patients.
- One for output:
    - Arrival rate of admitted ED patients to the CHR HW, which is equal to the rate of admitted ED patients in the FMC ED.

The other NN-based simulation metamodel is to calculate the LoS in the ED of admitted ED patients and it has the following interfaces:

- Two for inputs (the same as for the other NN-based simulation metamodel):
    - Patient arrival rate to the FMC ED, and

    - Boarding time of admitted ED patients.
- One for output:
    - LoS in ED of admitted ED patients.

There is only one NN-based simulation metamodel employed in the CHR HW module to handle the interactions with the FMC ED. It has the following interfaces:

- One for input:
    - Arrival rate of admitted ED patients to the CHR HW.
- One for output:
    - Boarding time of admitted ED patients.

The three NN-based simulation metamodels are developed following the same procedure as described in Chapter 7. Details on the three NN-based simulation metamodels are summarized in Appendix E.

8.3.3 *Development of the Integrated FMC ED & CHR HW Simulation Model*

After the development of the three NN-based simulation metamodels, we need to integrate them together to develop the integrated FMC ED & CHR HW simulation model. At this stage, we need to make a decision on which SM method and software package is to be used. The NN models are deterministic and do not require any special software package to be implemented, however, in order to simulate the interactions between two NN models, we need a software package, which can support dynamic simulation.

In this application example, ARENA® 12.0 is used for the development of the integrated FMC ED and CHR HW simulation model using the DES method. The only reason why DES method and ARENA® 12.0 are chosen is because they were used in the development of the original FMC ED and CHR HW DES models and we also want to build other versions of the integrated FMC ED & CHR HW simulation model using DMI and HMI approaches.

The integrated FMC ED and CHR HW DES model is a very simple model with its flow chart shown in Figure 8-4. In the model, parameters of the three NN models are stored in 2-D array variables.

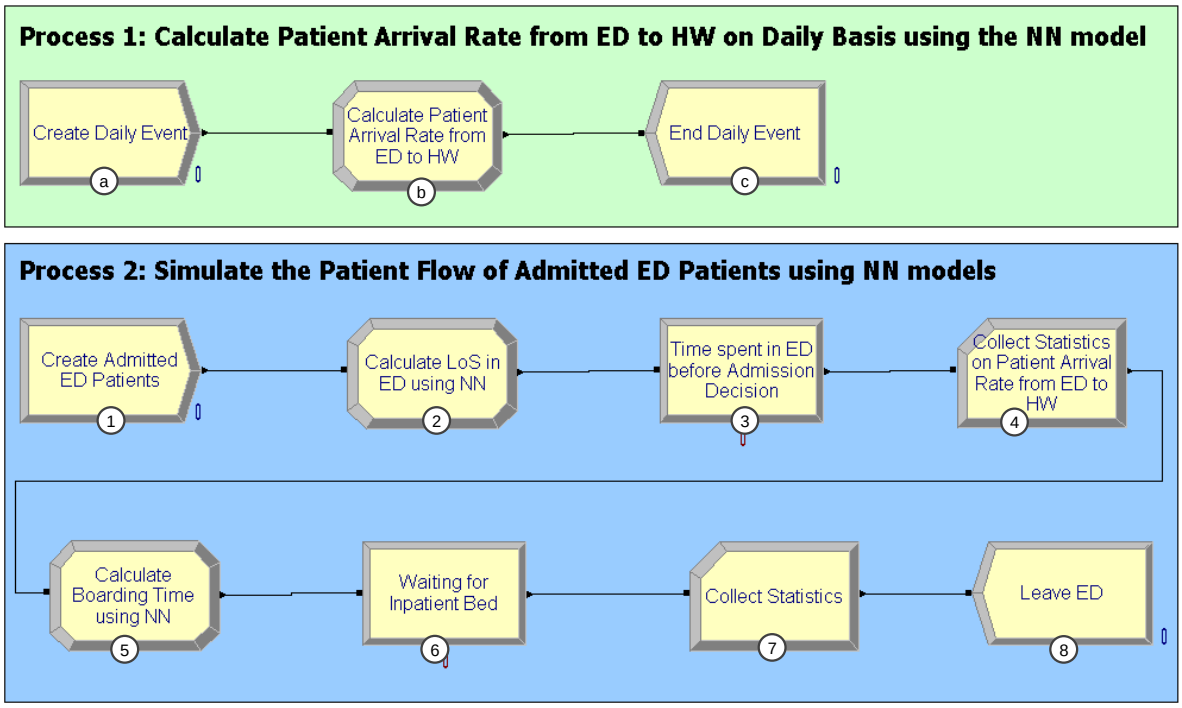


Figure 8-4 Flow Chart of the Integrated FMC ED & CHR HW DES Model

As indicated in Figure 8-4, two processes are used in the integrated FMC ED & CHR HW DES model. Process 1 is used to calculate the patient arrival rate of admitted ED patients to the CHR HW with the following three steps:

Step a: An event entity is created with a constant interval of a day at the *Create Daily Event* module.

Step b: This event entity moves to the *Assign* module, *Calculate Patient Arrival Rate from ED to HW*, and triggers the calculation of the arrival rate of admitted ED patients to the CHR HW using the patient arrival rate to the FMC ED and the boarding time of admitted ED patients as inputs. Both the inputs and output are stored in *variables*.

Step c: The event entity moves to the *Dispose* module, *End Daily Event* module, and is disposed.

Process 2 is to simulate the patient flow of admitted ED patients based on the NN metamodels. This process includes 8 steps and includes some randomness.

Step 1: At the *Create Admitted ED Patients* module, admitted ED patients are created following a Poisson process using the calculated *arrival rate of admitted ED patients to the CHR HW* in process 1 as the mean.

Step 2: The patient entity moves to the *Assign* module, *Calculate LoS in ED using* NN, and triggers the calculation of this patient entity's LoS in the FMC ED.

Step 3: The patient entity moves to the *Delay* module, *Time Spent in the ED before Admission Decision*, and is delayed for a period equal to the calculated LoS in the ED minus the current boarding time of admitted ED patients.

Step 4: The patient entity moves to the *Statistic* module, *Collect Patient Arrival Rate from ED to HW*. This module collects statistics on the current patient arrival rate from the FMC ED to the CHR HW.

Step 5: The patient entity moves to another *Assign* module, *Calculate Boarding Time using NN*, and triggers the calculation of the boarding time of the patient entity using the current patient arrival rate of admitted ED patients measured in Step 4.

Step 6: The patient entity moves to another *Delay* module, *Waiting for Inpatient Bed*, and waits for a period equal to the calculated boarding time in Step 5.

Step 7: The patient entity moves to another *Statistic* module, which collects statistics on LoS in ED of the admitted ED patients.

Step 8: The patient entity moves to a *Dispose* module and is disposed.

8.3.4 *Valid Range of the Integrated Simulation Model*

As discussed in Section 5.3 of Chapter 5, simulation metamodels are only valid to represent the original simulation model within the domain of the generated experimental results. When developing integrated simulation models using the MMI or HMI approaches, it is important to understand the valid range of the integrated simulation

model, defined as the range within which the inputs of all the simulation metamodels included are valid.

The valid ranges of the three NN-based simulation metamodels can be found in Appendix E and are summarize in Table 8-1.

Table 8-1 Valid Ranges of the FMC ED & CHR HW Simulation Metamodels

Simulation Metamodels	Input Parameters	Lower Limit	Upper Limit
The FMC ED Simulation Metamodels	Arrival Rate of ED Patients	80.0%	150.0%
	Boarding Time of Admitted ED Patients	60.0%	160.0%
The CHR HW Simulation Metamodel	Arrival Rate of Admitted ED Patients	70.0%	160.0%

There is only one input to the integrated FMC ED & CHR HW DES model: arrival rate of ED patients. This input is also an input of the two FMC ED NN-based simulation metamodels and its valid range is from 80% to 150%. However, based on this information, we cannot conclude that the valid range of the integrated FMC ED & CHR HW DES model is also from 80% to 150% because the CHR HW simulation metamodel interacts with one of the FMC ED simulation metamodels creating a circle: one input of one simulation metamodel is the output of another simulation metamodel and vice versa.

In order to determine the valid range of the integrated FMC ED & CHR HW DES model, a trial-and-error approach is used. First, the integrated DES model is run with a selected

range of test input. In this example, we choose the test range between 80% and 150% as this is the valid range for the FMC ED NN-based simulation metamodels. The following fifteen test inputs are used to run the integrated FMC ED & CHR HW DES model: 80% to 150% with a step of 5%. For each run, the inputs to all the simulation metamodels are collected and compared with their valid ranges. The experimental results are shown in Table 8-2.

Table 8-2 Experimental Results of the Integrated FMC ED & CHR HW DES Model (MMI)

Test Input	Arrival Rate of ED Patients	Boarding Time of Admitted ED Patients	Arrival Rate of Admitted ED Patients	Valid?
1	80%	27.3%	85.2%	No
2	85%	50.0%	88.3%	No
3	90%	72.0%	92.7%	Yes
4	95%	89.1%	97.2%	Yes
5	100%	100.0%	100.0%	Yes
6	105%	106.5%	101.2%	Yes
7	110%	110.3%	103.8%	Yes
8	115%	112.6%	104.1%	Yes
9	120%	115.1%	103.1%	Yes
10	125%	117.0%	104.8%	Yes
11	130%	117.8%	102.1%	Yes
12	135%	117.7%	105.6%	Yes
13	140%	117.7%	106.3%	Yes
14	145%	117.6%	104.7%	Yes
15	150%	118.3%	103.5%	Yes

The experimental results in Table 8-2 are examined for each test input to assess whether it is within the valid range of the integrated FMC ED & CHR HW DES model. For example, for the first test input, the two generated inputs of the three NN-based simulation metamodels are:

- For the two FMC ED NN-based simulation metamodels:

    - Patient arrival rate: 80%, which is within its valid range of (80%, 150%), and
    - Boarding time of admitted ED patients: 27.3%, which is outside of its valid range of (70%, 160%).
- For the CHR HW NN-bade simulation metamodel:
    - Arrival rate of admitted ED patients: 85.2%, which is within its valid range of (60%, 160%).

So, the integrated FMC ED & CHR HW DES model is not valid for the first test input.

The assessment results for the 15 test inputs are listed on the last column in Table 8-2. Based on the results, the estimated valid range of the integrated FMC ED & CHR HW DES model is from 90% to 150%. If the valid range of the integrated FMC ED & CHR HW DES model is deemed too narrow, we will need to go back to increase the valid ranges of some of the NN-based simulation metamodels by generating more experimental data. In the example here, if we want to make the lower limit of the valid range less than 90%, we need to rerun the original FMC ED DES model to generate experimental results with boarding time less than 60%.

8.4 Development of the Integrated FMC ED & CHR HW DES Models Using the DMI and HMI Approaches

For comparison purpose, two additional integrated FMC ED & CHR HW DES models are developed using the DMI and HMI approach. These two DES models are also developed

using ARENA® 12.0. In the following subsections, a high level summary is provided of each model.

8.4.1 *The DMI Approach*

With the DMI approach, the CHR HW DES model is used as an add-on module in the FMC ED DES model to replace the *Delay* module representing the boarding time of an admitted patient from the FMC ED. The entity representing an admitted ED patient is routed to the CHR HW module and waits for an inpatient bed. When an inpatient bed becomes available, the admitted ED patient entity is released from the occupied ED bed and moves to the acute care step in the CHR HW.

Though the DMI approach seems to be very straightforward, its implementation is far more than just plug-and play. There is much tedious and detailed work to be done in the integration process compared with the MMI approach. Some major examples are listed as follows:

Name conflicts: These two DES models were developed separately and there are many conflicts on the names used for variables, modules, tallies, etc. Thus, changes are required to avoid the conflicts. In this application example, we keep the FMC ED DES model unchanged as there are more variables and modules used in this DES model, and changes are all made to the CHR HW DES model.

CRN (Common Random Numbers) planning: Similar to the name conflicts mentioned above, changes have been made to the CHR HW DES model to reassign some of the random number streams.

Master version of the simulation models: As changes have been made to the CHR HW DES model, we need to decide which version of the model should be treated as the master version of the CHR HW DES model: the modified one or the original one. In this application example, the modified version is treated as the master version. Any changes to the CHR HW module in the integrated FMC ED & CHR HW DES model are duplicated in the original CHR HW DES model. So basically each change has to be implemented twice.

Experimental run time: With the same precision requirement, it takes much more time to run the integrated model than the FMC ED DES model and the CHR HW DES model. This is a painful and time consuming process for the verification of the integrated model.

8.4.2 *The HMI Approach*

With the HMI approach, we used the original FMC ED DES model and the NN metamodel developed using the experimental results of the CHR HW DES model. The NN metamodel uses the arrival rate of the ED patients as input and feeds the calculated boarding time of admitted ED patients to the FMC ED DES model.

In the integrated FMC ED & CHR HW DES developed using the HMI approach, only the CHR HW NN-based simulation metamodel is used and its valid range is shown in Table 8-1. The integrated FMC ED & CHR HW DES model is run for ten 485-day replications for the following 19 test inputs: 60% to 150% with a step of 5%. The experimental results are shown in Table 8-3.

Table 8-3 Experimental Results of the Integrated FMC ED & CHR HW DES Model (HMI)

Test Input	Arrival Rate of ED Patients	Arrival Rate of Admitted ED Patients	Valid?
1	60%	63.7%	No
2	65%	68.5%	No
3	70%	75.7%	Yes
4	75%	78.8%	Yes
5	80%	82.4%	Yes
6	85%	90.8%	Yes
7	90%	94.2%	Yes
8	95%	95.8%	Yes
9	100%	100.0%	Yes
10	105%	99.6%	Yes
11	110%	100.8%	Yes
12	115%	101.4%	Yes
13	120%	103.7%	Yes
14	125%	105.4%	Yes
15	130%	105.1%	Yes
16	135%	102.8%	Yes
17	140%	105.2%	Yes
18	145%	104.3%	Yes
19	150%	103.9%	Yes

The experimental results are examined for each test input as discussed in Subsection 8.3.4 to assess whether it is within the valid range of the integrated FMC ED & CHR HW DES model. The assessment results for the 19 test inputs are listed on the last column in Table 8-3. Based on the results, the estimated valid range of the integrated FMC ED & CHR HW DES model is from 70% to 150%.

8.5 Comparison Between DMI, MMI and HMI

8.5.1 *Minimum Number of Replications of the Integrated Simulation Models*

After the three integrated FMC ED & CHR HW DES models have been developed, there is one piece of preparation work to be done before we can proceed to compare their outputs and performances, this being to determine the minimum numbers of replications to meet a certain specified precision in the experimental results.

The precision of a simulation model is determined by the number and length of the replications used to generate the experimental results. In order to have a fair comparison between the outputs and performance of the three integrated FMC ED & CHR HW DES models, we need to first make sure that the same specified precision of the experimental results can be achieved with the proper number and length of the replications. In this application example, the following two outputs of the integrated FMC ED & CHR HW DES models are chosen with their specified precision shown below as well:

- Average LoS of admitted ED patients from the FMC ED:
    - MAD = 5 minutes
- Boarding time of admitted ED patients from the FMC ED:
    - MAD = 5 minutes

The procedure described in Appendix C is used to determine the minimum number of replications to meet the specified precision listed above. A replication length of 485 days including a 120-day warm up period is chosen. Each integrated FMC ED & CHR HW

DES model is run with 10 replications. The calculated minimum numbers of replications for the three integrated FMC ED & CHR HW DES models are shown in Table 8-4.

Table 8-4 Minimum Numbers of Replications with a 485-day Replication Length

Simulation Model Outputs	Unit	Replication, r										$\bar{Y}$	S_0	Target ϵ		$(Z_{0.025}S_0/\epsilon)^2$	Min R	$(t_{0.025,R-1}S_0/\epsilon)^2$
		1	2	3	4	5	6	7	8	9	10			% of $\bar{Y}$	Value			
DMI_LoS Admitted	Minute	743	739	736	740	744	743	739	721	739	741	739	6.50	0.7%	5.0	6.50	10	8.7
DMI_Boarding Time	Minute	208	195	204	202	196	210	206	188	205	218	203	8.37	2.5%	5.0	10.77	14	13.1
MMI_LoS Admitted	Minute	740	737	745	743	749	747	735	737	747	730	741	6.22	0.7%	5.0	5.95	9	8.2
MMI_Boarding Time	Minute	207	206	209	208	214	213	206	203	213	200	208	4.66	2.4%	5.0	3.34	6	5.7
HMI_LoS Admitted	Minute	741	749	742	741	746	747	733	746	729	734	741	6.65	0.7%	5.0	6.80	10	9.1
HMI_Boarding Time	Minute	212	211	212	207	202	213	196	208	199	209	207	6.01	2.4%	5.0	5.55	8	8.1

From Table 8-4, we can see that the minimum numbers of replications to meet the 5-minute MAD for the integrated FMC ED & CHR HW DES models using different integration approaches are:

- DMI approach: 14
- MMI approach: 9
- HMI approach: 10

8.5.2 *Experimental Run Times of the Integrated Simulation Models*

In this subsection, the experimental run times of the three integrated FMC ED & CHR HW DES models developed using the DMI, MMI and HMI approaches are compared by generating experimental results with the same specified precision and replication parameters as discussed in Subsection 8.5.1.

ARENA® 12.0 is used for generating the experimental runs of the three integrated simulation models for the baseline scenario, which represents the operations of the FMC ED and the CHR HW in 2009. ARENA® 12.0 is run on a Lenovo (IBM) ThinkPad T60p with the following specifications

- Processor: Intel Core Duo processor T7200 (2GHz)
- Memory: 512 MB
- Operating system: Windows Vista Business 32-bit

The experimental run times of the three integrated FMC ED & CHR HW models are summarized in Table 8-5.

Table 8-5 Experimental Run Times

Integration Approach	Replication Length (Day)	Minimum Number of Replications	Time for Each Replication (Minute)	Time for Each Experimental Run (Minute)
DMI	485	14	3.95	55.3
MMI	485	9	0.18	1.62
HMI	485	10	3.75	37.5

The results in Table 8-5 show that it takes 55.3 minutes for each experimental run of the directly integrated FMC ED & CHR HW DES model. By using the HMI and MMI approaches, the time for each experimental run can be reduced by 32.2% and 97.1% to 37.5 minutes and 1.62 minutes, respectively.

8.5.3 *Output Comparison of the Integrated FMC ED & CHR HW Simulation Models*

In this subsection, we compare the outputs from the experimental runs of the three integrated FMC ED & CHR HW DES models. In the comparison, the results for the DMI approach are used as a reference since the DMI approach directly integrates the unit simulation models and does not introduce errors. This is to be compared with the other integration approaches (HMI and MMI) which do introduce additional errors due to the development of NN-based simulation metamodels from the experimental results of the individual simulation models. In practice, it is very rare to develop integrated simulation models using multiple integration approaches in parallel. The primary objective of this subsection is to examine the possible impact of the errors introduced in the simulation metamodeling process on the performance of the integrated simulation models developed using the HMI and MMI approaches.

As discussed in subsections 8.3.4 and 8.4.2, the NN-based simulation metamodels used in the integrated FMC ED & CHR HW DES models developed using the MMI and HMI approaches are valid within the following ranges for the ED patient arrival rate:

- MMI: 90% to 150%
- HMI: 70% to 150%

Thus, the narrower range, (90%, 150%), is chosen to ensure the NN-based simulation models used in both integrated FMC ED & CHR HW DES models are valid. The three integrated FMC ED & CHR HW DES models are run for the following thirteen ED patient arrival rates: 90% to 150% with a step of 5%. The replication parameters used in

generating the experimental runs of the three integrated DES models are those determined in Subsection 8.5.1. Statistics on the following two outputs are collected for each experimental run:

- LoS in ED of admitted ED patients, and
- Boarding time of admitted ED patients.

LoS in ED of admitted ED patients

The LoS in ED of admitted ED patients collected from the experimental runs of each integrated FMC ED & CHR HW DES model are shown in Figure 8-5. The vertical axis in this figure represents the percentage deviations of the results for the MMI and HMI approaches from the results for the DMI approach (shown as the reference line). The horizontal axis represents the ED patient arrival rate, which ranges from 90% to 150%.

From Figure 8-5, we can see that:

- The percentage deviation of the LoS of the admitted ED patients generated from running the FMC ED & CHR HW DES model developed using the HMI approach is between -2% and 0%, while the MAPD of all data points is 1.14%.
- The percentage deviation of the LoS of the admitted ED patients generated from running the FMC ED & CHR HW DES model developed using the MMI approach is between -3.0% and 2.5%, while the MAPD of all data points is 2.17%.

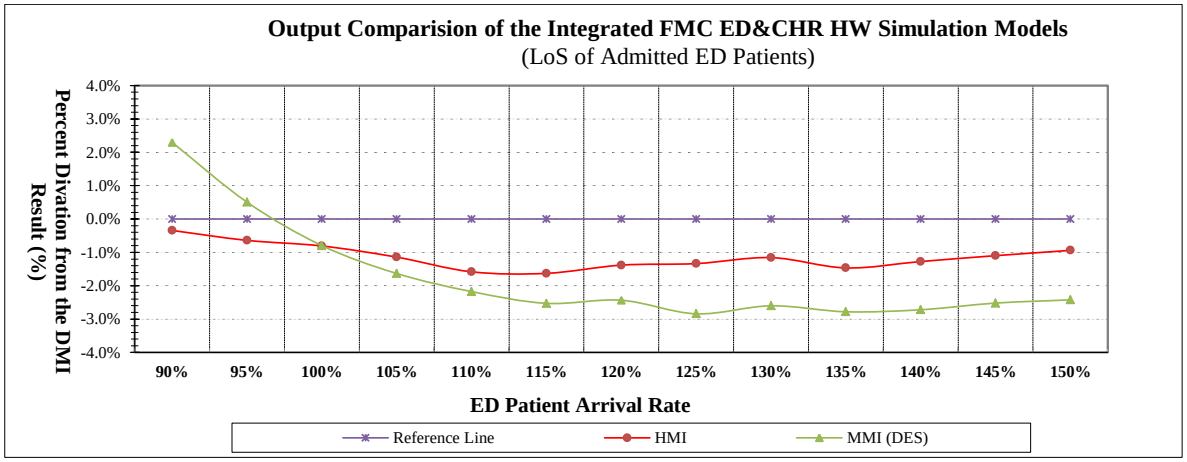


Figure 8-5 LoS in ED of Admitted ED Patients

Boarding Time of Admitted ED patients

The boarding time of admitted ED patients generated from the experimental run of each integrated FMC ED & CHR HW DES model is shown in Figure 8-6. Like those in Figure 8-5, the vertical axis represents the percentage deviations of the results for the MMI and HMI approaches from those for the DMI approach (shown as the reference line). The horizontal axis represents the ED patient arrival rate, which ranges from 90% to 150%.

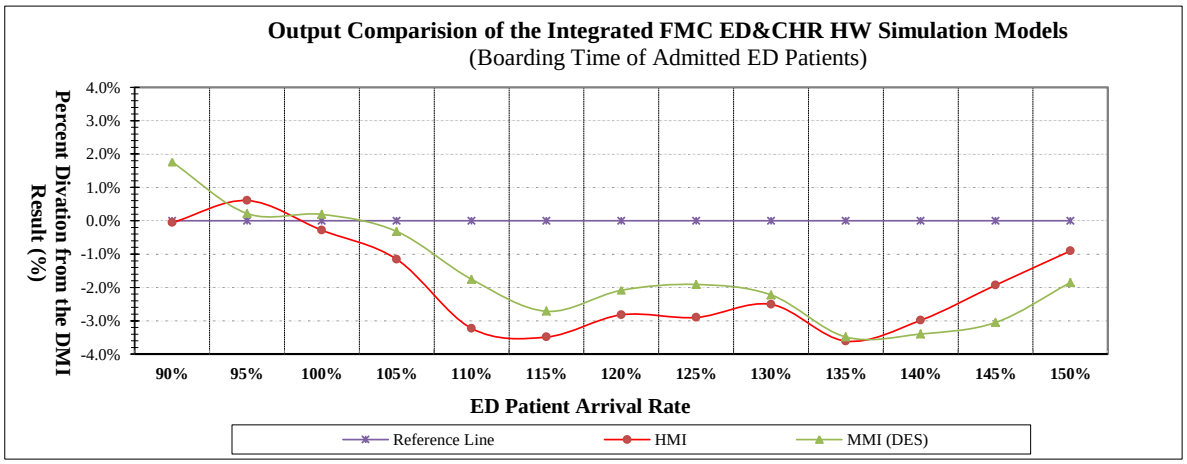


Figure 8-6 Boarding Time of Admitted ED Patients

From Figure 8-6, we can see that:

- The percentage deviation of the boarding time of the admitted ED patients generated from running the FMC ED & CHR HW DES model developed using the HMI approach is between -4% and 1%, while the MAPD of all data points is 2.04%.
- The percentage deviation of the LoS of the admitted ED patients generated from running the FMC ED & CHR HW DES model developed using the MMI approach is between -4.0% and 2%, while the MAPD of all data points is 1.92%.

As indicated in Appendix E, the precisions of both the FMC ED and CHR HW NN models are MAPD 2.5%, which can be met by the MAPDs of all the data points. However, the percentage deviations at certain data points go beyond the 2.5% MAPD and this can be due to the following reasons, which cause errors for the HMI and MMI approaches:

1) Mismatch of information on the interface between a NN model and another NN model or simulation model. For example, when the CHR HW DES model is used to generate experimental results for training the CHR HW NN model, probability distribution is used for generating the admitted ED patient entities. Once the CHR HW NN model is integrated with FMD ED DES model or NN model, the arriving of the admitted ED patients may not follow exactly the same distribution.
2) When multiple NN models are linked together, error propagation can happen so the error introduced by a NN model can be amplified by a sequent NN model. This

can be used to explain why for the predicted LoS of admitted ED patients, the performance for the MMI approach is worse than that for the HMI approach.

3) NN models are deterministic. When it is connected to a simulation model, it can impact the effect of the randomness in the simulation model. This may be able to explain why for the predicted boarding time of admitted ED patients, the performance for the HMI approach is actually slightly worse than that for the MMI approach.

Errors introduced in the simulation metamodeling process can affect the validity of the integrated simulation models developed using HMI or MMI approaches to represent the integrated simulation models developed using the DMI approach. Improving the accuracy of each individual NN models can help to improve the overall accuracy for the HMI or MMI. However, solely focusing on improving the validity for the HMI or MMI approach using the results for the DMI approach as references can be misleading as the ultimate objective for developing the integrated simulation models of the MUHS using HMI or MMI approach is to simulate the operations of the MUHS. The validation of the individual NN-based simulation metamodel should focus more on the trend of the input-output relationships rather than the absolute values and the validation of the integrated simulation models developed using MMI or HMI approaches should be done using the system level data collected from the real system.

8.6 Integrating Three or More Unit Simulation Models Using the MMI Approach

In the previous sections, we have demonstrated the development of the MUHS simulation model by integrating two unit simulation models using the MMI approach. In this section, we will further demonstrate how the MMI approach can be used for developing integrated simulation models for MUHSs with three or more units.

Building upon the example discussed earlier, let's assume the modeling objective has been revised as follows:

We want to understand the impact of the population growth in the CHR on the performances of the FMC ED considering the following factors:

- *Interactions between the FMC ED and the CHR HW,*
- *The increased patient volume from the CHR surgery operations to the CHR HW, and*
- *The increased patient volume from the CHR primary care to the FMC ED.*

Compared with the modeling objective described in Section 8.3, this modeling objective has a larger scope thus requiring an integrated simulation model to represent a larger and more complex MUHS.

According to this new modeling objective, the architecture of the new MUHS simulation model is designed with the result shown in Figure 8-7.

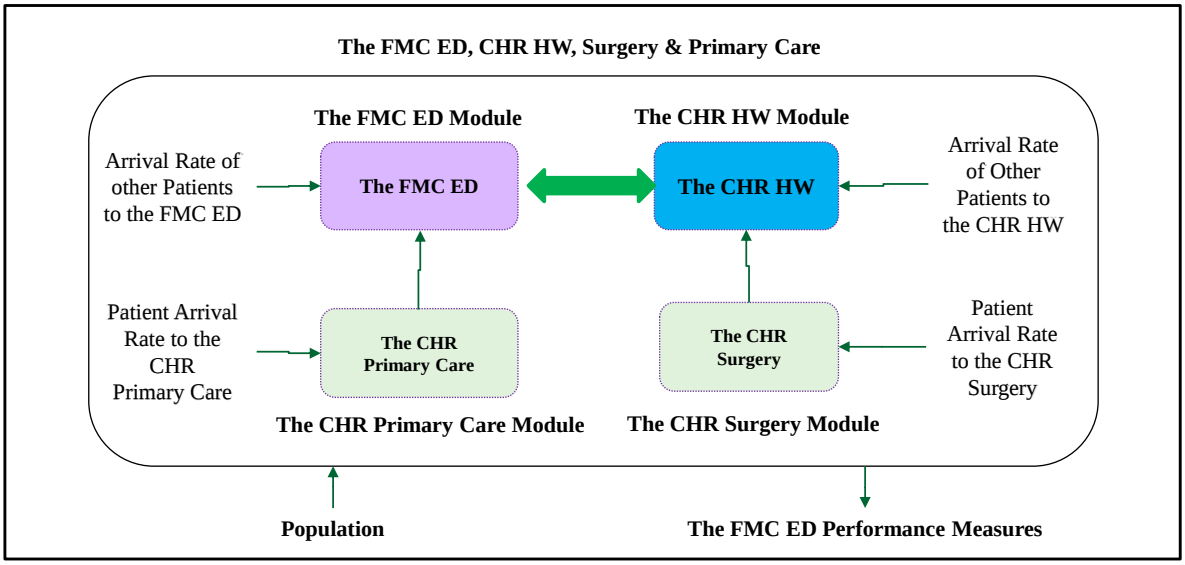


Figure 8-7 Architecture of the Integrated MUHS Simulation Model

As shown in Figure 8-7, the integrated MUHS simulation model includes four modules:

- The FMC ED module representing the operations of the FMC ED,
- The CHR HW module representing the operations of the CHR HW,
- The CHR surgery module representing the CHR surgery operations, and
- The CHR primary care module representing the operations of the CHR primary care services.

The integrated MUHS simulation model has one input:

- Population

and one or more outputs depending on the requirements of the decision-makers:

- Waiting times of admitted, discharged and all ED patients in the waiting room,
- LoS in ED of admitted, discharged and all ED patients,

- Percent of ED patients choosing LWBS in the waiting room,
- etc.

The integrated MUHS simulation model has three inter-module interfaces:

- A bidirectional interface between the FMC ED module and CHR HW module,
- A unidirectional interface between the CHR surgery module and the CHR HW module, and
- A unidirectional interface between the CHR primary care module and the FMC ED module.

In addition, the MUHS also requires a global interface module to translate the population size into the following arrival rates to different units:

- Patient Arrival Rate to CHR Primary Care,
- Patient Arrival Rate to CHR Surgery,
- Arrival Rate of other Patients to the FMC ED, and
- Arrival Rate of Other Patients to the CHR HW.

Based on the architecture of the MUHS simulation model, a detailed design is performed and the result is shown in Figure 8-8.

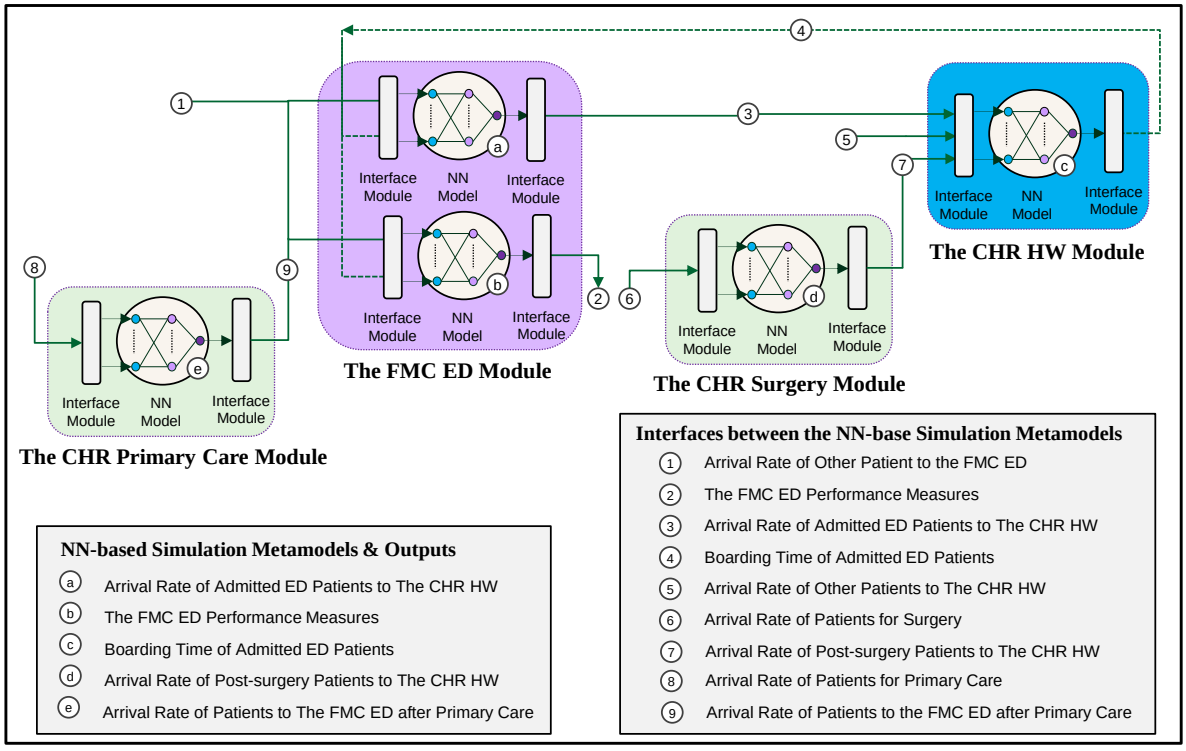


Figure 8-8 Detailed Design of the Integrated Simulation Model

As listed in Figure 8-8, the following NN-based simulation metamodels are required:

- One NN model in the CHR HW module to handle the interaction with the FMC ED module,
- One NN model in the CHR surgery module to generate inputs to the CHR HW module,
- One NN model in the CHR primary care module to generate inputs to the FMC ED module,
- One NN model in the FMC ED module to handle the interaction with the CHR HW module, and

- Multiple NN models in the FMC ED module to calculate different ED performance measures. It is also possible to use a NN model with multiple outputs (with a more complex architecture).

The interfaces listed in Figure 8-8 include:

- Four external interfaces as inputs:
    - Patient Arrival Rate to the CHR Primary Care,
    - Patient Arrival Rate to the CHR Surgery,
    - Arrival Rate of other Patients to the FMC ED, and
    - Arrival Rate of Other Patients to the CHR HW.
- Several external interfaces as outputs:
    - The FMC ED Performance Measures (possibly multiple measures here).
- Four internal interfaces:
    - Arrival rate of admitted ED patients to the CHR HW,
    - Boarding time of admitted ED patients,
    - Arrival Rate of Post-surgery Patients to The CHR HW, and
    - Arrival Rate of Patients to the FMC ED after Primary Care.

Based on the detailed design, NN-based simulation metamodels can be developed following the same procedure as described in Chapter 7 and the integrated MUHS simulation model can be developed using the chosen SM method and SM software package as demonstrated in Subsection 8.3.3. The valid range of the integrated MUHS simulation model needs to be verified. Given the limited space in the dissertation, we will

not discuss the details on the development of the NN-based simulation metamodels and the integrated MUHS simulation model.

8.7 Discussion and Conclusions

In this chapter, an application example is used to demonstrate the application of the FMI approach in integrating unit simulation models with a focus on the MMI approach. The application example starts with a two-unit MUHS and then expands to include two more units, and provides some valuable insights on the application of the MMI approach. From the technical perspective, this example shows some advantages of the MMI approach when compared with the DMI and HMI approaches, including:

1) It eliminates most of the operational details modeled in the unit simulation models and minimizes the complexity of the integrated simulation models.
2) It significantly reduces the experimental run time of the integrated simulation models.
3) It eliminates the need to deal with the name conflicts, CRN reassignment and other required modifications to the unit simulation models.
4) The implementation of NN-based simulation metamodels does not require special software packages thus adding flexibility to the model integration process.

However, this example also helps to uncover several disadvantages of the MMI approach, including:

1) The NN-based simulation metamodels are only valid within the domain of the experimental results used to develop these simulation metamodels. The valid range of an integrated MUHS simulation model developed using the MMI approach needs to be further checked.
2) It introduces errors in the representation of the original simulation models, though these errors can be minimized by using appropriate simulation metamodeling techniques and paying special attention to the design of the experiments to run the unit simulation models.

These advantages and disadvantages also apply to the HMI approach.

However, the discussions in this chapter also point out that the application of the MMI or HMI approaches should not only focus on improving accuracy of the NN-based simulation metamodels using the DMI approach as reference. Instead, the validation of integrated simulation models, regardless whether they are developed using DMI, HMI or MMI approaches, should focus on whether they can properly represent the operations of the MUHS to be studied.

From the perspective of the development process of the NN-based simulation metamodels and the integrated MUHS simulation model, this application example also yields a number of useful findings including:

1) Proper upfront architectural design and detailed design are essential for the successful development of NN-based simulation metamodels and the integrated MUHS simulation model.

2) It is important to verify and become aware of the valid ranges of the integrated MUHS simulation models.
3) Even with the MMI approach, it can become a complex project for the development of the integrated MUHS simulation model. This indirectly proves the value that the system decomposition technique can bring to the AHSM approach.

Chapter 9 Research Summary, Conclusions and Future Directions

Much has been written in the healthcare literature on the urgency of improving the operational efficiency of healthcare organizations and the challenges in decision-making for managing the operations of healthcare systems, especially MUHSs. In order to deal with various decision problems in MUHSs, healthcare managers are in need of decision support tools which can help them model decision problems effectively and analyze them efficiently. SM is one of the most commonly used decision support tools in healthcare, with applications spanning over four decades, and it is regarded by many as the "technique of choice" to support decision-making in complex and fast changing healthcare environments. However, SM has not yet become an integral part of analysis before major decisions in healthcare are made. Further, most of the SM studies in healthcare are unit-specific and there is a lack of simulation models of healthcare systems consisting of multiple units. There are many contributing factors to the current status of SM applications in healthcare, amongst which are the following two that the author believes are the most important: rapidity and flexibility. Rapidity means how quickly a simulation model can be developed to address a decision problem while flexibility means the reusability of existing simulation models to address new decision problems. Current SM practices focusing on unit-specific applications lack the required rapidity and flexibility to effectively support decision-making in MUHSs.

In order to improve the rapidity and flexibility of SM applications in healthcare, one possible solution is to separate the development and maintenance of the unit simulation models from the use of these simulation models. Simulation models can be developed

and maintained at the unit level, and when a new decision problem occurs, these unit simulation models can be integrated to build an integrated simulation model of the overall MUHS within which the decision problem occurs.

Conceptually, the proposed solution sounds simple and straightforward, but there are two major challenges to operationalize this solution. The first is how to map the real-world decision problem into the virtual world of models and decide which unit simulation models are to be developed or reused? The second is once the unit simulation models have been identified, how can they be efficiently integrated together to represent the real-world decision problem?

This research is motivated to tackle these two challenges and it is divided into the following four tasks:

1) *Review related literature,*
2) *Develop a conceptual model to represent and describe the operations of MUHSs,*
3) *Develop a HSM approach to effectively support decision-making in MUHSs, and*
4) *Demonstrate the real-world potential of the proposed HSM approach via application examples.*

A summary of this research is provided in next section by briefly reviewing each of the above four research tasks. Discussion of and conclusions from this research are provided in Section 9.2. The novelty and major contributions of this research are discussed in Section 9.3, and the limitations of this work are summarized in Section 9.4. Finally, some possible directions for future research in this area are explored in Section 9.5.

9.1 Research Summary

9.1.1 *Literature Review*

Part 1: Complexity and Dynamics in Healthcare Systems and the CASs and ND Systems Theories

This part of the research provides a high-level review of the applications of complexity theories in the study of healthcare systems. It is agreed by most scholars that the whole system approach is the most appropriate approach to studying healthcare systems. Two frequently used theories of complex systems are reviewed: the CASs theory and the ND systems theory. The CASs theory focuses on the dynamics of complex systems and argues that the dynamics are due to the interactions between independent elements, which are defined as "adaptive agents". The CASs theory emphasizes that complex systems should be studied as a whole and they are not easily manageable as no one is "in charge". The ND systems theory focuses on the hierarchical structure of complex systems and argues that almost all complex systems exhibit hierarchical structures at the whole system level with the ND property: they can be divided into hierarchical layers of subsystems in such a way that interactions between elements belonging to same subsystem are much stronger than interactions between elements belonging to different subsystems. With the ND property, the study of a complex system can be simplified by subdividing the study of the whole system into studies of its subsystems combined with an aggregate study of the interactions between the subsystems.

Though their foci are different, the theories of CAS and ND systems agree on two fundamental features of complex systems: inter-element interactions and hierarchical structure at the whole system level. Both theories recognize the existence of inter-element interactions. The CASs theory defines the inter-element or inter-agent interactions as a process of learning from and adapting to each other. The ND systems theory assumes inter-element interactions as a natural existence, but insists that these inter-element interactions are not all of the same frequency and intensity. Regarding the hierarchical structure of complex systems, the CASs theory argues that hierarchical structure is a natural emergence through a self-organization process while the ND systems theory argues that the hierarchical structure with the ND property is the most probable structure to emerge via an "evolutionary process" because of the existence of intermediate stages.

Part 2: SM Methods and Their Applications in Healthcare

This part of the research first reviews the four social roles that simulation models can play in the decision-making process: 1) learn as a group; 2) express and experiment; 3) explore; and 4) predict. Understanding these social roles can help in choosing the appropriate SM method and in building simulation models. Next the research provides a comparison between the three basic SM methods commonly used in healthcare: DES, SD and ABS. Results from the comparison can help understand the advantages and disadvantages of each SM method, and thus guide the use of them in properly modeling different decision problems. The research also reviews the hybrid simulation and distributed simulation approaches, which are recommended by some scholars in

healthcare SM. The technical issues that affect the use of various SM methods in simulating MUHSs are also discussed.

At the end of this part of the research, a high-level literature review is provided on SM applications in healthcare with a focus on studying MUHSs. The main conclusions from the literature review can be summarized as follows.

- The majority of HSM studies uses DES,
- There is a lack of simulation models of healthcare systems consisting of multiple units, and
- Most of the healthcare simulation models are developed by academics for research and/or student projects, rather than consultants or management engineers doing routine analysis on the operations of healthcare systems. This helps to explain why the implementation of HSM results is rare.

9.1.2 *ND-CASs: A Conceptual Model to Represent and Describe MUHSs*

The review and comparison of the CASs theory and the ND systems theory lead to the introduction of the ND-CASs model to represent and describe the operations of MUHSs. The ND-CASs model integrates the key concepts from the theories of CASs and ND systems and serves as a conceptual tool to help smooth and speed up the SM-based decision-making process for managing the operations of MUHSs. A ND-CAS is composed of adaptive agents, which are intelligent and interact with each other and with the environment. The inter-agent interactions are diverse and of differing frequency and intensity. At the whole system level, a ND-CAS can exhibit multiple hierarchical

structures along different dimensions, though viewed along a particular dimension, there is only one stable hierarchical structure. The hierarchical structures have the ND property and can be designed or can emerge from the inter-agent interactions. The hierarchical structures of a ND-CAS can be maintained and modified via a co-evolutionary process amongst the agents, inter-agent interaction and the hierarchical structure. Compared with the changes in interactions between subsystems and agents at the lower layers, the changes of the hierarchical structures happen at a much slower pace.

Compared with the CASs theory or the ND systems theory, the ND-CASs model represents a more holistic view for the study of complex systems. With ND-CAS, the dynamics of complex systems can be represented and explained in terms of adaptive agents and their evolving interactions. The hierarchical structure at the whole system level, emerging via the co-evolutionary process, can help make the study of complex systems more manageable.

MUHSs can be described and represented as ND-CASs, with adaptive agents representing the units in the MUHS and the inter-agent interactions characterizing the interconnections between the units. Each unit in a MUHS is operated independently with its own operational goals. Inter-unit interactions are determined by the needs of both units to deliver their own services and they vary in frequency and intensity, depending on the closeness of the relationship between specific units. Using ND-CASs to represent MUHSs brings some benefits for the study of MUHSs including:

- First, the dynamics and complexity in the operations of MUHSs can be explained by the independent operations of each unit and their interactions. The inter-unit

interactions lead to the creation and adjustment of the hierarchical structures at the whole system level with layers of subsystems.

- These hierarchical structures have the ND property so that the study of MUHSs can be divided into the studies of each unit and subsystem, combined with aggregated studies on the inter-unit/subsystem interactions. This can make the study of MUHSs, especially those with a large number of units, more manageable.
- The ND-CASs model provides a simple representation of MUHSs focusing on both the independent operations of the units and the inter-unit interactions. Thus it can be used as a conceptual framework for the stakeholders involved in the decision-making process of MUHSs to develop a common understanding of the operations of MUHSs and relate the decision problems to the overall operations of the MUHSs as well as to the individual units and subsystems.
- Once identified, the hierarchical structures of MUHSs can be used in the architectural design of the MUHS simulation model, which is of vital importance for planning and coordinating different activities in the development of the MUHS simulation model. They can also be used as guidelines for the development of new unit simulation models or for the evaluation of the reusability of the existing unit simulation models.

9.1.3 *AHSM: A HSM Approach for Rapid and Flexible Decision-Making in MUHSs*

In this part of the research, an AHSM approach is proposed for developing MUHS simulation models in a rapid and flexible manner to effectively support decision-making in MUHSs. The development of AHSM is propelled by three driving forces: 1) the

complexity of decision problems; 2) the dynamics of the decision-making process; and 3) the limitations of current SM practices in healthcare. AHSM is based on a cyclical lifecycle approach introduced to describe the SM process for modeling decision problems, and is intended to support the full lifecycle of decision-making in MUHSs. It is a high-level SM approach to facilitate communications, coordination, and interactions between healthcare modelers, managers and other stakeholders in the decision-making process of MUHSs.

AHSM includes two stages. The first stage is called "*Planning*" and its main objective is to identify the hierarchical structures of a MUHS and perform an architectural design of the overall SM study. The second stage is called "*Executing*" and its main objective is to develop unit, subsystem and the overall MUHS simulation models.

To support the implementation of AHSM, the following three techniques/approaches are proposed in the research:

- A PCA-base system decomposition technique to identify the hierarchical structure of a MUHS with a large number of units.
- A NN-based simulation metamodeling technique to represent simulation models of the individual units in the MUHS using NN models.
- A flexible model integration approach for integrating the simulation models, or simulation metamodels of the units, to represent the subsystems and the overall MUHSs.

AHSM offers a number of benefits for the effective support of decision-making in MUHSs, such as:

- AHSM eliminates the need to develop a single complex simulation model to represent the overall MUHS.
- Unit/subsystem simulation models can be developed independently and in-parallel. This can speed up the model development process and greatly reduce the effort and time required to study MUHSs.
- Increases the reusability of existing simulation models regardless of the SM method and/or SM software package used to develop these simulation models.
- AHSM is robust to changes in MUHSs, so that changes in one or more units can be handled by adjusting the simulation models of these units independently. Changes at the whole system level, i.e. changes in the hierarchical structure, can be handled by rearranging the simulation models at the lower levels. This again improves the rapidity and flexibility of SM studies of MUHSs.

9.1.4 *AHSM Application Examples*

Three application examples are used to demonstrate the real-world potential of AHSM. The first example shows how to decompose the CHR healthcare system into hierarchical layers of subsystems. This application example shows that the proposed system decomposition technique follows a logical procedure and is easy to use. The only subjective portions in this approach are the definition of units, the measures of the inter-unit interactions and the number of subsystems to be retained at each decomposition step. Once these are decided, whoever does the decomposition will get the same result. This

makes the decomposition result acceptable for all the stakeholders involved in the decision-making process so that they can have the same starting point on how to deal with the decision problems.

The second application example demonstrates the development of NN-based simulation metamodels to represent an ED DES model. It clearly shows that NN is a good candidate for developing simulation metamodels.

The third application example demonstrates the application of the FMI approach with a focus on the underlying MMI approach. The application example starts with a two-unit MUHS and then expands to include two more units. It confirms that the MMI approach and the FMI approach in general can make significant contributions to the successful implementation of the AHSM approach. It also shows that even with the MMI approach, the development of the integrated MUHS simulation model can become a somewhat complex project when the number of units in the MUHS increases. This indirectly proves the value that the system decomposition technique can bring to the AHSM approach.

9.2 Major Conclusions

Based on the research summary presented in the previous section, the major conclusions of this research can be summarized as follows:

1) The ND-CASs model introduced in the research provides a simple representation of MUHSs for understanding and managing the complexity and dynamics in the operations of the MUHSs. Using the ND-CASs model, the dynamics and complexity

in the operations of MUHSs can be explained by the independent operations of each unit and their interactions, and the study of MUHSs can become more manageable by dividing the overall study of the MUHSs into isolated studies of each unit and subsystem, combined with aggregated studies on the inter-unit/subsystem interactions. The ND-CASs model can also help stakeholders involved in the decision-making process develop common understanding of the operations of the MUHSs and make the solutions of decision problems more relevant to the overall operations of the MUHSs as well as to the individual units and subsystems. This will smooth and speed up the decision-making process. In addition, the ND-CASs model can also be used to assist in designing the architecture of the MUHS simulation models and provides guidelines for the development of unit, subsystem and overall MUHS simulation models.

2) The AHSM approach proposed in this research aims to improve the rapidity and flexibility for developing MUHS simulation models to effectively support decision-making in MUHSs. The complexity of decision problems and the dynamics of the decision-making process in MUHSs, together with the limitations of current SM practices in healthcare, are the three driving forces for the introduction of the AHSM approach. AHSM is a high-level SM approach intended to support the full lifecycle of decision-making in MUHSs by facilitating communications, coordination and interactions between healthcare modelers, managers and other stakeholders in the decision-making process of MUHSs.
3) The implementation of AHSM needs supporting techniques and approaches. Three major ones are proposed in this research. As a logical and low-complexity tool, the system decomposition technique can be used to decompose a MUHS with a large

number of units into layers of subsystems. As shown in Figure 9-1, the system decomposition technique can be viewed as an operational version of the ND-CASs model. The identified hierarchical structure of a MUHS can be used in the architectural design of the MUHS simulation model and serves as a guiding map for the development of unit and subsystem simulation models as well as the overall MUHS simulation model. The simulation metamodeling technique can be used to simplify the unit simulation models while retaining their main features using NNs. It can reduce the experimental run time of the simulation models and eliminate the need for special software package to run a particular simulation model. It can also add rapidity and flexibility to the model integration process when used with the MMI or HMI approach. Finally, a procedural FMI approach is proposed as a framework for selecting the appropriate approach for performing different model integration tasks for the development of the MUHS simulation models.

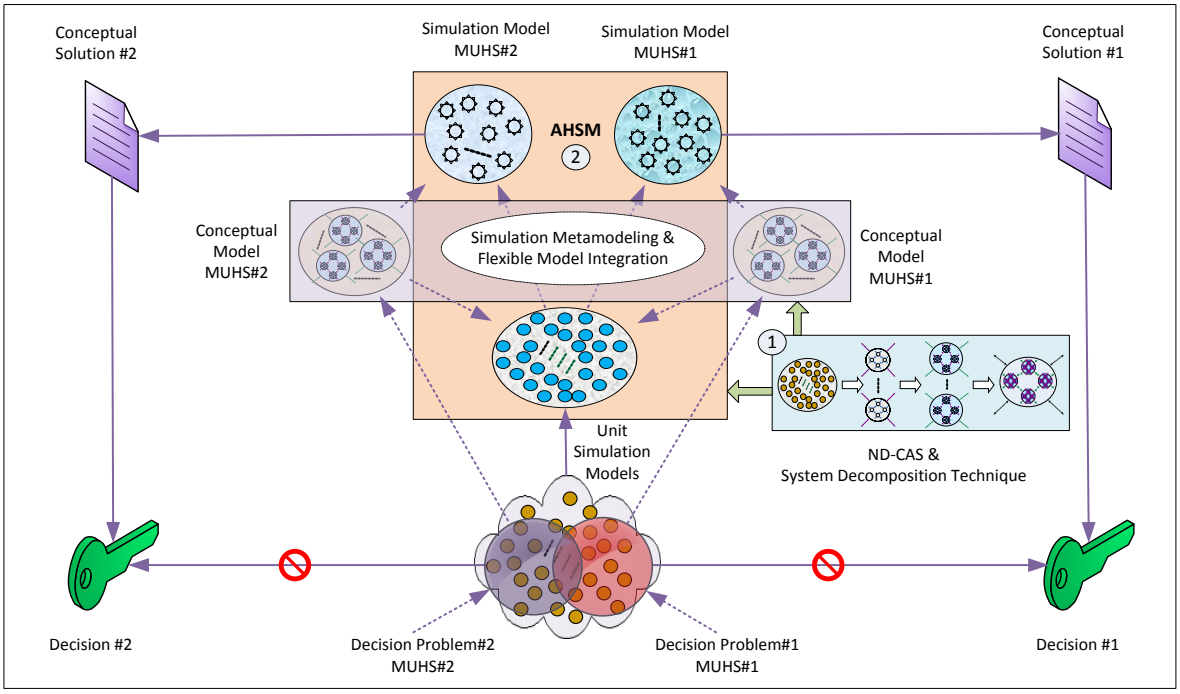


Figure 9-1 The Role of the ND-CASs Model and the System Decomposition Technique in AHSM

9.3 Research Novelty and Contributions

The novelty of this research comes from the introduction of the AHSM approach and the ND-CASs model into the field of healthcare simulation modeling. The novelty of AHSM resides in its intention to support the full lifecycle of decision-making for managing the operation of MUHS by facilitating communications, coordination and interactions between healthcare modelers, managers and other stakeholders in the decision-making process of MUHSs. Inspired by theories of systems, the AHSM approach views the SM application in MUHSs as an ongoing, multi-scale, multi-level and interconnected activity. This represents a different approach compared with those used in the current SM practices in healthcare, which focus on unit-/facility-specific and one-time only SM applications.

The implementation of the AHSM approach is supported by several well-established techniques/approaches i.e. PCA, NN-based metamodeling and model integration. But what is new for these techniques/approaches in this research is their application in the domain of healthcare simulation modeling and their combination as a toolkit to support the implementation of the AHSM approach.

The novelty of the ND-CASs model comes from the integration of the key concepts in the theories of CASs and ND systems. These two theories have been very successful for being used as standalone theories but it is in this research that they are used together to help in understanding and managing the complexity of MUHSs.

The contributions of this research can be summarized as follows:

The most important contribution of this research is the AHSM approach. AHSM offers the HSM community a solution and a toolkit for developing simulation models of MUHSs with a large number of units in a rapid and flexible manner. AHSM also provides healthcare organizations a framework for planning SM activities with a broad scope and from a long-term perspective, thus maximizing the return from the investment in SM activities.

Yet, most of the claimed contributions of AHSM approach still exist in theory and they have not been fully demonstrated in the research. This is partially because AHSM is a process-oriented approach and it requires buy-ins from the healthcare organizations first for its implementation. In addition, it also takes a long time to realize and visualize its real effects. But at the technical level, this research does prove that the AHSM approach has

the capability to improve the SM applications in MUHSs by reusing existing unit simulation models. It can push ahead the current SM practices in healthcare without dramatic changes. As mentioned in early chapters of the research, the unit-/facility-specific SM applications are not problems by themselves. But if healthcare organizations only focus on these SM applications, it will be a problem. The AHSM can definitely help senior managers in the healthcare organizations to steer away from that direction.

Another important contribution of this research is the ND-CASs model, which is introduced to represent and describe the operations of MUHSs. The ND-CASs model is developed from a theoretical perspective and integrates the key concepts in the theories of CASs and ND systems to provide a more holistic representation and description of certain types of complex systems such as MUHSs. On the practical side, the HSM community can use the ND-CASs model as the theoretical foundation to better model MUHSs. For healthcare organizations, the ND-CASs model can serve as a conceptual platform for different stakeholders involved in the decision-making process of MUHSs to build a common understanding of decision problems and better relate them to the operations of the MUHSs as well as to the operations of the individual units to smooth and speed up the decision-making process. Part of the value the ND-CASs model has been demonstrated in the decomposition example of the CHR healthcare system. But its full value needs yet to be demonstrated in the real-world applications.

One more contribution of this research is that the AHSM supporting techniques and approaches can be used separately as general purpose tools in HSM studies. For example, the system decomposition technique can be used to assist the modular design of

healthcare simulation models, and simulation metamodels can be implemented into EXCEL or other tools to help healthcare managers in making day-to-day decisions without the need to run the original simulation models.

9.4 Research Limitations

This research covers a variety of subtopics, for many of which there are few prior research studies to be used as reference. Given the limited space in this dissertation, it is impossible to address all the subtopics in detail and a trade-off between breadth and depth has been made for some of them. For example, while the AHSM approach is the main focus of this research, it has to be used in conjunction with many existing SM methods and techniques, e.g. those for data collection, model verification and validation. In this research, it is assumed that these SM methods and techniques can naturally work together with the AHSM approach. However, the research does not spend extra time investigating the possible challenges and opportunities that AHSM may have created for these SM methods and techniques and exploring the possibilities for the better use of them to support the AHSM approach. Another example is the SM methods discussed in the research. The review of literature on HSM applications focuses on five commonly used SM methods, but the DES method is primarily used in the application examples. The AHSM approach is definitely not designed only to support the DES method or the five SM methods covered in the literature review. Instead, AHSM is supposed to be neutral to underlying SM methods. However, this research does not explore the possibility of using other SM methods. There is one more example, which is the use of NN as the primary technique for developing simulation metamodels, though there are a number of other

alternative techniques, e.g. RSM and multiple regression techniques. Compared with other techniques, NN has many unique characteristics but its use as a simulation metamodeling technique is a recent development. The choice of using NN in this research is not to recommend NN as the only technique for developing simulation metamodels. Instead, the purpose is just to promote NN as another good candidate technique of simulation metamodeling. However, detailed comparisons between NN and other simulation metamodeling techniques are not explored in this research.

9.5 Future Research Directions

This research can be extended in at least the following two directions. First, further research can be conducted to improve the AHSM approach by addressing the limitations identified in the previous section. Possible research questions might include: *1) how can the existing SM methods and techniques be effectively integrated into the AHSM approach in the study of MUHSs?; 2) how to select SM methods in the study of MUHSs using the AHSM approach?; and 3) how to choose the appropriate simulation metamodeling techniques in the application of the AHSM approach in the study of MUHSs?*

Second, AHSM represents a new concept for applying SM in healthcare. To make it successful, future research is needed on a possible redefined decision-making process for managing the operations of MUHSs with AHSM playing a central role in that process. In the meantime, more AHSM-based and application-focused researches are required to prove the value that AHSM can deliver to support the making of operational decisions in MUHSs.

References

Abdul Majid, M. (2011). Human behaviour modelling: an investigation using traditional discrete event and combined discrete event and agent-based simulation (Doctoral dissertation, University of Nottingham).

Addanki, S., Cremonini, R. and Penberthy, J. S. (1990). Reasoning About Assumptions in Graphs of Models, in Readings in Qualitative Reasoning About Physical Systems (Weld, D.S. and J. de Kleer eds.), Morgan Kaufman, San Mateo, CA, 546-552.

Alam, F. M., McNaught, K. R. and Ringrose,T. J. (2004). A comparison of experimental designs in the development of a neural network simulation metamodel. Simulation Modelling: Practice and Theory 12(7-8) 559-578.

Argent, R. M. (2004). An overview of model integration for environmental applications—components, frameworks and semantics. Environmental Modelling & Software, 19(3), 219-234.

Badiru, A.B. and Seiger, D.B. (1993). Neural network as a simulation metamodel in economic analysis of risky projects, Technical Report (Department of Industrial Engineering, University of Oklahoma).

Badiru, A. and Siege, D.B. (1998). Neural network as a simulation metamodel in economic analysis of risky projects. European Journal of Operational Research 105, 130-142.

Banks, J., Carson II, J.S., Nelson, B.L. and Nicol, D.M. (2005). Discrete-event system simulation (4th ed.). Upper Saddle River, NJ: Prentice-Hall.

Barton P, Bryan S. and Robinson S. (2004). Modelling in the economic evaluation of health care: selecting the appropriate approach. Journal of Health Services Research Policy; 9(2): 110-118.

Bauer, K. W. Jr., Kochar, B. and Talavage, J. J. (1985). Simulation Model Decomposition by Factor Analysis. In: Proceedings of the 1985 Winter Simulation Conference (E D. Gantz, E Cl. Blais and E S. Solomon, eds.), (Dec. ll-13), 185-188.

Bauer, K.W. Jr., Kochar, B. and Talavage, J. J. (1991). Discrete Event Simulation Model Decomposition by Principal Components Analysis, ORSA Journal on Computing, 3(1), pp. 23-32.

Bayer, S., Bolt, T., Kapsali, M. and Brailsford, S.C. (2010). The social role of simulation models, International System Dynamics Conferences, Seoul, Korea.

Begun, J.W. and White, K.R. (1999). The Profession of Nursing as a Complex Adaptive System: Strategies for Change. pp. 189-203 in J.J. Kronenfeld (ed.), Research in the Sociology of Health Care, vol. 16 (Greenwich, CN: JAI Press).

Begun, J.W., Zimmerman B and Dooley K. (2002). Health care organizations as complex adaptive systems. In: Mick SS, Wyltenbach ME, eds. Advances in Health Care Organization Theory. San Francisco, Calif: Jossey-Bass: 253-288.

Beynon, M., Rasmequan, S. and Russ, S. (2002). A new paradigm for computer-based decision support, Decision Support Systems 33 (2) 127-142.

Bhargava, H. K., Kimbrough, S. and Krishnan R. (1991). Unique Names Violations: A Problem for Model Integration, ORSA Journal of Computing, Vol. 3, No. 2, 107-120.

Bhrammanee, T. and Wuwongse, V. (2008). ODDM: a framework for model bases, Decision Support Systems 44 (3) 689-709.

Bishop, C.M. (1995). Neural Networks for Pattern Recognition. Clarendon Press, Oxford.

Blanning, W. R. (1975). The construction and implementation of metamodels. Simulation, 24-25 (6), 177-184.

Borshchev, A. and Filippov, A. (2004). From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools. In: Proceedings of the 22nd International Conference of the System Dynamics Society, July 25-29, 2004, p. 45.

Box, G.E.P. and Draper, N.R. (1987). Empirical Model-building and Response Surfaces. Wiley, New York.

Bradley, G. and Clemence R. (1987). A Type Calculus for Executable Modeling Languages, IMA Journal of Mathematics in Management, Vol. 1, No. 4, 277-291.

Brailsford, S.C. and Hilton, N.A. (2000). A comparison of discrete event simulation and system dynamics for modelling healthcare systems. In: Riley J (ed.) Proceedings from ORAHS 2000, Glasgow, Scotland, pp 18-39.

Brailsford, S.C., Lattimer, V.A., Tarnaras, P. and Turnbull, J.C. (2004). Emergency and on-demand health care: modelling a large complex system. J Opn Res Soc 55(1): 34-42.

Brailsford, S.C., Katsaliaki, K., Mustafee, N. and Taylor, S. J. E. (2006). Modelling Very Large Complex Systems using Distributed Simulation: A Pilot Study in a Healthcare Setting. In Proceedings of the 2006 Operational Research Society Simulation Workshop (SW06), Leamington Spa, UK. pp. 257-262.

Brailsford, S.C., Desai, S.M. and Viana J. (2010). Towards the Holy Grail, combining system dynamics and discrete-event simulation in healthcare. In Proceedings of the 2010 Winter Simulation Conference (JOHANSSON BB, JAIN S, MONTOYA-TORRES J, HUGAN J and YUCESAN E, eds.), IEEE, Baltimore, Maryland.

Brandeau, M.L., Sainfort, F. and Pierskalla, W.P. (2004). Health care delivery: Current problems and future challenges, in Operations Research and Health Care: A Handbook of Methods and Applications (eds. Brandeau, Sainfort and Pierskalla). Kluwer Academic Publishers. Boston, MA, chap. 1.

Briggs, J. and Peat, F. D. (1989). Turbulent Mirror: An Illustrated Guide to Chaos Theory and the Science of Wholeness. Harper & Row, New York.

Brownlee, J. (2007). Complex Adaptive Systems, Complex Intelligent Systems Laboratory, Centre for Information Technology Research, Faculty of Information Communication Technology, Swinburne University of Technology: Melbourne, Australia.

Carlile, P.R. (2002). A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development, Organization Science, 13(4), pp. 442-455.

Chahal, K. and Eldabi, T. (2008a). System dynamics and discrete event simulation: A meta-comparison. In Operational Research Society Simulation Workshop (SW08). 189-197.

Chahal, K. and Eldabi, T. (2008b). Applicability of Hybrid Simulation to Different Models of Governance in UK Healthcare, Proceedings of the 2008 Winter Simulation Conference S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

Cochran, J. and Bharti, A. (2006). A multi-stage stochastic methodology for whole hospital bed planning under peak loading. International Journal of Industrial and Systems Engineering, 1(1):8-36.

Cooke, D., Rohleder, T., Rogers, P. and Xu, S. (2008). System Dynamics Modeling for Health Care Capacity Planning, 34th Annual Conference on Operational Research Applied to Health Services, Toronto, ON, Canada.

Cooper, K., Brailsford, S.C., Davies, R. and Raftery, J. (2006). A review of health care models for coronary heart disease interventions. Health Care Management Science; 9: 311-324.

Dattée, B. and Barlow, J. (2010). Complexity and whole-system change programmes. Journal of Health Services Research & Policy, 15, 19-25.

Davies, R. and Davies, H. (1994). Modelling patient flows and resource provision in health systems. Omega 22:123-131.

Davies, H.T.O. and Davies, R. (1995). Simulating health systems: Modelling problems and software solutions, European Journal of Operational Research 87: 35-44.

De Wolf, T. and Holvoet, T., (2005). Emergence Versus Self-Organisation: Different Concepts but Promising When Combined, Engineering Self Organising Systems: Methodologies and Applications, Lecture Notes in Computer Science, Volume 3464, May 2005, pp. 1-15.

Dillon, W. and Goldstein, M. (1984). Multivariate Analysis: Methods and Applications, Wiley, New York.

Dolk, D. and Kottemann, J. (1993). Model Integration and a Theory of Models, Decision Support Systems, 9:1, pp. 51-63.

Donabedian, A. (1966). Evaluating the quality of medical care. Milbank Memorial Fund Quarterly; 44:166-206.

Dooley, K. and Plsek, P. (2001). A Complex Systems Perspective on Medication Errors. Working Paper, Arizona State University.

Draman, M., Altinel, İK., Bajgoric, N. and Tamer Unal, A. and Birgoren, B. (2002). A clone-based graphical modeler and mathematical model generator for optimal production planning in process industries, European Journal of Operational Research 137 (3) 483-496.

Dreyfus, H.L. and Dreyfus, S.E. (1986). Mind Over Machine: The Power of Human Intuition and Expertise in the Era of the Computer, Free Press, New York.

Egidi M. & Marengo L. (2002). Cognition, institutions, near decomposability: rethinking Herbert Simon's contribution, CEEL, Working Paper, 6-02.

Elam, J. J. and Lee, R. M. (1986). Guest Editors' Introduction, Decision Support Systems, special issue, Vol.2, No.1, 1-2.

Eldabi, T., Irani, Z. and Paul, R.J. (2002). A proposed approach for modelling health-care systems for understanding. J Mng Med 16(4): 170-187.

Eldabi, T., Paul, R. J. and Young T. (2007). Simulation modeling in healthcare: reviewing legacies and investigating futures, Journal of the Operations Research Society, Vol. 58, pp. 262-270.

England, W. and Roberts, S. (1978). Applications of computer simulation in health care. In: Highland HJ, Hull LG and Neilsen NR (eds). Proceedings of the 1978 Winter Simulation Conference. Institute of Electrical and Electronics Engineers, Miami Beach, Florida, USA, 4-6 December, pp 665-676.

Eoyang, G. and Berkas, T. (1998). Evaluation in Complex Adaptive System, in M. Lissack and H. Gunz (eds) Managing Complexity in Systems. Westport, CT: Quorum Books.

Ewenstein, B. and Whyte, J. (2009). Knowledge practices in design: The role of visual representations as 'Epistemic objects', Organization Studies, 30 (1), pp. 7-30.

Fetter, R.B. and Thompson, J.D. (1965). The simulation of hospital systems. Opns Res 13: 689-711.

Fierbinteanu, C. (1999). A decision support systems generator for transportation demand forecasting implemented by constraint logic programming, Decision Support Systems 26 (3) 179-194.

Fone, D., Hollinghurst, S., Temple, M., Round, A., Lester, N., Weightman, A., et al. (2003). Systematic review of the use and value of computer simulation modelling in population health and health care delivery. J Publ Health Med; 25: 325-335.

Forrester, J. W. (1961). Industrial Dynamics. Cambridge, Mass: MIT Press.

Fraser, S.W. and Greenhalgh, T. (2001). Coping with complexity: educating for capability. BMJ;323:799-803.

Friedman, L. W. and Pressman, I. (1988). The metamodel in simulation analysis: can it be trusted? Journal of the Operational Research Society, 39 (10), 939-948.

Fujimoto, R.M. (1999). Parallel and Distributed Simulation Systems. John Wiley & Sons Inc., New York.

Fujimoto, R.M. (2003). Distributed simulation systems. In Proceedings of the 2003 Winter Simulation Conference, 124-134.

Geoffrion, A.M., (1989a). Reusing Structured Models via Model Integration, Proceedings of Twenty-Second Annual Hawaii International Conference on the System Sciences, IEEE Computer Society, pp. 601-611.

Geoffrion, A.M. (1989b). Computer-based modeling environments, European Journal of Operational Research 41 (1) 33-45.

Geoffrion, A.M. (1992a). The SML Language for Structured Modeling: levels 1 and 2, Operations Research, Vol. 40: 1, pp. 38-57.

Geoffrion, A.M. (1992b). The SML Language for Structured Modeling: levels 3 and 4, Operations Research, Vol. 40:1, pp. 58-75.

Geoffrion, A. M. (1987). An Introduction to Structured Modeling, Management Science, Vol.33, No.5, 547-588.

Golub, A.L. (1997). Decision Analysis: An Integrated Approach, John Wiley & Sons, New York.

Gunal, M. and Pidd, M. (2009). Discrete event simulation for performance modelling in healthcare: A review of the literature. Working paper. Department of Management Science, Lancaster University.

Gunal, M. (2012). A guide for building hospital simulation models, Health Systems 1, 17-25.

Hair, J.F., Black, W.C., Babin, B.J., Anderson, R.E. and Tatham, R.L. (2006). Multivariate Data Analysis, 6th Ed., Prentice-Hall, Upper Saddle River, NJ.

Hancock, W. and Walter, P. (1984). The use of admissions simulation to stabilize ancillary workloads. Simulation 43: 88-94.

Harrison, E.F. (1999). The Managerial Decision-Making Process. Houghton Mifflin College Div.

Haykin, S. (1999). Neural Networks: A comprehensive foundation. Upper Saddle River, N. J., Prentice Hall.

Heylighen, F. (1989). Self-Organization, Emergence and the Architecture of Complexity, in: Proc. 1st Eur. Conf. on System Science, (AFCET, Paris), p. 23-32.

Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer Feedforward Networks are Universal Approximators. Neural Networks, 2, pp. 359-366.

Hurrion, R.D. (1992). Using a neural network to enhance the decision making quality of a visual interactive simulation model, Journal of the Operational Research Society 43 333-341.

Hurrion, R.D. (1998). Visual interactive meta-simulation using neural networks, International Transactions in Operational Research 5 261-271.

IEEE 1516. (2000). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA). Institute of Electrical and Electronics Engineers.

Iyer, B., Shankaranarayanan, G. and Lenard, M.L. (2005). Model management decision environment: a web service prototype for spreadsheet models, Decision Support Systems 40 (2) 283-304.

Jacobson, S. H., Hall, S. N. and Swisher, J. R. (2006). Discrete-Event Simulation Of Health Care Systems. In R. W. Hall (Ed.), Patient Flow: Reducing Delay in Healthcare Delivery: Springer, 211-252.

Jiang, J.J. and Klein, G. (2000). Side effects of decision guidance in decision support systems, Interacting with Computers 12 (5) 469-481.

Jun, J.B., Jacobson, S.H. and Swisher, J.R. (1999). Application of discrete-event simulation in health care clinics: a survey. Journal of the Operational Research Society, 50(2), 109-123.

Kaiser, H.F. (1958). The varimax criterion for analytic rotation in factor analysis. Psychometrika, 23(3), pp. 187-200.

Kate, A.S. and Jatinder, N.D.G. (2000). Neural networks in business: Techniques and applications for the operations researcher. Computers & Operations Research 27(11,12): 1023.

Katsaliaki, K. and Mustafee, N. (2010). Improving decision making in healthcare services through the use of existing simulation modelling tools and new technologies, Transforming Government: People, Process and Policy, Vol. 4 Iss: 2, pp.158 - 171.

Kelly, K. (1994). Out of Control. Addison-Wesley, Reading, MA.

Kelton, W.D. and Law., A.M. (2000). Simulation modeling and analysis. Boston, MA: McGraw Hill.

Khuri, A. and Cornc H.J.A. (1987). Response Surfaces, Designs and Analyses. New York: M. Dekker.

Kilmer, R.A. and Smith, A.E. (1993). Using artificial neural networks to approximate a discrete event stochastic simulation model, in: C.H. Dagli, L.I. Burke, B.R. Fernandez, J. Ghosh (eds.), Intelligent Engineering Systems Through Artificial Neural Networks, vol. 3, ASME Press, New York, pp. 631-636.

Kilmer, R.A., Smith, A.E. and Shuman, L.J. (1994). Neural networks as a metamodelling technique for discrete event stochastic simulation, in: C.H. Dagli, B.R. Fernandez, J. Gosh, R.T. Kumara (Eds.), Intelligent Engineering Systems Through Artificial Neural Networks, vol. 4, ASME Press, New York, pp. 1141-1146.

Kleijnen, J.P.C. (1975). A comment on Blanning's ''metamodel for sensitivity analysis: The regression metamodel in simulation". Interfaces 5 (3), 21-23.

Kleijnen, J.P.C. (1979). Regression metamodels for generalising simulation results. IEEE Transactions on Systems, Man and Cybernetics. SMC-9 (2), 93-96.

Kohn, L.T., Corrigan, J. and Donaldson, M.S. (eds.). (2000). To Err is Human: Building a Safer Health System (Washington, DC: National Academy Press).

Kontopoulos, K.M. (1993). The Logics of Social Structure. Cambridge: Cambridge University Press.

Krishnan, R. and Chari, K. (1993). Model management: survey, future research directions and a bibliography, ORSA CSTS Newsletter, 14 (1), pp. 1-22.

Krishnan, R., Piela, P. and Westerberg, A. (1993). Reusing Mathematical Models in ASCEND, in Recent Developments in Decision Support Systems (Holsapple, C and A. Whinston eds.), NATO ASI Series, Springer-Verlag, 275-294.

Kuo, C. and Reitsch, A. (1995). Neural networks vs. conventional methods of forecasting. Journal of Business Forecasting Methods and Systems, 14(4), 17-22.

Lane, D.C., Monefeldt, C. and Rosenhead, J.V. (2000). Looking in the wrong place for healthcare improvements: a system dynamics study of an accident and emergency department. Journal of the Operational Research Society, 51, 518-531.

Lane D.C. (2000). You just don't understand me: modes of failure and success in the discourse between system dynamics and discrete event simulation, Working paper no. LSEOR 00.34, London School of Economics.

Langabeer, J.R. (2008). Health care operations management: a quantitative approach. Boston: Jones and Bartlett Publishers.

Lehaney, B. and Hlupic, V. (1995). Simulation modelling for resource allocation and planning in the health sector. J Roy Soc of Health 115: 382-385.

Leonard, C. G., Gerald, A. B., Stephen, J. D. and Diane, M. G. (2006). Developing a policy simulator at the acute-aged care interface. Australian Health Review, 30(4), 450.

Liew, A. and Sundaram, D. (2009). Flexible modelling and support of interrelated decisions. Decision Support Systems, 46(4), 786-802.

Looney, C.G. (1996). Advance in feed-forward neural networks: Demystifying knowledge acquiring black boxes. IEEE Transaction on Knowledge and Data Engineering, 8, 211-226.

Lorenz, T. and Jost, A. (2006). Towards an orientation framework in multiparadigm modeling. Proceedings of the System Dynamics Society Conference. Nijmegen, The Netherlands."

Lowery, J.C. (1992). Simulation of a hospital's surgical suite and critical care area. In: Swain JJ, Goldsman D, Crain RC and Wilson JR (eds.). Proceedings of the 1992 Winter Simulation Conference. Institute of Electrical and Electronics Engineers, Arlington, Virginia, USA, 13-16 December, pp1071-1078.

Macal, C. and North, M. (2010). Tutorial on agent-based modelling and simulation. Journal of Simulation, 4(3):151-162.

Marion, R. and Bacon, J. (2000). Organizational Extinction and Complex Systems. Emergence 1(4):71-96.

Mathur, K. and Solow, D. (1994). Management Science: The Art of Decision Making, Prentice Hall, Englewood Cliffs, New Jersey.

Matthes, S. (1988). Discrete Event Simulation Model Decomposition, Master's thesis, AFIT/GOR/ENS/88M Air Force Institute of Technology, Wright-Patterson AFB OH, Mar 1988.

McDaniel, Jr., R.R. and Driebe, D.J. (2001). Complexity science and health care management. Adv Strat Manage; 2:11-36.

McDaniel, Jr., R.R. (1997). Strategic Leadership: A View from Quantum and Chaos Theories. Health Care Management Review 22(1):21-37.

Meckesheimer, M. (2001). A Framework for Metamodel-Based Design: Subsystem Metamodel Assessment and Implementation Issues, Ph.D. Dissertation, Industrial Engineering Dept., Pennsylvania State University, University Park, PA.

Mielczarek, B. and Uzialko-Mydlikowska, J. (2012). Application of computer simulation modeling in the health care sector: a survey, SIMULATION vol. 88 no. 2 197-216.

Morecroft, J.D.W. and Robinson, S. (2008). Explaining puzzling dynamics: comparing the use of system dynamics and discrete-event simulation, SD plus meeting, LSBU.

Morgan, G. (1997). Images of Organization, second ed. (Thousand Oaks, CA: Sage).

Mustafee, N., Katsaliaki, K. and Taylor., S. (2010). Profiling Literature in Healthcare Simulation. Simulation 86(8/9):543-558.

Myers, R.H. (1976). Response Surface Methodology. Boston: Allyn and Bacon.

Owen, C., Love, D. and Albores, P. (2008). Selection of Simulation Tools for Improving Supply Chain Performance. Proceedings of the 2008 OR Society Simulation Workshop 199-207.

Paley, J. (2010). The appropriation of complexity in health care. Journal of Health Services Research & Policy, 15(1): 59-61.

Parunak, H.V.D., Savit, R. and Riolo, R.L. (1998). Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. In Sichman, J.S., Conte, R. and Gilbert, N. (Eds.), Multi-Agent Systems and Agent-Based Simulation, Springer Verlag.

Patel, M. and Nagl, S. (2011). The Role of Model Integration in Complex Systems Research. SCS M&S Magazine, n1 (January).

Piela, P., McKelvey, R. and Westerberg, A. (1992). An Introduction to ASCEND: Its Language and Interactive Environment, in the proceedings of the 25th Annual Hawaii International Conference on System Sciences, Vol. III, IEEE Computer Society Press, Los Alamitos, CA, 449-461.

Pierreval, H. and Huntsinger, R.C. (1992). An investigation on neural network capabilities as simulation metamodels, in: Proceedings of the 1992 Summer Computer Simulation Conference, Society for Computer Simulation, San Diego, CA, pp. 413-417.

Plsek, P. and Greenhalgh T.(2001). The challenge of complexity in health care. Br Med; 323: 625-628.

Power, D.J. and Sharda, R. (2007). Model-driven decision support systems: concepts and research directions, Decision Support Systems 43 (3) 1044-1061.

Priesmeyer, H.R. and Sharp, L.F. (1995). Phase Plane Analysis: Applying Chaos Theory in Health Care. Quality Management in Health Care 4(1):62-70.

Priesmeyer, H.R., Sharp, L.F., Wammack, L. and Mabrey, J.D. (1996). Chaos Theory and Clinical Pathways: A Practical Application. Quality Management in Health Care 4(4): 63-72.

Pugh, G. A. (2006). Agent-Based Simulation of Discrete-Event Systems , In Proceedings of the 2006 Illinois-Indiana and North Central Joint Section Conferences.

Rizzoli, A. E., Davis, J. R. and Abel, D. J. (1998). Model and data integration and re-use in environmental decision support systems. Decision support systems, 24(2), 127-144.

Rodriguez, J.F.D. (2008). Metamodeling Techniques to Aid in The Aggregation Process of Large Hierarchical Simulation Models, PhD Dissertation, Department of The Air Force, Air University, Wright-Patterson Air Force Base, Ohio.

Rohleder R., Rogers P., Cooke D. and Xu, S. (2009). Emergency Department Simulation: A Report for The Calgary Health Region, Calgary, AB, Canada.

Rowlands, G., Sims, J. and Kerry, S. (2005). A lesson learnt: The importance of modelling in randomized controlled trials for complex interventions in primary care. Family Practice 22: 132-9.

Royston, G., Dost, A., Townshend, J. and Turner, H. (1999). Using system dynamics to help develop and implement policies and programmes in health care in England. System Dynamics Review, 15(3), 293.

Rumelhart, D.E., Hinton, G.E. and William, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.), Parallel distributed processing: Explorations in the microstructure of cognition (pp. 318-362). Cambridge, MA: MIT Press.

Santos, M.I. and Santos, P.M. (2007). Simulation metamodels for modeling output distribution parameters. In: Henderson, S.G., Biller, B., Hsieh, M.-H., Shortle, J., Tew, J.D., Barton, R.R. (Eds.), Proceedings of the Winter Simulation Conference. IEEE, pp. 910-918.

Sapsed, J. and Salter, A. (2004). Postcards from the edge: local communities, global programs and boundary objects. Organization Studies. 25: 1515-1534.

Sarle, W.S. (1994). Neural networks and statistical models. Proceedings of the SAS Users Group International Conference, pp. 1528-1550.

Schieritz, N. and Milling, P.M. (2003). Modeling the Forest or Modeling the Trees - A Comparison of System Dynamics and Agent-Based Simulation, Eberlein, R.L. et al. (eds.), Proceedings of the 21st International Conference of the System Dynamics Society, New York City.

Scholl, H.J. (2001). Looking across the fence: Comparing findings from SD modeling efforts with those of other modeling techniques, 19th Int. Conf. Syst. Dyn. Soc.

Schön, D. A. (1983). The Reflective Practitioner, New York: Basic Books.

Seila, A.F. and Brailsford, S. (2009). Opportunities and Challenges in Health Care Simulation - Advancing the Frontiers of Simulation - Springer.

Sharp, L.F. and Priesmeyer, H.R. (1995). Tutorial: Chaos Theory - A Primer for Health Care. Quality Management in Health Care 3(4): 71-86.

Sibthorpe, B., Glasgow, N. and Longstaff, D. (2004). Complex Adaptive Systems: A Different Way of Thinking about Health Care Systems. Canberra: Australian Primary Care Research Institute, Australian National University (ANU). (http://www.anu.edu.au/aphcri/Publications/Background_paper_stream1.pdf).

Silver, M.S. (1991). Decisional guidance for computer-based decision support, MIS Quarterly 15 (1) 105-122.

Simon H.A. (1962). The Architecture of Complexity, Proceedings of the American Philosophical Society 106 p. 467-482.

Simon, H. (1977). The New Science of Management Decision, Prentice-Hall, Englewood Cliffs, NJ.

Smith, P. (1995). Large scale models and large scale thinking: The case of the health services. Omega 23(2): 145-157.

Smith-Daniels, V.L., Schweikhart, S.B. and Smith-Daniels, D.E. (1988). Capacity management in health care services: Review and future research directions. Decis Sci 19: 889-918.

Sprague, R.J. (1980). A framework for the development of decision support systems, MIS Quarterly 4 (4) 1-26.

Starr, P.J. (1991). Integration of simulation and analytical submodels for supporting manufacturing decisions. International Journal of Production Research, 29 (9), 1733-1746.

Sterman, J. (2000). Business Dynamics: Systems Thinking and Modeling for a Complex World. Irwin/McGraw-Hill: New York.

Stern, H.S. (1996). Neural networks in applied statistics, Technometrics, 38, 205-214.

Sun, Y. and Cheng, L. (2002). A Survey on Agent-Based modeling and Equation-based modeling, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.9669&rep=rep1&type=pdf.

Sweetser, A. (1999). A comparison of system dynamics and discrete event simulation, 17th Int. Conf. of the System Dynamics Society, Wellington, New Zealand, pp.8.

Tang, X. (2002). Model integration, J. of Systems Engineering, 16(5) 322-329.

Tzafestas, S.G., Dalianis, P.J. and Anthopoulos, G. (1996). On the overtraining phenomenon of backpropagation neural networks. Mathematics and Computers in Simulation, 40, 507-521.

Valinsky, D. (1975). Simulation. In L. J. Shuman, R. D. Speas Jr., & J. P. Young (Eds.), Operations Research in Health Care: A Critical Analysis. Baltimore: Johns Hopkins University Press, 114-176.

Vanberkel, P.T., Boucherie, R.J., Hans, E.W., Hurink, J.L. and Litvak, N. (2009). A survey of health care models that encompass multiple departments. Working paper University of Twente.

Vieira, I.T., Cheng, R.C.H., Harper, P.R. and De Senna, V. (2010). Small world network models of the dynamics of HIV infection. Annals of Operations Research 178(1), 173-200.

Vissers, J. and Beech R. (2005). Health operations management: patient flow logistics in health care. London: Routledge.

Wakeland, W.W., Gallaher, E.J., Macovsky, L.M. and Aktipis, C.A. (2004). A Comparison of System Dynamics and Agent-Based Simulation Applied to the Study of Cellular Receptor Dynamics, hicss, vol. 3, pp.30086b, Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 3.

West, B. and Deering, B. (1995). The Lure of Modern Science: Fractal Thinking, World Scientific, River's Edge, NJ.

Wilson, T. and Holt, T. (2001). Complexity and clinical care. BMJ; 323:685-8.

Wilson, J.C.T. (1981). Implementation of computer simulation projects in health care. J Opl Res Soc 32: 825 -832.

Xu, S., Rogers, P., Rohleder, T. and Cooke, D., (2008). Improving Emergency Department Physician Management via Computer Simulation, Proceedings of the 2008 Industrial Engineering Research Conference, J. Fowler and S. Mason, eds, Vancouver, BC, Canada.

Xu, S. (2010). Meta-modeling of An Emergency Department DES Model: A Neural Networks Approach, Proceedings of 2010 POMS Annual Conference.

Yu, X.H. and Chen, G.A. (1997). Efficient backpropagation learning using optimal learning rate and momentum. Neural Networks, 10, 517-527.

Yu, B. and Popplewell, K. (1994). Metamodels in manufacturing: a review. International Journal of Production Research 32(4): 787 - 796.

Yu, T.T., Scanlan, J.P. and Wills, G.B. (2007). Agent-Based and Discrete-Event Modelling: A quantitative approach,. Proceedings of the 7th AIAA Aviation Technology, Integration and Operation Conference, ATIO. Belfast, Northern Ireland.

Zeigler, B. P., Kim, T. G. and Praehofer, H. (2000). Theory of Modeling and Simulation, 2nd Ed., New York, NY, Academic Press.

Zimmerman, B., Lindberg, C. and Plsek, P. (1998). Edgeware: Insights from Complexity Science for Health Care Leaders (Irving, TX: VHA).

Appendix A List of Software Packages

ARENA® 12.0

http://www.arenasimulation.com/Products_Products.aspx

IBM® SPSS® Neural Networks

http://www-01.ibm.com/software/analytics/spss/products/statistics/neural-networks/

MATLAB® 2006

http://www.mathworks.com/tagteam/70533_91199v01_MATLABDataSheet_v9.pdf?s_cid=ML2012_bb_datasheet

Vensim® 6.5

http://www.vensim.com/brochure.html

Appendix B FMC ED DES Model Data & Results

B.1 FMC ED Resources

In the FMC ED DES model, the following two key ED resources, ED beds and EPs, are modeled explicitly. The other resources are modeled implicitly (or in an approximate manner). Additional details of the FMC ED DES model are described in the following paragraphs.

ED Beds

The FMC ED includes a main ED area and a Minor Emergency Treatment (MET) area to provide treatment to patients with minor injuries. The main ED is open 24-hour-a-day while the MET area is open 16 hours every day (12:00AM to 2:00AM & 10:00AM to 12:00AM). The main ED has a fixed total of 45 treatment beds, which are allocated to the following three areas: (i) the trauma area with 3 beds; (ii) the monitored bed area with 16 beds; and (iii) the non-monitored bed area with 26 beds. The MET area has 8 beds. The model assumes that these beds are permanently available (i.e. it does not consider any temporary bed closures due to nurse staffing shortfalls).

In the model, when patients have completed triage they will be directed to either a main ED bed or a MET bed (provided that MET is open). Depending on the bed type to which they are directed, patients will wait either in the main ED waiting room or in the MET waiting room. Table B-1 shows the ED bed allocation policy implemented in the model.

Table B-1 ED Bed Allocation Policy

		Allocated ED Beds			
		Trauma	Monitored	Non-monitored	MET
Allocated Bed at Triage	Trauma	*1	2		
	Monitored		1	2	
	Non-Monitored			1	
	MET			2	1
* Priority					

As shown in the above table, in the main ED, a patient directed to a trauma bed at triage can be placed in either a trauma bed or a monitored bed. A patient directed to a monitored bed at triage can be placed in either a monitored bed or a non-monitored bed. A patient directed to non-monitored beds at triage can only be placed in a non-monitored bed. A patient directed to a MET bed at triage can only be placed in a MET bed unless the MET area is closed. In that case, the patient will be moved to the main ED and wait for a non-monitored bed.

In the main ED, after stabilization, a trauma patient will be moved out of the trauma bed and switched to a monitored bed or non-monitored bed for further treatment. After 1st EP assessment, some non-trauma patients in monitored beds will move to a non-monitored bed for further treatment in the main ED. Table B-2 shows the bed reassignment policy implemented in the model. (e.g. after stabilization, 54.4% of trauma patients will be moved to monitored beds, and the other 45.6% will be moved to non-monitored beds).

Table B-2 ED Bed Reassignment Policy

		To		
		Trauma	Monitored	Non-Monitored
From	Trauma		54.4%	45.6%
	Monitored			11.4%

Emergency Physicians (EPs)

The FMC ED is staffed with ten different EP shifts every day. One EP shift covers 7 hours, and each of the other EP shifts covers 8 hours. Four of the nine EP 8-hour shifts are actually split-shifts wherein the EP spends the first 4 hours in the main ED, and the following 4 hours in the MET area.

As shown in the Table B-3, 14 different EP shifts were defined in the model with 10 shifts for the main ED, and 4 shifts for the MET area (the 4 shifts in the MET area and their counterparts in the main ED with the same EP assigned covers 4 hours each).

Table B-3 EP Shift Patterns

Area	Shift No.	Starting Time	Ending Time	Shift Duration	Transfer to	Comments
Main ED	S1	6:00 AM	10:00 AM	4	S11	The first 4 hrs in the main ED, and the next 4 hours in MET
	S2	7:00 AM	3:00 PM	8	S6	
	S3	10:00 AM	2:00 PM	4	S12	The first 4 hrs in the main ED, and the next 4 hours in MET
	S4	11:00 AM	7:00 PM	8	S8	
	S5	2:00 PM	6:00 PM	4	S13	The first 4 hrs in the main ED, and the next 4 hours in MET
	S6	3:00 PM	11:00 PM	8	S9	
	S7	6:00 PM	10:00 PM	4	S14	The first 4 hrs in the main ED, and the next 4 hours in MET
	S8	7:00 PM	3:00 AM	8	S10	
	S9	10:00 PM	6:00 AM	8	S1	
	S10	12:00 AM	7:00 AM	7	S2	
MET	S11	10:00 AM	2:00 PM	4	S12	
	S12	2:00 PM	6:00 PM	4	S13	
	S13	6:00 PM	10:00 PM	4	S14	
	S14	10:00 PM	2:00 AM	4	S10	

After an EP working one of the split-shift moves from the main ED to the MET area, she/he will only pick new patients from the MET waiting room, but will continue to take care of her/his previous patients in the main ED. When an EP ends a shift, she/he will transfer her/his patients to another EP according to a pre-defined scheme. A patient can only be re-assessed by the EP who did the stabilization or 1st assessment for the patient or by the EP to whom the patient has been handed over (when the original EP ends a shift).

The following six types of EP activity were identified during the shadowing of EPs:

1) Stabilization for trauma patients;
2) 1st EP assessment;
3) EP re-assessment after test results are known or after trial of therapy;
4) EP discussion with MD consultants and/or other ED staff on the patient's situation;
5) EP involvement in the trial of therapy;
6) EP dealing with other activities not related to the care of her/his patients, such as discussion with other EP on their patients, responding to pages from other ED staff, etc.

Mean values and distributions of each EP activity were estimated from the EP activity observation and analysis with results shown in the Table B-4 (e.g. the 1st EP assessment time in the main ED can be estimated using a Weibull distribution with a mean value of 15.0 minutes).

Table B-4 Distribution of EP Activity Times

Physician Time Type	Mean (Minute)	Probability Distribution
Stabilization	18.8	Weibull(20.75,1.29735)
1st Assessment in main ED	15.0	Weibull(16.22,2.24842)
1st Assessment in MET	10.3	Weibull(9.01,1.7367)
Re-assessment in main ED	4.9	Lognormal(4.69,4.82)
Re-assessment in MET	2.6	Lognormal(2.62,1.83)
Consultation in main ED	4.7	Lognormal(4.63,5.06)
Consultation in MET	2.3	Lognormal(2.30,1.70)
Trial of Therapy (CTAS1&2)	12.0	Exponential(12)
Trial of Therapy (CTAS3)	6.9	Weibull(7.21,1.41392)
Trial of Therapy (CTAS4&5)	8.3	Gamma(1.87,6.40446)
Other activities	3.5	Triangular(2,3.5,5)

To reduce the complexity of the model, activities 2) and 4) were combined, as were activities 3) and 5), resulting in four fundamental tasks that EPs have to deal with: stabilization; 1st assessment (including discussions with other staff for a proportion of patients); reassessment (including involvement in trial of therapy for a proportion of patients); other tasks not related to care of patients officially under their care. The proportion of patients needing EP involvement in discussions with other staff or in trial of therapy, determined from analysis of the data from the EP shadowing, are show in Table B-5 (e.g. for 59.8% of CTAS 3 patients in the main ED, EPs need to consult with MD consultants or other ED staff on the patient's situation).

Table B-5 Additional EP Task Probabilities (by Treatment Location and CTAS Score)

Treatment Area	Physician Activities	CTAS 1	CTAS 2	CTAS 3	CTAS 4	CTAS 5	Total
Main ED	EP & Other ED Staff Consultation	80.0%	64.6%	59.8%	54.2%	50.0%	61.7%
	Patient Trial of Therapy	0.0%	10.2%	10.4%	4.2%	50.0%	9.9%
MET	EP & Other ED Staff Consultation	80.0%	44.4%	46.5%	42.1%	60.0%	45.3%
	Patient Trial of Therapy	0.0%	22.2%	18.6%	13.2%	20.0%	16.8%

Within the model, EP task durations are randomly generated to match the results from the analysis of data from the EP shadowing, also considering the results in Table B-5. For the direct patient-related tasks (stabilization, 1st assessment, and reassessment), there are three corresponding queues in which patients wait until the EP is available. The other EP tasks are considered to be of two types: 1) high priority (HP) activities; and 2) low priority (LP) activities. For each EP, other tasks are generated according to the pattern described in Table B-6 (e.g. high priority activities arrive according to a Poisson process with an average rate of 2.35 tasks per hour). The durations of these other tasks are sampled from a triangular distribution with a minimum of 2 minutes, a maximum of 5 minutes, and a most likely value of 3.5 minutes. [All numeric values are estimates based on the EP activity observation and analysis].

Table B-6 EP "Other" Task Details

	Arrival Rate (per EP per hour)	Time Consumption (Minute)
Other Activity (high priority)	2.35	Triangular (2,3.5,5)
Other Activity (low priority)	3.3	Triangular (2,3.5,5)

Allowing two classes of other tasks results in there being five different types of task that might be waiting for an EP when a previous task completed. Table B-7 shows the rules implemented in the model whereby an EP chooses which type of task to do next. These rules are our assumptions based on our understanding of the EP behaviour (these can easily be changed if desired).

Table B-7 EP Task Priorities

MD Activity	Priority
Stabilization	1
Other activity (HP)	2
Reassessment	3
1st assessment	4
Other activity (LP)	4

Note: The lower the numeric value, the higher the task priority
HP: High Priority LP: Low Priority

ED Nurses

The following types of nurse are included in the model, though only in an approximate manner due to the lack of data on nurse tasks and task durations:

1) EMS triage nurse: 1 always present;
2) Walk in triage nurse: 1 always present;
3) Trauma bed nurses: 3 always present (each responsible for one trauma bed);
4) Monitored bed nurses: 8 always present (each responsible for two monitored beds);
5) Non-monitored bed nurses: 6 always present (each responsible for 4 or 5 non-monitored beds); and
6) MET bed nurses: 2 always present (each responsible for four MET beds).

All the nurses are assumed to be on duty 24 hours a day, and only those nurse activities directly related to the patient-flow are considered. Specifically, the two triage nurses will only do the triage, but they do back up each other for workload sharing. For the ED bed nurses, only their initial assessment times when a new patient moves to an ED bed and the

finish up time prior to a patient's departure from the ED are considered in the model. Their activity times in the model are estimates using a triangular distribution based on interviews with the FMC ED nurses and other staff (see Table B-8).

Table B-8 Nurse Activity Time Distributions

ED Nruse Time Type	Mean (Minute)	Probability Distribution
Triage	8.0	Triangular(5,7,12)
Admitting	10.7	Triangular(7,10,15)
Initial Assessment	7.3	Triangular(5,7,10)
Finish Up	7.3	Triangular(5,7,10)

B.2 Operational Policies of the FMC ED

This section will describe the general operational policies of the FMC ED that are implemented in the model.

The Main ED and the MET Area

The main ED and the MET area are modeled separately in the DES model with each having its own waiting room, treatment beds, nurses and EPs. When the MET area is closed, any remaining patients in the MET waiting room will be transferred to the main ED waiting room for non-monitored beds. Similarly, when the MET area is closed, any MET patients waiting for a 1st EP assessment will be transferred to the queue waiting for 1st EP assessment in the main ED. However, any patients being treated in the MET area will continue to stay there to finish any remaining treatment prior to departure from the ED. These patients will be taken care of by an EP working in the main ED.

Resource Allocation Priority

Patients waiting for ED beds or for a 1st EP assessment are prioritized based on their CTAS score and the bed type assigned at triage. Initially, a patient's priority is set such that patients with a lower CTAS score have a higher priority, with ties broken based on ideal bed type (trauma, then monitored, then non-monitored) and on time of arrival.

In order to avoid unreasonably long waiting times for patients with higher CTAS scores, a patient's priority is changed if the waiting time becomes excessive. For example, if a patient has been waiting for over one hour in a queue, the patient's priority is changed to match that of a patient whose CTAS score is one less (waiting for the same bed type). Also a maximum waiting time threshold is defined so as to change the priority of any patient that has waited more than some defined threshold (currently this is defined to be 4 hours) to that of a CTAS 1 patient needing a trauma bed.

LWBS Patients

In order to avoid overflowing the FMC ED with unrealistically high queues, a LWBS mechanism is implemented in the model to allow patients to LWBS when the waiting time in the waiting room becomes intolerably long. Upon arrival in the ED, each patient entity is assigned a "waiting time tolerance" such that if the patient is still waiting when waiting time reaches this tolerance, the patient will LWBS. This tolerance is randomly sampled using a NN model that considers both patient CTAS score and the hour of patient arrival (this NN model mimics a waiting time tolerance modeling approach developed by

Dr. Jason Scarlett of Alberta Health Services). Figure B-1 shows some examples of the patient LWBS behaviour from the waiting room.

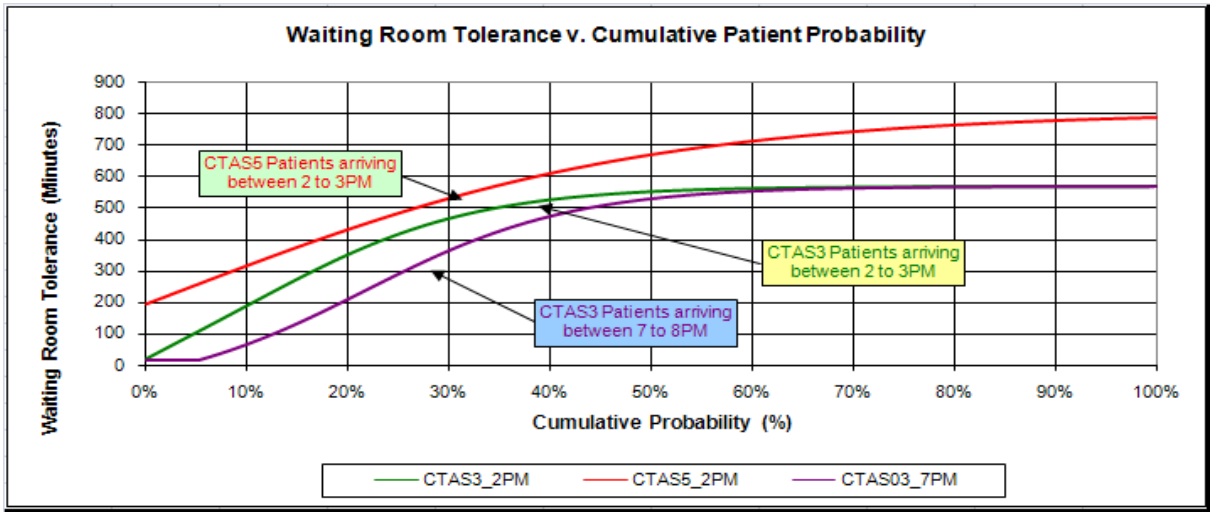


Figure B-1 Examples of Patients' LWBS Behaviour

The figure above shows that in general, CTAS 3 patients who arrive in the ED between 2PM and 3PM have higher waiting room tolerance than the CTAS 3 patients who arrive in the ED between 7PM and 8PM, but have lower waiting room tolerance than CTAS 5 patients who arrive in the ED between 2PM and 3PM.

For modeling patients who choose to LWBS while waiting for a 1st EP assessment in an ED bed, a fixed probability is applied to each CTAS patient group to determine which patients will choose to LWBS. The LWBS rates by CTAS score, which were derived from the patient record analysis, are shown in the Table B-9. (e.g. 2.13% of CTAS 3 patients, who have got an ED bed, will choose to LWBS).

Table B-9 Historical Rates of LWBS from a Bed by CTAS Score

CTAS	1	2	3	4	5
LWBS (%)	0.0%	0.6%	2.1%	3.4%	3.2%

The time that patients who LWBS from a bed spend in that bed before leaving is sampled in the model from distributions based on historical data analysis (see Table B-10 for the distributions used).

Table B-10 Parameters of Patients who LWBS from a Bed

CTAS	1	2	3	4	5
Waiting Time on ED beds before LWBS	0	Exponential(99.48)	Weibull(102.81,0.93)	Exponential(86.10)	Weibull(75.22,1.0)

B.3 Model Input Data

In addition to the model structural assumptions defined in the previous section, the model's behaviour derives from the following input data parameters:

Patient Arrival Pattern

Patients arrive in the ED via one of the following two options: 1) by EMS (Emergency Medical Services); or 2) walk in. The patient arrival rate with either option varies significantly by hour of day, and less significantly by day of week. Figure B-2 shows the hourly arrival rates derived from the historical patient record analysis.

Hour of Day	EMS	Walk-in
0	2.54	3.61
1	1.90	2.69
2	1.89	2.29
3	2.00	2.02
4	1.33	1.59
5	1.25	1.91
6	1.23	2.47
7	1.46	3.45
8	2.04	5.75
9	2.76	7.13
10	3.15	6.99
11	3.63	7.32
12	3.49	7.29
13	3.62	7.22
14	3.78	7.09
15	3.54	6.76
16	3.48	6.20
17	3.74	5.96
18	3.65	6.15
19	3.16	6.65
20	3.29	6.26
21	3.16	6.42
22	2.86	5.88
23	2.98	4.50
Grand Total	65.93	123.60
Note: Hour 2 is from 2am to 3am		

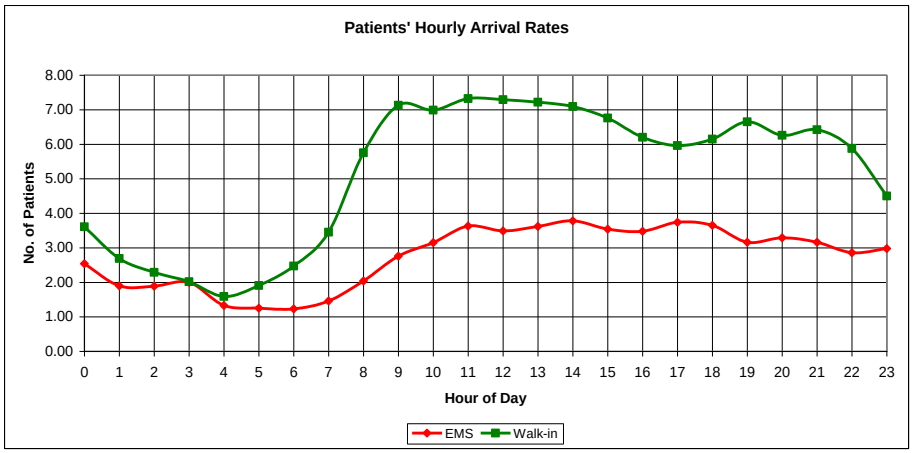


Figure B-2 Patient Arrival Rates

Patient Acuity

The patient mix in terms of CTAS score is defined in Figure B-3 (e.g. CTAS 4 patients comprise 16.3% of all patients).

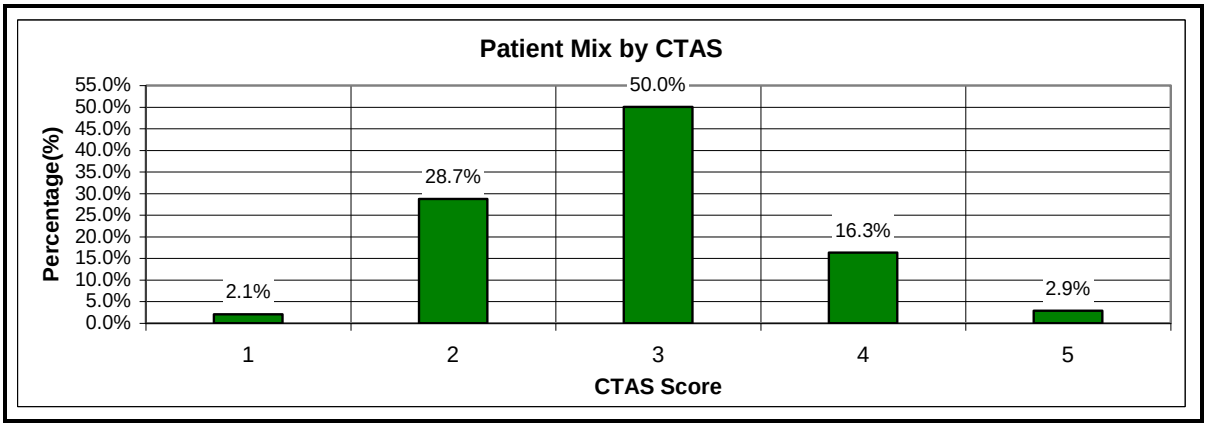


Figure B-3 Patient Acuity

The ED bed requirement (by CTAS) is defined in Table B-11 (e.g. 15.7% of CTAS 3 patients require a monitored ED bed at triage when the MET area is closed).

Table B-11 Patient Bed Requirements

CTAS	MET Close			MET Open			
	Trauma	Monitored	Non-Monitored	Trauma	Monitored	Non-Monitored	MET Beds
1	85.7%	10.9%	3.4%	86.2%	10.3%	1.6%	1.9%
2	19.8%	56.9%	23.3%	19.9%	47.0%	27.3%	5.8%
3	1.0%	15.7%	83.3%	1.4%	10.7%	55.0%	32.9%
4	0.1%	1.4%	98.5%	0.1%	0.5%	16.4%	83.0%
5	0.0%	0.6%	99.4%	0.0%	0.0%	10.6%	89.4%
Grand Total	7.6%	23.3%	69.2%	8.3%	19.5%	38.1%	34.1%

DI, Lab Tests and Trial of Therapy

In the model, no distinction is made between DI and lab tests. These are lumped together and considered simply as "tests". To differentiate lab tests before and after 1st EP assessment, the lab tests ordered before 1st EP assessment are termed as "pre-lab tests", and the lab tests ordered after 1st EP assessment are termed as repeated lab tests. Table B-12 shows the percentages of patients needing tests at different points (as well as trial of therapy after 1st assessment), depending on whether the patient is being treated in the main ED or the MET area. The pre-lab tests and repeated lab tests rates came from the DI and CLS data analysis, and the trial of therapy rates came from the EP activity observation and analysis (e.g. 37.9% of CTAS 3 patients in the MET area need to take pre-lab tests, and none of CTAS 1 patient in either the main ED or the MET area needs trial of therapy).

Table B-12 Patient Treatment Needs (Tests and Trial of Therapy)

CTAS	Pre-lab Test		Repeated LabTest		Trial of Therapy	
	Main ED	MET	Main ED	MET	Main ED	MET
1	17.6%	14.3%	54.4%	71.4%	0.0%	0.0%
2	23.2%	37.5%	74.4%	17.4%	10.2%	22.2%
3	31.4%	37.9%	61.5%	8.0%	10.4%	18.6%
4	21.4%	38.3%	32.7%	5.1%	4.2%	13.2%
5	22.3%	36.1%	17.0%	4.1%	50.0%	20.0%
Grand Total	27.0%	37.9%	64.4%	6.8%	10.24%	18.45%

Need for Consultation and Admission

Similar to the approach above for randomly generating patient test needs, the probability that a patient needs an admitting service consult, and the probability of being admitted following such a consult, depend on CTAS and on the treatment location (main versus MET). The values used in the model are show in Table B-13 (e.g. for CTAS 3 patients in the main ED, 40.1% will need a consultant and 69.8% of these will need to be admitted to the main hospital).

Table B-13 Patient Consult and Admission Rates

CTAS	Main ED		MET	
	Consulted	Admitted	Consulted	Admitted
1	78.4%	88.0%	0.0%	0.0%
2	54.0%	80.0%	28.6%	39.8%
3	40.1%	69.8%	15.0%	46.9%
4	21.0%	58.8%	8.3%	34.8%
5	10.5%	62.5%	7.2%	17.0%
Grand Total	46.3%	75.6%	11.9%	41.1%

Waiting Time for Consultants and Inpatient Beds

Waiting time for MD (Medical Doctor) consultants and inpatients beds are derived from the analysis of historical patient records with the values used in the model shown in Table

B-14 (e.g. the waiting times of CTAS 3 patients for an inpatient bed can be estimated using a gamma distribution with a mean of 243.4 minutes.)

Table B-14 Patient Consult and Boarding Delay Distributions

Time Type	Mean (Minute)				
	CTAS 1	CTAS 2	CTAS 3	CTAS 4	CTAS 5
Waiting for Consultants Time					
For admitted patients	119.7	214.6	251.0	196.1	179.9
For discharged patients	444.0	321.0	252.2	100.5	94.2
Waiting for InPt Beds Time	146.2	226.8	243.4	185.8	290.6

Time Type	Probability Distribution				
	CTAS 1	CTAS 2	CTAS 3	CTAS 4	CTAS 5
Waiting for Consultants Time					
For admitted patients	Weibull(77.0,0.59)	Weibull(180,0.75)	Weibull(214,0.76)	Weibull(107,0.52)	Weibull(153,0.75)
For discharged patients	Gamma(213.0,2.30)	Weibull(323,1.01)	Weibull(231,0.85)	Weibull(56,0.53)	Weibull(23,0.37)
Waiting for InPt Beds Time	Exponential(146.16)	Gamma(247.9,0.92)	Gamma(219.08,1.11)	Gamma(131.31,1.42)	Lognormal(261,369.1)

B.4 Model Verification and Validation

After the model was built, a combination of techniques was used to verify and validate the model. These techniques included ensuring that the patient went to the proper place and was treated by the appropriate personnel in the proper order and developing detailed model documents to be reviewed by other members of the research team. Following this, data files created from pilot runs of the model were analyzed to ensure high face validity. After that, the simulation model and the results generated from pilot runs were compared to the historical data collected from the real ED's operations. The pilot run used for model validation involved 10 replications with a replication length of 485 days including a warm up period of 120 days (i.e. results are based on 10 simulated years of operation). Table B-15 shows the results from the pilot run and compares these with historical data from the real system. The results in Table B-15 show that the model results match well with those from the patient historical data analysis.

Table B-15 Pilot Simulation Run Results Compared with Historical Data

Performance Indicators	Unit	Historical Data	Simulation Result	
			95% CI Low	95% CI High
Waiting Room Time	Minute	103.5	99.5	103.9
EP Time	Minute	79.2	78.7	82.4
Work up Time	Minute	135.0	131.5	132.4
Consultation Time	Minute	234.1	232.6	235.6
Boarding Time	Minute	227.8	226.5	228.7
LoS Total	Minute	458.7	456.7	464.2
LoS Admit	Minute	754.0	751.2	761.8
LoS Discharge	Minute	355.6	350.7	357.9
LWBS % Total		8.86%	8.61%	9.03%
CTAS1 LWBS WR%	Percent	0.00%	0.00%	0.00%
CTAS2 LWBS WR%	Percent	2.90%	2.62%	2.88%
CTAS3 LWBS WR%	Percent	10.07%	9.57%	10.15%
CTAS4 LWBS WR%	Percent	7.38%	7.16%	7.72%
CTAS5 LWBS WR%	Percent	4.84%	4.23%	5.09%
LWBS WR% Total	Percent	7.20%	6.86%	7.28%
CTAS1 LWBS Bed%	Percent	0.00%	0.00%	0.00%
CTAS2 LWBS Bed%	Percent	0.61%	0.54%	0.64%
CTAS3 LWBS Bed%	Percent	2.13%	1.90%	1.98%
CTAS4 LWBS Bed%	Percent	3.36%	3.12%	3.34%
CTAS5 LWBS Bed%	Percent	3.21%	2.79%	3.41%
LWBS Bed% Total	Percent	1.84%	1.73%	1.79%

Appendix C Confidence Intervals with Specified Precision

The half-length H of a 100(1-α)% confidence interval (CI) for a mean θ, based on the t distribution, is given by (Banks et al., 2005:395, Eq. 11.12)

$$H = t_{\alpha/2,\, R-1}\frac{S}{\sqrt{R}} \quad (\text{Eq. C-1})$$

Where S is the sample standard deviation and R is the number of samples or number of replications in our case.

Suppose that an error criterion ϵ is specified, the following analytical procedure can be used to determine the number of replications for estimating the mean θ, which will be within the 100(1-α)% confidence interval.

Assume that an initial sample of size R_0 ($R_0 \geq 2$) independent replications has been generated. The R_0 replications will be used to obtain an initial estimate S_0 of the population standard deviation σ. To meet the half-length criterion, a sample size R must be chosen such that $R \geq R_0$ and

$$H = t_{\alpha/2,\, R-1}\frac{S_0}{\sqrt{R}} \leq \epsilon \quad (\text{Eq. C-2})$$

Solving for R in Inequality (C-2) shows that R is the smallest integer satisfying $R \geq R_0$ and

$$R \geq \left(\frac{t_{\alpha/2,R-1}S_0}{\epsilon}\right)^2 \quad (\text{Eq. C-3})$$

Since $t_{\alpha/2,\, R-1} \geq z_{\alpha/2}$, an initial estimate for R is given by

$$R \geq \left(\frac{z\alpha / 2 S_0}{\epsilon} \right)^2 \quad \text{(Eq. C-4)}$$

Where $z_{\alpha/2}$ is the 100(1-α/2) percentage point of the standard normal distribution. After determining the final sample size, R, collect $R-R_0$ additional observations (i.e. make $R-R_0$ additional replications, or start over and make R total replications) and calculate the half-length of the 100(1-α)% CI for θ. If the confidence interval is too large, the procedure may be repeated, using Inequity (C-3), to determine an even larger sample size.

Example C.1

Suppose that it is desired to use the FMC ED DES model to estimate patients' average time in the waiting room to within ± 2.53 minutes with probability 95%. An initial sample of size $R_0 = 4$ is taken, with the results given in Table C-1. For each run, the replication length is 485 days including a warm-up period of 120 days.

Table C-1 Results of Four Independent Runs of the FMC ED DES Model

Replication, r	1	2	3	4
Waiting Room Time (Minute)	97.44	102.94	100.47	104.35

An initial estimate of the population standard deviation is $S_0 = 3.0328$. The error criterion is $\epsilon = 2.53$, and the confidence coefficient is $1-\alpha = 0.95$. From Inequality (C-4), the final sample size must be at least as large as

$$\left(\frac{Z_{0.025} S_0}{\epsilon} \right)^2 = \left(\frac{1.96 * 3.0328}{2.53} \right)^2 = 5.51$$

As the sample size has to be integer, so the smallest sample size has to be 6.

Next, Inequity (C-3) can be used to test possible candidates (R =6, 7, 8,…) for final sample size, as shown in Table C-2.

Table C-2 Results with Additional Independent Runs of the FMC ED DES Model

R	6	7	8	9
$t_{0.025,R-1}$	2.57	2.45	2.36	2.31
$(t_{0.025,R-1}s_0/\epsilon)^2$	9.5	8.6	8.0	7.6

Thus, R = 9 is the smallest integer satisfying Inequity (C-3), so R - R_0= 9 – 4 = 5 additional replications are needed. The results of the total 9 replications are shown in Table C-3.

Table C-3 Results of Nine Independent Runs of the FMC ED DES Model

Replication, r	1	2	3	4	5	6	7	8	9
Waiting Room Time (Minute)	97.44	102.94	100.47	104.35	107.09	100.93	101.32	104.75	100.01

With 9 replications, the estimated population standard deviation is s = 2.9232. The 95% confidence interval half-length can be calculated by (C-1) as:

$H = t_{0.025,8} * s/\sqrt{9} = 2.31*2.9232/\sqrt{9}= 2.25< 2.5$

So with 9 replications, the half-length H can be ensured to be as small as desired.

Reference

Banks, J., Carson II, J.S., Nelson, B.L., and Nicol, D.M. (2005). Discrete-event system simulation (4th ed.). Upper Saddle River, NJ: Prentice-Hall.

Appendix D Parameters of FMC ED NN Models

D.1 Waiting Room Time NNs

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WR_CTAS1
Input Layer	(Bias)	-.092	-1.937	
	Patient Arrival Rate	.919	1.163	
	Average Boarding Time	.056	.020	
Hidden Layer 1	(Bias)			1.401
	H(1:1)			.896
	H(1:2)			1.671

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WR_CTAS2
Input Layer	(Bias)	-1.012	-.703	
	Patient Arrival Rate	-1.662	.737	
	Average Boarding Time	-.132	.024	
Hidden Layer 1	(Bias)			.308
	H(1:1)			-.667
	H(1:2)			1.198

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WR_CTAS3
Input Layer	(Bias)	1.120	.525	
	Patient Arrival Rate	1.794	-.715	
	Average Boarding Time	.131	-.012	
Hidden Layer 1	(Bias)			.020
	H(1:1)			.804
	H(1:2)			-.883

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WR_CTAS4
Input Layer	(Bias)	-1.308	.566	
	Patient Arrival Rate	-1.671	-.590	
	Average Boarding Time	-.120	.026	
Hidden Layer 1	(Bias)			.059
	H(1:1)			-.869
	H(1:2)			-1.091

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WR_CTAS5
Input Layer	(Bias)	1.068	.212	
	Patient Arrival Rate	1.575	-.351	
	Average Boarding Time	.155	.007	
Hidden Layer 1	(Bias)			.158
	H(1:1)			.559
	H(1:2)			-1.989

D.2 Work-up Time of Consulted Patients NNs

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WU_Consult_CTAS1
Input Layer	(Bias)	.949	-.618	
	Patient Arrival Rate	-.052	-.823	
	Average Boarding Time	-.139	-.008	
Hidden Layer 1	(Bias)			.505
	H(1:1)			-1.588
	H(1:2)			-1.754

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WU_Consult_CTAS2
Input Layer	(Bias)	1.016	-.488	
	Patient Arrival Rate	-1.154	-1.060	
	Average Boarding Time	-.122	-.315	
Hidden Layer 1	(Bias)			.206
	H(1:1)			-.856
	H(1:2)			-.806

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WU_Consult_CTAS3
Input Layer	(Bias)	-.344	-1.424	
	Patient Arrival Rate	.435	-1.284	
	Average Boarding Time	.199	-.068	
Hidden Layer 1	(Bias)			2.554
	H(1:1)			3.596
	H(1:2)			2.634

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WU_Consult_CTAS4
Input Layer	(Bias)	-.276	.019	
	Patient Arrival Rate	.890	-1.079	
	Average Boarding Time	.089	.112	
Hidden Layer 1	(Bias)			-.082
	H(1:1)			-.715
	H(1:2)			.791

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WU_Consult_CTAS5
Input Layer	(Bias)	-.352	.054	
	Patient Arrival Rate	-.726	.572	
	Average Boarding Time	-.236	.153	
Hidden Layer 1	(Bias)			-.036
	H(1:1)			-.756
	H(1:2)			.536

D.3 Work-up Time of Non-consulted Patients NNs

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WU_NoConsult_CTAS1
Input Layer	(Bias)	.078	-.646	
	Patient Arrival Rate	-.408	-1.125	
	Average Boarding Time	-.001	-.159	
Hidden Layer 1	(Bias)			-.283
	H(1:1)			-.820
	H(1:2)			-.971

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WU_NoConsult_CTAS2
Input Layer	(Bias)	.585	-.145	
	Patient Arrival Rate	-.803	-.467	
	Average Boarding Time	-.077	-.125	
Hidden Layer 1	(Bias)			.291
	H(1:1)			-1.116
	H(1:2)			-.959

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WU_NoConsult_CTAS3
Input Layer	(Bias)	1.798	-.389	
	Patient Arrival Rate	2.007	.777	
	Average Boarding Time	-.113	.153	
Hidden Layer 1	(Bias)			1.299
	H(1:1)			-1.282
	H(1:2)			2.567

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WU_NoConsult_CTAS4
Input Layer	(Bias)	-1.001	.561	
	Patient Arrival Rate	1.566	2.447	
	Average Boarding Time	.027	.018	
Hidden Layer 1	(Bias)			-.214
	H(1:1)			-.786
	H(1:2)			-.549

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	WU_NoConsult_CTAS5
Input Layer	(Bias)	.022	-.179	
	Patient Arrival Rate	.383	.997	
	Average Boarding Time	-.082	.135	
Hidden Layer 1	(Bias)			.170
	H(1:1)			.691
	H(1:2)			1.149

D.4 EP Time NNs

Predictor		Predicted				
		Hidden Layer 1				Output Layer
		H(1:1)	H(1:2)	H(1:3)	H(1:4)	EPTime_CTAS1
Input Layer	(Bias)	-1.425	2.058	-.158	-1.105	
	Patient Arrival Rate	-.878	1.556	.160	-.079	
	Average Boarding Time	.081	.145	-.280	.553	
Hidden Layer 1	(Bias)					-2.016
	H(1:1)					-.813
	H(1:2)					1.460
	H(1:3)					.331
	H(1:4)					-.612

Predictor		Predicted				
		Hidden Layer 1				Output Layer
		H(1:1)	H(1:2)	H(1:3)	H(1:4)	EPTime_CTAS2
Input Layer	(Bias)	-.707	.697	.771	-.773	
	Patient Arrival Rate	.792	1.528	-.514	-1.159	
	Average Boarding Time	.288	.222	.429	-.046	
Hidden Layer 1	(Bias)					-.942
	H(1:1)					-1.145
	H(1:2)					-2.141
	H(1:3)					-.927
	H(1:4)					-4.065

Predictor		Predicted				
		Hidden Layer 1				Output Layer
		H(1:1)	H(1:2)	H(1:3)	H(1:4)	EPTime_CTAS3
Input Layer	(Bias)	.392	-1.489	-2.240	2.186	
	Patient Arrival Rate	-.060	-1.313	-1.169	1.570	
	Average Boarding Time	.245	-.022	-.500	-.206	
Hidden Layer 1	(Bias)					-.001
	H(1:1)					-2.958
	H(1:2)					-1.904
	H(1:3)					-.772
	H(1:4)					-1.041

Predictor		Predicted				
		Hidden Layer 1				Output Layer
		H(1:1)	H(1:2)	H(1:3)	H(1:4)	EPTime_CTAS4
Input Layer	(Bias)	1.043	-.751	1.300	-.471	
	Patient Arrival Rate	1.336	-1.504	.485	.320	
	Average Boarding Time	.071	-.237	.378	-.257	
Hidden Layer 1	(Bias)					-.167
	H(1:1)					1.462
	H(1:2)					.754
	H(1:3)					.636
	H(1:4)					1.930

Predictor		Predicted				
		Hidden Layer 1				Output Layer
		H(1:1)	H(1:2)	H(1:3)	H(1:4)	EPTime_CTAS5
Input Layer	(Bias)	.138	1.285	1.183	-.613	
	Patient Arrival Rate	.732	.865	-.643	.255	
	Average Boarding Time	-.193	.120	.369	.268	
Hidden Layer 1	(Bias)					-.296
	H(1:1)					.813
	H(1:2)					.984
	H(1:3)					-1.040
	H(1:4)					-.532

D.5 LWBS NNs

Predictor		Predicted				
		Hidden Layer 1				Output Layer
		H(1:1)	H(1:2)	H(1:3)	H(1:4)	LWBSRate_CTAS1
Input Layer	(Bias)	-.401	-.473	-.402	.331	
	Patient Arrival Rate	-.585	.618	.398	.042	
	Average Boarding Time	.498	-.205	.459	-.556	
Hidden Layer 1	(Bias)					.135
	H(1:1)					-.432
	H(1:2)					.435
	H(1:3)					.614
	H(1:4)					-.052

Predictor		Predicted				
		Hidden Layer 1				Output Layer
		H(1:1)	H(1:2)	H(1:3)	H(1:4)	LWBSRate_CTAS2
Input Layer	(Bias)	-.693	-.336	.537	-1.043	
	Patient Arrival Rate	.759	-1.263	.803	.438	
	Average Boarding Time	.026	-.065	.055	.389	
Hidden Layer 1	(Bias)					.790
	H(1:1)					1.286
	H(1:2)					-.883
	H(1:3)					-.563
	H(1:4)					.249

Predictor		Predicted				
		Hidden Layer 1				Output Layer
		H(1:1)	H(1:2)	H(1:3)	H(1:4)	LWBSRate_CTAS3
Input Layer	(Bias)	-.613	.479	-1.276	-.168	
	Patient Arrival Rate	.627	-.511	-1.718	-.112	
	Average Boarding Time	.103	.132	-.115	.045	
Hidden Layer 1	(Bias)					.473
	H(1:1)					.850
	H(1:2)					-.732
	H(1:3)					-.712
	H(1:4)					.870

Predictor		Predicted				
		Hidden Layer 1				Output Layer
		H(1:1)	H(1:2)	H(1:3)	H(1:4)	LWBSRate_CTAS4
Input Layer	(Bias)	.111	-.219	-.734	-.257	
	Patient Arrival Rate	.277	.334	.793	-1.331	
	Average Boarding Time	-.016	-.024	-.029	-.073	
Hidden Layer 1	(Bias)					.360
	H(1:1)					-.365
	H(1:2)					-.114
	H(1:3)					.979
	H(1:4)					-.896

Predictor		Predicted				
		Hidden Layer 1				Output Layer
		H(1:1)	H(1:2)	H(1:3)	H(1:4)	LWBSRate_CTAS5
Input Layer	(Bias)	.542	.569	.678	.503	
	Patient Arrival Rate	-.596	1.416	-.631	-.056	
	Average Boarding Time	-.099	.103	.087	-.246	
Hidden Layer 1	(Bias)					.478
	H(1:1)					-.568
	H(1:2)					.600
	H(1:3)					-.821
	H(1:4)					-.021

Appendix E Simulation Metamodels for Chapter 8

E.1 The CHR HW NN-based Simulation Metamodel

Scope of the Simulation Metamodel

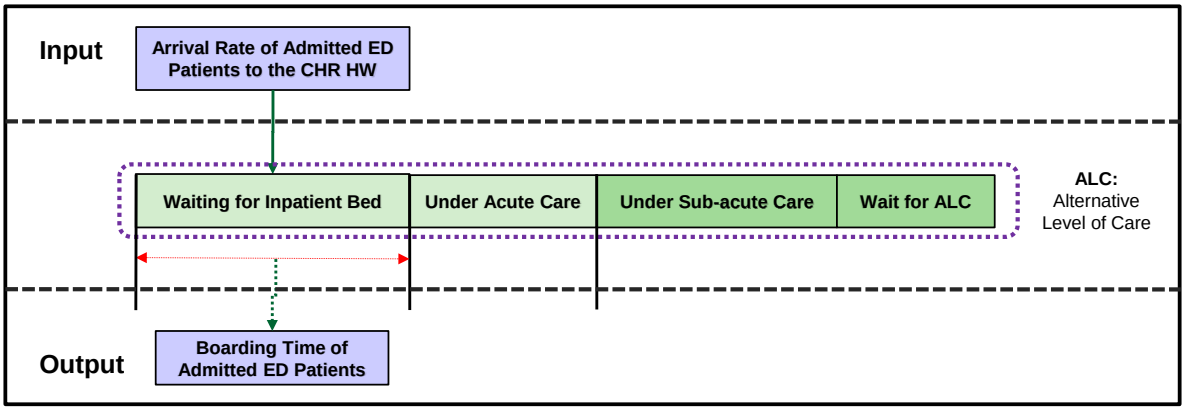


Figure E-1 Scope of the CHR HW Simulation Metamodel

Valid Range

- Arrival Rate of Admitted ED Patients to the CHR HW: 70% to 160%

Performance Target

Table E-1 Performance Targets of the CHR HW NN-based Simulation Metamodel

Simulation Model Output	Performance Measure	Performance Target (Minute)
Boarding Time of Admitted ED Patients	MAD	5

5 minutes MAD is about 2.5% MAPD when the boarding time of admitted ED patients is 30 minutes or longer.

Architecture of the Final NN Model

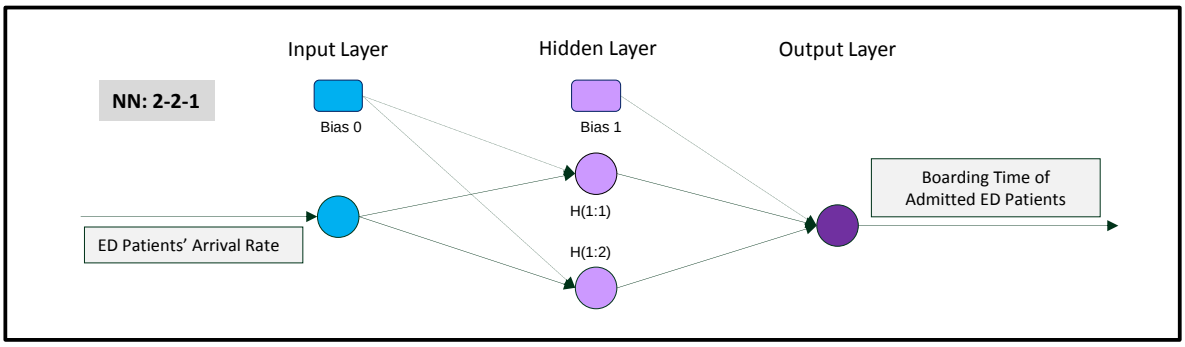


Figure E-2 Architecture of the CHR HW NN-based Simulation Metamodels

Parameters of the Trained NN Model

Table E-2 Parameters of the Trained 2-2-1 NN Model (CHR HW)

2-2-1 Architecture				
Predictor		**Predicted**		
		Hidden Layer 1		**Output Layer**
		H(1:1)	**H(1:2)**	**Boarding Time of Admitted ED Patients**
Input Layer	(Bias)	1.928	.057	
	ED Patients Arrival Rate	3.004	1.506	
Hidden Layer 1	(Bias)			-.261
	H(1:1)			.662
	H(1:2)			.610

Performance of the Final NN-based Simulation Metamodel

Table E-3 Performance Evaluation of the 2-2-1 NN Model

Simulation Model Output	Data Sets	Result (Min)	Target (Min)	Evaluation (Individual)	Evaluation (Overall)
ED_Boarding time	Training Set	1.762	5.00	Pass	Pass
	Test Set	1.761	5.00	Pass	
	Total	1.761	5.00	Pass	

E.2 The FMC ED NN-based Simulation Metamodels

Scope of the Simulation Metamodels

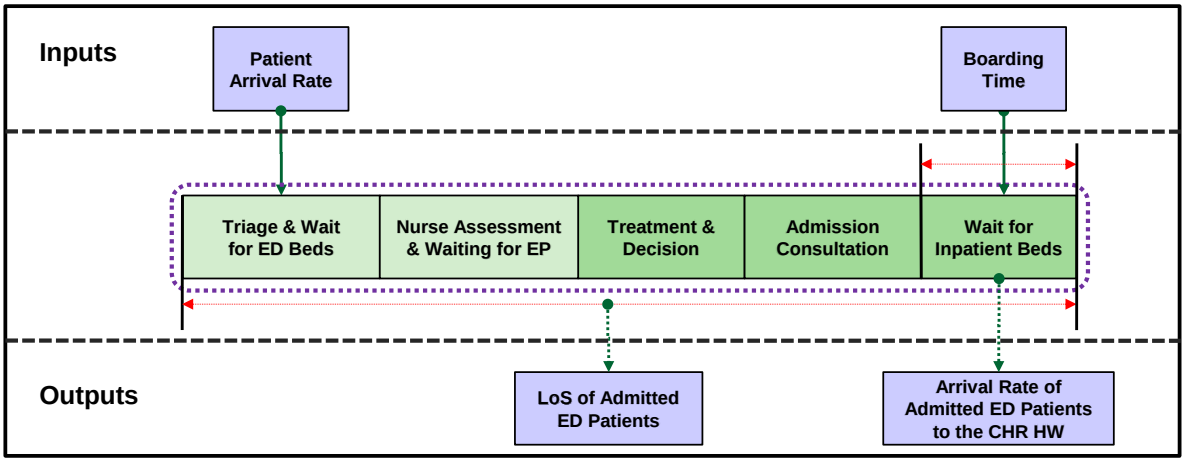


Figure E-3 Scope of the FMC ED Simulation Metamodels

Valid Ranges

- Patient Arrival Rate: 80% to 150%
- Boarding Time of admitted ED Patients: 60% to 160%

Performance Target

Table E-4 Performance Targets of the FMC ED NN-based Simulation Metamodels

Simulation Model Output	Performance Measure	Performance Target (%)
LoS of Admitted ED Patients	MAPD	2.500
Arrival Rate of Admitted ED Patients to the CHR HW	MAPD	2.500

Architecture of the Final NN Model

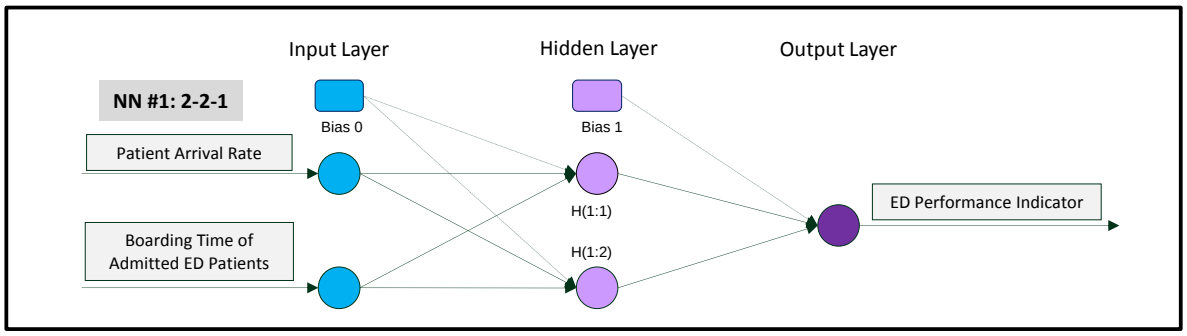


Figure E-4 Architecture of the FMC ED NN-based Simulation Metamodels

Parameters of the Trained NN Model

Table E-5 Parameters of the Trained 2-2-1 NN Models (FMC ED)

2-2-1 Architecture				
Predictor		**Predicted**		
		Hidden Layer 1		**Output Layer**
		H(1:1)	**H(1:2)**	**Arrival Rate of Admitted ED Patients to the CHR HW**
Input Layer	(Bias)	-1.945	.324	
	Patient Arrival Rate	-1.210	.096	
	Boarding Time of Admitted ED Patients	-.070	-.142	
Hidden Layer 1	(Bias)			-2.799
	H(1:1)			-2.855
	H(1:2)			1.711

Predictor		**Predicted**		
		Hidden Layer 1		**Output Layer**
		H(1:1)	**H(1:2)**	**LoS of Admitted ED Patients**
Input Layer	(Bias)	-0.312	-1.293	
	Patient Arrival Rate	-0.127	-2.083	
	Boarding Time of Admitted ED Patients	-.120	-.043	
Hidden Layer 1	(Bias)			-1.541
	H(1:1)			-4.698
	H(1:2)			-0.441

Performance of the Final NN-based Simulation Metamodel

Table E-6 Performance Evaluation of the 2-2-1 NN Model

Simulation Model Output	Data Sets	Result (%)	Target (%)	Evaluation (Individual)	Evaluation (Overall)
LoS of Admitted ED Patients	Training Set	0.342	2.500	Pass	Pass
	Test Set	0.436	2.500	Pass	
	Total	0.413	2.500	Pass	
Arrival Rate of Admitted ED Patients to the CHR HW	Training Set	0.160	2.500	Pass	Pass
	Test Set	0.156	2.500	Pass	
	Total	0.157	2.500	Pass	