12-1999

A study of the reduction of simulation modeling development time

Ammar Mohamed Aamer

To the Graduate Council:

I am submitting herewith a thesis written by Ammar Mohamed Aamer entitled "A study of the reduction of simulation modeling development time." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Industrial Engineering.

Rapindir Sawhney, Major Professor

We have read this thesis and recommend its acceptance:

Kenneth E. Kirby, Zayne Claycombe

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Ammar M. Aamer entitled "A Study of the Reduction of Simulation Modeling Development Time." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Industrial Engineering.

Rapinder Sawhney

Rapinder Sawhney, Major Professor

We have read this thesis
and recommend its acceptance:

Kenneth E. Kirby

Wayne Claycombe

Accepted for the Council:

Anne Mayhew

Associate Vice Chancellor and
Dean of the Graduate School

A STUDY OF THE REDUCTION OF SIMULATION MODELING DEVELOPMENT TIME

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Ammar M. Aamer

December, 1999

DEDICATION

This thesis is dedicated to my parents

whose love, support, encouragement, and memories

have inspired me to be who I am.

ACKNOWLEDGMENTS

I wish to express special gratitude to Dr. Rapinder Sawhney for supporting me to continue my education and being my respected friend as well as major professor and committee chairman. His valued advice, guidance, and encouragement were instrumental in completing this thesis.

I also wish to thank the other committee members, Dr. Ken Kirby and Dr. Wayne Calycombe, who gave valued education and advice, and who were infinitely tolerant.

Appreciation is also in order for Michel Pittman, the project principal contact who helped throughout the simulation project.

I also want to thank my parents, brothers, sisters, nephews, nieces, and the rest of my family for their constant love and support throughout school and life.

The final acknowledgement is reserved for my lovely fiancée without whom this thesis would never have been possible.

ABSTRACT

This paper presents new ideas dealing with simulation model development that are derived from the principles of lean manufacturing, for example, the concept of treating the time that is spent in production activities as lead-time, and the feasibility of reducing the lead-time through different mechanisms, which are presented in a framework [44]. One of the main obstacles in developing simulation model is time. Simulation lead-time is composed of nine steps of simulation model development. These steps are: problem definition, establishing boundaries, establishing variables, data collection, model development, verification & validation, documentation, experimentation, and implementation. This paper presents the idea of treating the time spent in simulation modeling development as a lead-time. At the same time, it presents a new framework to reduce lead-time, which has never been addressed before.

A new framework to reduce the simulation modeling development long lead-time similar to the Toyota production framework for reducing the production lead-time will be presented in this paper. The framework developed as a result of an actual simulation case study, which took place at a local company, and which took a very long lead-time. The framework was composed of different steps, techniques, and mechanisms that should reduce simulation modeling development lead-time every time a simulation project is conducted. One of the

goals of this framework is to reduce one of the main obstacles of simulation model, which is the long lead-time. One of the new mechanisms that is presented in this framework is a geographical distributed communication tool, which is called NetMeeting. This tool is an application of the concept of distributed and Web-Based simulations.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE PAGE

CHAPTER I

PROBLEM STATEMENT

A. Background

In today's world, competition is forcing manufacturers to modify their facilities much faster than in the past. If we were to go back to the early 1910s, when the auto industry was initiated, we would find that it was based on craft production, which was a specialist car builder. Due to Increasing demands, requirements of better productivity, and customer needs in the auto industry, the Ford era emerged. This era was based on the idea of mass production. Thereafter, the auto industry expanded significantly in terms of the variety of products and the complexity of processes. This rapid change has forced manufacturers to look for better ways to be more flexible to adapt the rapid change; therefore, the Toyota era, based on the principles of lean production, met the variation in customer demands and the high productivity need. Table 1 presents a comparison between the old era and how the auto industry has been forced to seek fast solutions to survive in today's competitive world.

Table 1: Comparison between old and new eras [25].

Factor	Craft	Mass	Lean
Labor	Skilled	Unskilled	Skilled
Capital Intensity	Low	High	Lower
Productivity	Low	High	Higher
Interchangeable Parts	No	Yes	Yes
Product Price	High	Low	Low
Product Quality	High	Medium	High
Worker Motivation	High	Low	Medium
Vertical Integration	Low	High	low

It has been known historically that the industrial world has been forced to compete due to the following reasons:

1. Competitive products are continuously being introduced to the market;
2. New processes;
3. New services;
4. New relationships with customers;
5. New relationships with suppliers; and
6. New organization forms.

Evidently, the rapid change and high competition in the market these days have forced the decision-makers to seek new decision tools which mean saving money and time: "such jobs that used to take 100 hours can be done in five." The following are some examples of the decision tools:

1. Enterprise Resource Planning (ERP);
2. Business Process Reengineering (BPR);
3. Simulation modeling; and
4. Others.

At the same time, the Next Generation Manufacturing (NGM) initiative consisting of industry, academics, and government have established via imperatives, the characteristics of the next generation of manufacturers. These imperatives include four different categories: people-related, business-process related, technology-related, and integration-related imperatives. One of the primary concepts of the technology-related imperative is pervasive simulation.

When simulation is incorporated into all aspects of manufacturer functions, it is referred to as pervasive simulation, which includes all functions of a life cycle of getting a product to market, including design, engineering analysis, and production. World-class organizations are promoting this pervasive use of simulation because it has the ability to eliminate the following wastes:

1. Reiterative efforts to develop a product or design a process in order to meet the requirements of all functions of an organization;
2. Develop appropriate prototypes; and
3. Destructive testing of products.

Elimination of the above wastes indicates that pervasive simulation is rapidly becoming a necessary tool for competing in today's manufacturing arena. Even though pervasive is a great concept; it is not readily implemented, for three primary reasons:

1. Many people are not familiar with this concept;
2. Development of simulation model is expensive in terms of resources and time requirement; and
3. Life expectancy of the model is very short.

Since simulation resources are quite hard to control, let's focus on the controllable metric, time, and one of the most successful systems in controlling time. The Lean system considers long time processes as nonvalue-added processes, which must be eliminated. In lean manufacturing environments, production time is referred to as production lead-time. The lower the lead-time the more flexible manufacturing responds to market demands; smooth production requires short lead-time. According to Yasuhiro Monden [44], lead-time in manufacturing environment is the time interval from production dispatching to developing of completed products. Lead-time in a manufacturing environment is a component of the following:

- Queue time before processing
- Set up time
- Run time
- Wait time after processing
- Move time

Figure 1 illustrates the components of production lead-time. If we pay close attention to the illustrated production lead-time, we will find that only run time is a value-added process. Everything else is considered nonvalue-added processes. Monden presented in his book entitled Toyota Production System a framework for reducing nonvalue-added processes from the production lead-time. These reductions are based on three steps:

- Reduction of processing time
- Reduction of conveyance time
- Reduction of waiting time

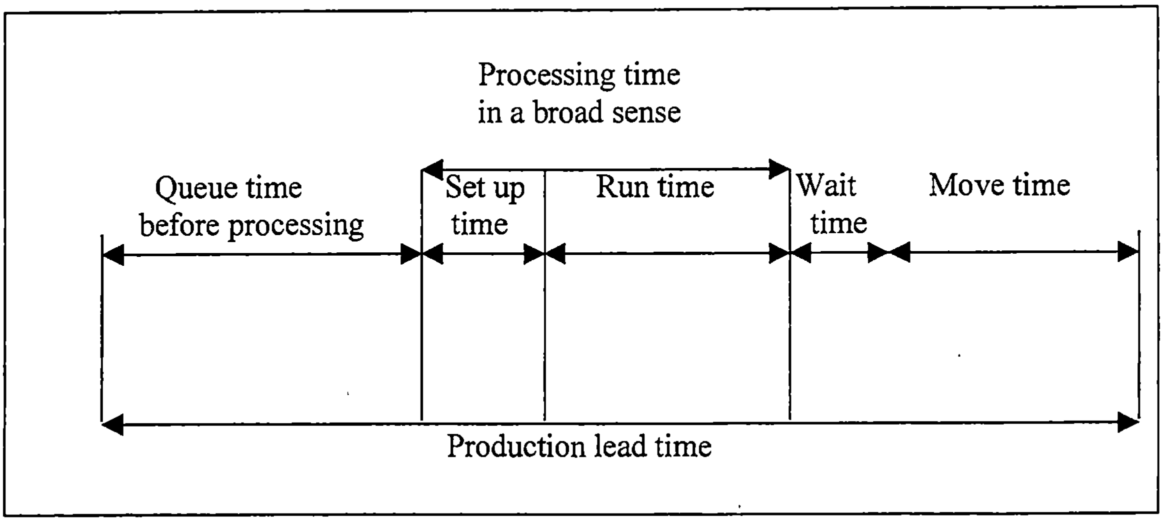

Figure 1: Components of production lead-time [44].

Moreover, each one of the previous steps is composed of different components as illustrated in Figure 2. On the other hand, the process of simulation model goes through the following nine steps:

1. Problem definition;
2. Establishing boundaries;
3. Establishing variables relationships;
4. Data collection;
5. Model development;
6. Verification and validation;
7. Documentation;
8. Experiment analysis; and
9. Implementation.

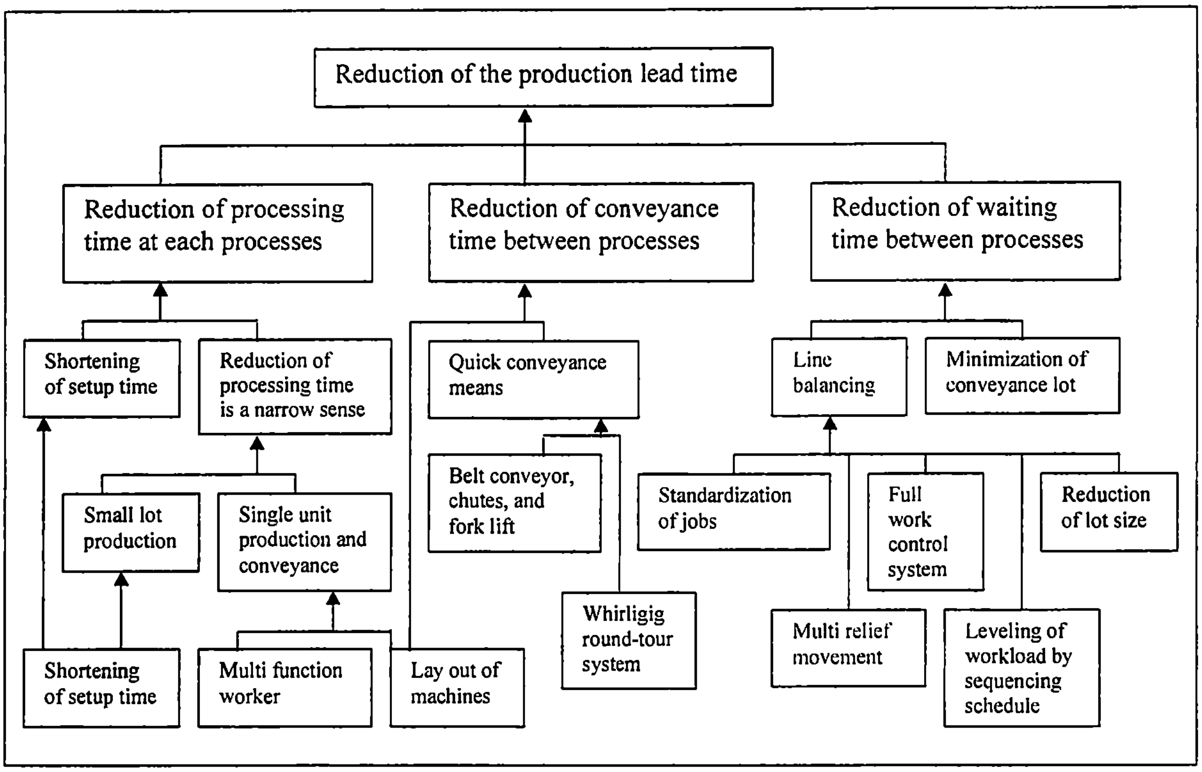

Figure 2: Framework for reducing lead-time [44].

We will consider time interval from the first step of problem definition of a simulation model to the final step of documentation process as lead-time, according to the previous definition. Figure 3 illustrates the components of the new concept of simulation lead-time.

The simulation modeling process has also both value-added and nonvalue-added processes that will be discussed in this paper. Therefore, the less time spent on nonvalue-added processes, the more pervasive the simulation modeling is.

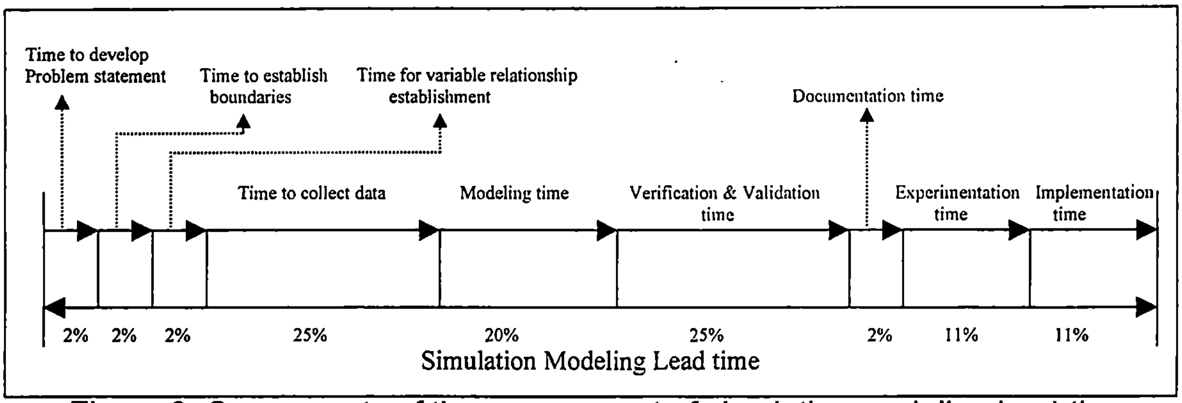

Figure 3: Components of the new concept of simulation modeling lead-time

B. Problem Statement

To achieve the NGM goal of pervasive implementation of simulation modeling, simulation-modeling long lead-time must be reduced. Since it was possible to reduce lead-time in a lean manufacturing environment, it could be possible for us to reduce simulation-modeling lead-time. Therefore, our focus in this research is to follow the steps of a lean system and come up with a new framework that could be used to reduce the long lead-time in the processes of simulation modeling. This framework focuses on the techniques and mechanisms that could be utilized to reduce the time spent in each step of the simulation model. Distributed simulation is one of the mechanisms that is included in the new framework and is known as a way to collaborate both geographical distributed modeler and decision-makers in the development and execution of simulation model by observing, controlling, fine-tuning, and helping to debug simulation models. There has been an evolutionary trend involving the concept of distributed simulation since the early 1980s. Microsoft Company introduced a product that we believe is an applicable application to implement the concept of distributed simulation to achieve our goal of simulation modeling lead-time reduction. This software is called NetMeeting; it will be utilized in this research as a mechanism of distributed simulation.

To illustrate the objectivity of the new lead-time reduction framework, a distributed simulation study will be presented. First, an actual study of a simulation-modeling project that took place at a local company is presented. This study was based on the current traditional simulation model. To better serve our purpose, a distributed simulation study based on the same simulation will be discussed. In this new study, Distributed Manufacturing Model Development Simulation (DMMDS) represented in NetMeeting is utilized. A comparison between both studies in terms of the new concept of simulation modeling lead-time is presented to illustrate the feasibility of the new framework to reduce the simulation modeling lead-time.

CHAPTER II

LITERATURE REVIEW:

EVOLUTION OF DISTRIBUTED SIMULATION SYSTEMS

A. Introduction

The idea of spending more time and effort in the planning stages of a project using simulation to develop all possible scenarios is merely an extension of Deming's philosophy on quality: it is less expensive to get it right the first time. Our goal is to help to implement simulation modeling pervasively in industries. Therefore, the purpose of this study is to accomplish a reduction in simulation modeling lead-time, which is one of the main reasons for its lack of pervasive uses. The focus of this literature review is on the Distributed Simulation models, which is a mechanism to reduce simulation lead-time.

Distributed Simulation has been defined differently by researchers. In the broad view, Distributed Simulation is a multiple simulation software processes that independently execute and interact with each other. It is commonly used for military simulations in practicing and training: " Distributed systems allow various simulations or suites, connected via a high spread network, to operate on a common exercises" [4]. It has been used by the U.S. Department of Defense (DoD) to describe the cooperative utilization of physical distributed simulations toward common objective. We, however, define Distributed Simulation in this research as a Distributed Manufacturing Model Development Simulation

(DMMDS), which is a geographically-distributed communication tool. This allows decision-makers in different locations to develop and fine tune simulation models.

Distributed simulation has been developed through different stages since the early 1980s [4]. In the early uses of simulation, SIMNET was sufficient for small team training events. However, to simulate crews with tactically significant opposing forces, it became readily apparent that augmenting the synthetic battle space required a more scalable solution than simply adding more manual simulator [40]. Therefore, distributed simulation has grown significantly starting with SIMNET and ending with Mobile Cooperative Technique (MCT). Each one of the distributed simulation techniques has provided better features capable of simulating complex simulation models. This development is based on programming language, which has risen dramatically from 100,000 to over 1,000,000 lines. For instance, the World Wide Web gives a predominate approach to modeling through high interactivity, which is an extension to pre-developed distributed simulation. Figure 4 shows the distributed simulation process of development, staring with SIMNET and ending with MCT.

B. SIMNET

SIMNET is a distributed tank simulation funded by the Defense Advance Research Projects Agency (DARPA) in 1989. It made it possible for individual, computerized, tank-crew trainers to be connected over local area networks and

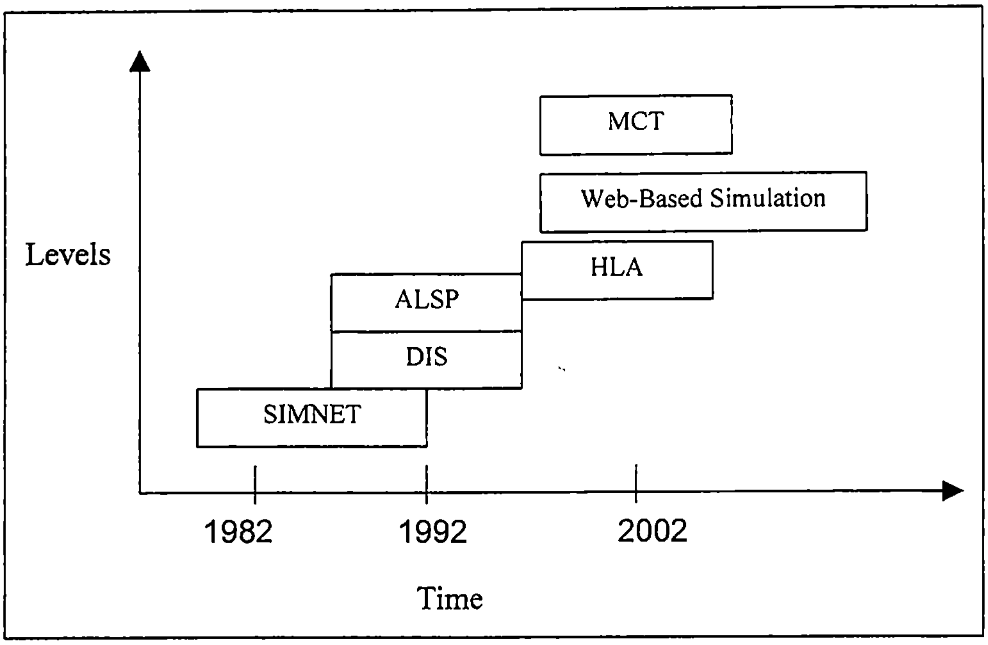

Figure 4: Evolution of Distributed Simulation

Defense Simulation Internets (DSI) to collaborate in a single, virtual battlefield. SIMNET consisted of a series of tank simulators whose viewers were coupled to a 3-D image generator. This image generator ended somehow to a more realistic representation of a battle space environment from a digitized representation of the terrain, soil types, and features in the area of interest. The tank simulators were interconnected via Ethernet and used a common protocol (the SIMNET protocol) to share static data. In each tank simulator, crew positions (gunner, loader, driver, and commander) were represented using a concept referred to as selective fidelity. Rather than real vision blocks or views ports, special image generators provided the illusion of movement. This data was also broadcast to the other simulators participating in the exercise to enable representations of the tank within their local view ports [40]. The success of this model encouraged other researchers to develop more advanced concepts.

C. ALSP

One result of SIMNET is Aggregate Level Simulation Protocol (ALSP), developed in the early 90s. ALSP provides a mechanism for the integration of the existing simulation models to support training via theater-level simulation exercises. ALSP is known for the largest "real world" application of parallel and distributed simulation theory. Despite significant representational differences between ALSP and SIMNET, several of the underlying principles of SIMNET are applicable to ALSP [20]:

- Dynamic configurability. Simulations are permitted to join and depart a simulation exercise without restriction.
- Geographical distribution. Simulations can be located in different geographical locations, yet exercise over the same logic terrain.
- Autonomous entities. Each simulation controls its own resources; first its own weapons, and when hit, conduct damage assessment locally.
- Communication by message passing. Information from one simulation is distributed to all other simulations using message-passing protocol.

In addition, ALSP challenge has several unique requirements beyond those of SIMNET [20]:

- Simulation time management. Typically, simulation time is independent on wall-clock time. For the results of distributed simulation to be "correct," time must be consistent across all "processes" involved in the simulation.
- Data management. The schemes for internal state representation may differ widely among existing simulations. A common representational system and concomitant mapping and control mechanisms are needed.
- Architecture independence. The architectural characteristics, e.g. implementation language, user interface, and time flow mechanism, of existing simulations may differ widely. The architecture implied by ALSP should be unobtrusive to existing architectures.

D. DIS

Distributed Inter-active Simulation (DIS) systems rely on the interconnection of large numbers of real time vehicle simulators. DIS requires tremendous band width and communication resources. Much research has focused on overcoming the problem of the bandwidth and communication. Bassiouni, Chiu, Loper, Gransey, and Williams have developed techniques in contemporary DIS systems. One of these techniques is filtering; a promising technique to improve the scalability of distributed simulation. This idea of filtering is to analyze the semantic contents of the state update messages of a simulated entity and transmit only the one that is found to be relevant to the other entities.

E. HLA

The most recent development of distributed simulation is High Level Architecture HLA, which has been defined as the "architecture for reuse and interpretation of simulation" (5). Mary Shaw and David Garlan, define the software architecture as the size and complexity of software systems increase, the design and specification of overall system structure become more significant issues than the choice of algorithms and data structures of computation. Structural issues include the organization of a system as a composition of components; global control structure; the protocols for communication, synchronization, and data access; the assignment of functionality to design elements; the composition of design elements; physical distribution; scaling and performance; dimensions of evolution; and selection among design alternatives. HLA is based on the premise that no single simulation can satisfy the requirements of all uses and users. It was developed to provide " structure that will support reuse of capabilities available in different simulations, ultimately reducing the cost and time required to create a synthetic environment for a new purpose and providing developers the option of distributed collaborative development of complex simulation applications" (5). HLA is already a standard for use in the U.S. Department of Defense, has been nominated for standardization in NATO, and is in discussion by the Object Management Group (OMG).

IMS MISSION, which is one of the projects sponsored by NIST, has conducted a development of Distributed Manufacturing Simulation (DMS) that is based on HLA. IMS MISSION project goal is to integrate and utilize new, knowledge-aware technologies of distributed persistent data management, as well as conventional methods and tools, in various enterprise domains, to meet the needs of globally distributed enterprise modeling and simulation [48].

During the development stage of the architecture of DMS, researchers addressed the following problems and issues:

1. The interoperability between engineering and simulation software used to model and predict the behavior of manufacturing system is currently extremely limited.
2. The cost of transferring data between simulation and other software applications is very high. Users must either re-enter data when they use different software applications or pay high costs to system integrators for custom solutions. In some cases, closed systems may totally eliminate the possibility of integration.

3. Each industrial user must rebuild its own copies of models of its manufacturing systems and resources. This is true even if the models are representations of generic or commercial off-the-shelf manufacturing equipment. If the industrial user has several different simulation packages, the models must be reconstructed for each package.
4. Neutral interface specifications that would permit quick and easy integration of commercial off-the-shelf software into integrated environment do not exist.

DMS execution is based on the Run Time Infrastructure (RTI), which is based on the HLA. There are three HLA modules that are required for the integration of distributed simulation:

1. Federation manager;
2. RTI Fedex module; and
3. RTI exec module.

The relationship between HLA modules and the various elements of distributed manufacturing simulation execution environment is illustrated in Figure 5. According to the IMS MISSION, an HLA-based simulation is called a federation, consisting of each simulator, visualization system, real production system, or output analysis system that is integrated by HLA RTI. The term ambassador that is used in Figure 5 is used to describe an interface, that is a collection of

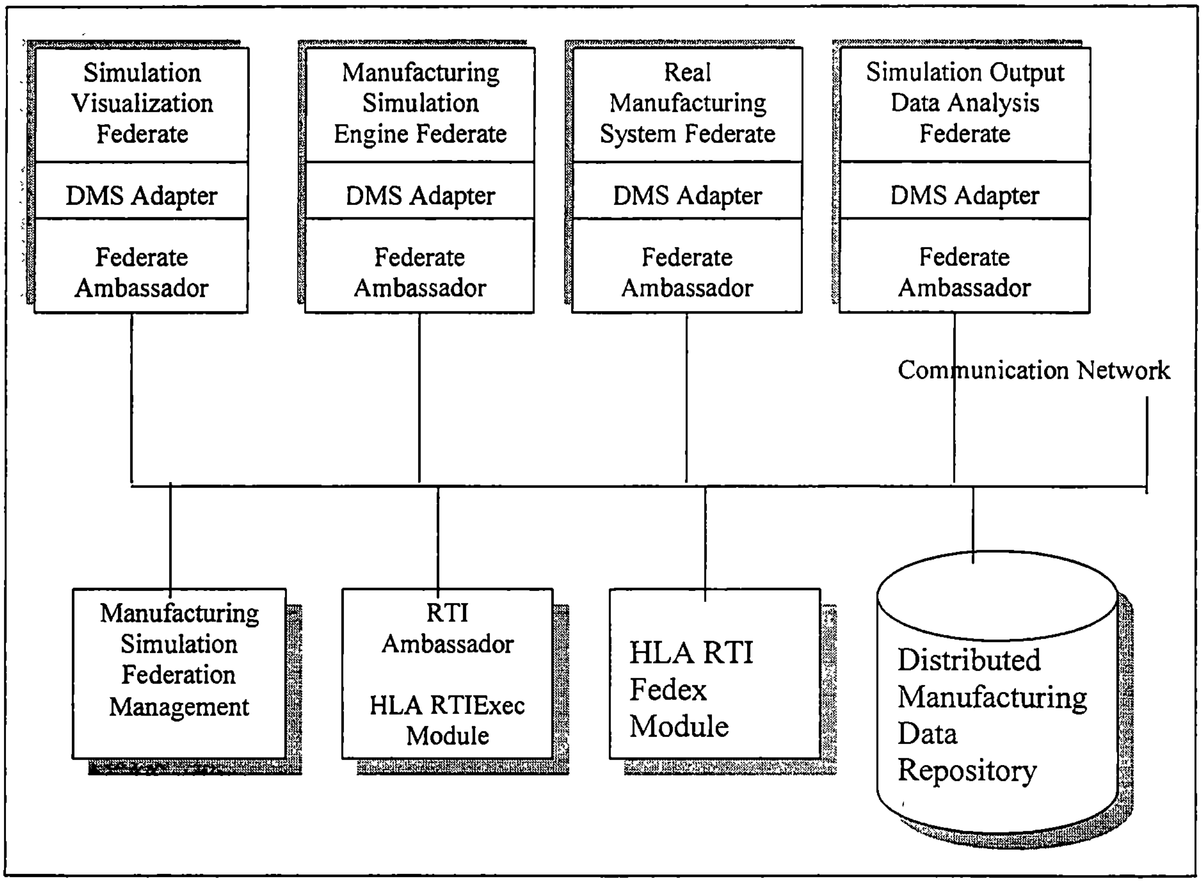

Figure 5: Relationship between Distributed Manufacturing Simulation environment elements integrated by the HLA Run-Time Infrastructure [48].

methods that allow a simulation federate to talk to the RTI and vice versa. The DMS adapter can be thought of as a specialized implementation of the Federate Ambassador that has been customized for use by the manufacturing simulations. Also, it hides the complex interface to the RTI ambassador and exposes a simplified interface for manufacturing simulations to interact. Meanwhile, Distributed Manufacturing Data Repository may include the following types of data stores and management systems:

- Computer files systems
- Web pages and files
- Object-oriented data base management systems
- Relation data base management systems
- Special purpose library management systems
- Software source code control systems

F. Web-Based Simulation

Fishwick offered his perspective on the issue of Web-based simulation and identified the following three impacted areas by the World Wide Web:

1. Education and training;
2. Publication; and
3. Simulation programs.

As an extension to Fishwick's categories, Ernest Page's review of current literature suggests five areas of foci:

- Simulation as hypermedia. Text, images, audio, video simulation- the nature of WWW design enables the production, storage and retrieval of documents containing any or all of these and other kinds of elements. The availability of simulation as a desktop, browser-based commodity has the potential to significantly alter current teaching and training methodologies, both for simulation as a technique, and for disciplines that apply simulation, like engineering, physics, and biology. Paradigms that focus on distance learning and interactive, simulation based education and training are emerging.
- Simulation research methodology. The ability to rapidly disseminate models, results and publications on the web permits new approaches to the conduct of simulation research and scientific research in general. The practical, economic and legal issues associated with the electronic publication of documentation, for example, are numerous. The electronic publication of simulation models raises additional considerations.
- Web-based access to simulation programs. Most commonly associated with the term Web-based simulation, this area includes both the remote execution of existing simulations from a web browser through HTML forms and CGI scripts, and the development of mobile code (e.g. applets) simulation that run on the client side.

- Distributed modeling and simulation. This includes activities that deal with the use of the WWW ad web-oriented technologies (e.g. CORBA, Java RMI) as infrastructure to support distributed simulation execution. Internet gaming issues are included here.
- Simulation of the WWW. Modeling and analysis of the WWW for performance characterization and optimization.

As an example presented by Page, SIMJAVA is a discrete event simulation package authored by Ross McNab and Fred Howell that is written in Java and conceptually based on Sim++ library for C++. A companion package, Simanim, allows the construction of animated SIMJAVA applets. Naturally supportive of the process interaction conceptual framework, a SIMJAVA simulation typically consists of a collection of objects (from the Sim_entity class) each of which runs in its own thread within Java Virtual Machine. Objects are connected via ports (from the Sim_port class) and interact by sending and receiving events (from the sim_event class) along these ports. A static class controls the objective threads, coordinates the advance of simulation time, and maintains the event queues. Page presented in Figure 6 a simple illustrative example of SIMJAVA architecture. The master program, running on a SUN SPARCstation Sim1, contains the single method, main (), which accepts a commandline argument representing the number of distributed clients (in this case two), creates an instance of Sim_system, and binds it in the Remote Method Invocation (RMI)

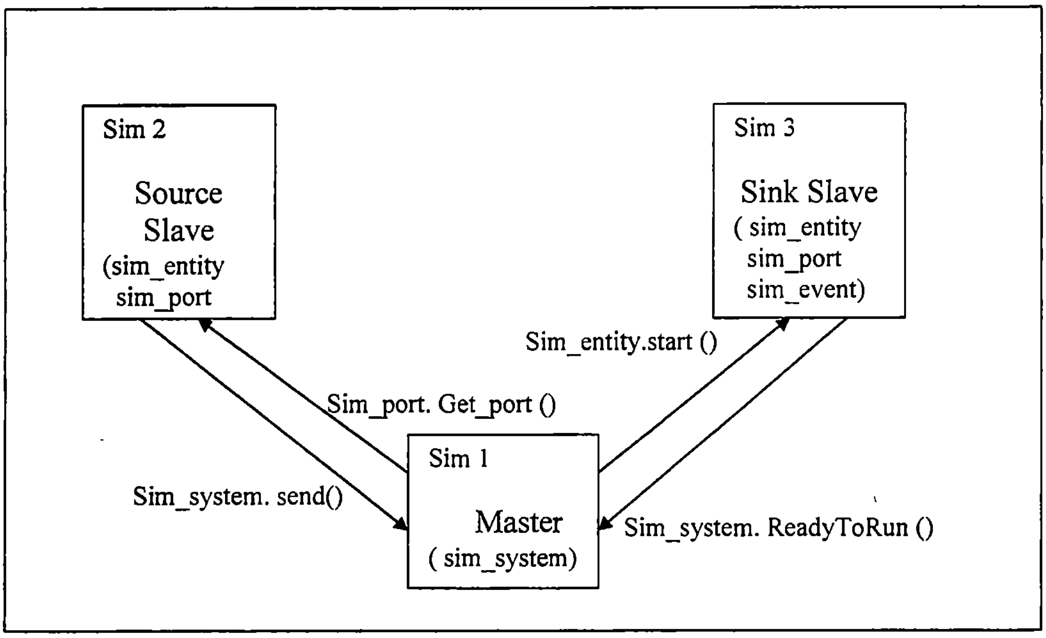

Figure 6: Simple SIMJAVA Architecture [36].

registry. RMI is the object-oriented analogue of the traditional Remote Procedure Call (RPC) for distributed computation. Following the RMI conventions, a single object is used to bootstrap the system. Remote references for the remaining objects are acquired through passed parameters and return values. The source and sink programs, running on Sun SPARCstation Sim2 and Sim3 respectively, each contain their constructor which is extended to accept an instance of the Sim_system (through the interface types). Port creation and addition are wrapped in a try catch block for the java.rmi.RemoteException. Finally, the add () and readyToRun () methods are invoked on Sim_system. Once the number of calls to readyToRun () equals the number of expected clients, a user supplied method that links the entity port is invoked by Sim_system.

The article entitled " Web-based Simulation in SIMJAVA using Remote Method Invocation" stated that the marriage of distributed simulation and Web-based simulation seems a natural one, and initiatives are underway to incorporate web-based simulation code delivery into the next generation DoD standard, the HLA. The conceptual framework support available in modern simulation programming languages and simulation support environment is absent in the HAL Runtime Infrastructure. An implementation of the RTI in Java will provide code mobility and web-based invocation, but the integration of conceptual framework support requires additional steps to be taken.

G. MCT

Peter Sapaty presented in his paper entitled "Mobile Intelligence in Distributed Simulation" strategic issues of using mobile intelligence for formalization, implementation, and management of large Distributed Interactive Simulation in open computer network. Mobile Cooperative Techniques (MCT) are based on mobility of interpreted programs, or program flow, in computer networks rather than traditional data flow. They can do complex jobs in distributed environments by smart agents migrating in space and bringing necessary operations and control to distributed data and other programs. By optimizing the distribution and interaction of data operations in space, MCTs may reduce traffic in computer networks and provide high flexibility in organizing complex processes in changeable environments. Traditional functions of runtime infrastructure (RTI) like federation, deceleration, object, ownership, and time management may performed in a completely distributed manager by using MCT, without central physical resources causing bottlenecks and vulnerability. These functions may be efficiently performed by a network of the same computers that run applications (Simulation), where the management operations may be initiated from any computer and by different users. Mobile intelligence may help create and implement border RTI concepts for heterogeneous distributed systems with a richer set of function. These advanced RTIs may provide dynamic conversions of data structures between dissimilar simulations, self-evolution within open scenarios with varying numbers of participants, and recovery from damages. High level RTI management functions like distributed pattern recognition, distributed simulation assessment needs in making complex decisions may also be provided by MCT.

MCT may provide radical advantages to distributed integrated simulation. The modeled worlds may be arbitrarily and seamlessly portioned and distributed between many computers without a need to replicate the same terrain in different machines. Terrain may be dynamic; its change may spread across machine boundaries (for example, landslides, craters, flooding also smog, fire, etc.). Models of entities (planes, tanks) may freely migrate through different parts of terrain and between computers. All dynamic distribution of terrain and mobility of models as well as visibility of neighboring parts of space in other computers can be done by migrating cooperative agents. Different users may observe any part of the distributed space by using the roaming cockpit techniques based on mobile intelligence. Such organization of distributed simulation may be scalable without limit to any number of machines.

According to Sapaty, mobile intelligence may improve management of complex communication networks supporting large DIS exercises especially those integrating live entities and real command and control systems. For this reason, using high-speed digital cellular networks for wireless communication in DIS is of growing interest.

Different possibilities of applying mobile cooperative technologies for large distributed interactive simulations have been considered. Mobile intelligence with interpreted self-spending and self-evolving programs not connected to particular physical resources may provide high functionality, flexibility, and robustness in organization and management of distributed systems. Most of the presented ideas have been programmed and tested using the distributed WAVE system publicity available via Internet. These ideas may also be implemented within other mobile agent technologies; however, more efforts and much longer codes will be needed as WAVE has special support for such classes of problems. More work has yet to be done on a future development of mobile simulation and control algorithm and performance measurement. It is clear, however, that the optimal solutions suitable for DIS may be found in efficient integration of traditional computations with MCT.

Integration of both distributed geographically simulation and fast-growing communication tools worldwide (Internet), will result in a web-based distributed manufacturing model development simulation. This new idea will be utilized in this research to achieve the goal of simulation modeling lead-time reduction and pervasive implementation of simulation modeling.

CHAPTER III

FRAMEWORK

The focus of this chapter is to develop a conceptual framework for reducing simulation-modeling lead-time that is similar to the framework developed by Monden for reducing the manufacturing lead-time. The methodology to develop the framework will decompose each simulation-modeling step into its components. Each component will be discussed in terms of the nonvalue-added activities. Finally, mechanisms and methods to eliminate the nonvalue-added activities are identified.

Components of simulation modeling lead-time, data collection, model development, verification & validation, experimentation, and implementation steps represent an estimated 92 percent of the simulation modeling lead time; the other 8 percent is spent in problem definition, establishing boundaries, establishing variables, and documentation (see Figure 3). Figure 7 represents the proposed framework for reducing simulation-modeling lead-time. The figure is broken up into 6 main time reductions: problem definition time reduction, data collection time reduction, model development time reduction, verification & validation time reduction, experimentation time reduction, and implementation time reduction. The other three steps (boundaries, variables establishment, and documentation) are implicitly included in the step of problem definition time reduction. The figure presents techniques and mechanism to perform the simulation modeling more quickly. Each simulation-modeling step will be discussed in the following sections.

A. Problem Definition

Problem definition is a very crucial step in the development of simulation models because it defines the depth of details required for the rest of processes of modeling. Many simulation models fail because of misinterpretation of the purposes of the models; misinterpretation causes tremendous time to be spent on modeling useless details. To avoid the waste of time, problem or objectives of simulation modeling should always be identified at the beginning of the modeling through two steps. First, boundaries of the models should be declared. For example, time line, the use or need for simulation model, and budget line. Second, based on the identified constraints and simulation objectives, the expected output variables should be identified for example, WIP, utilization, and others. Output is determined based on the available input and the required outputs. Defining the problems with the simulation team should lead to clear understanding of the project objectives. This team should be involved in a process called concurrent engineering in which design engineers, manufacturing specialists, marketers, buyers, and quality specialists work jointly to define the appropriate simulation objectives. If the buyers and executives from the corporate office are located at different geographical locations, a large delay in the process of problem definition could occur. Such problems could be avoided

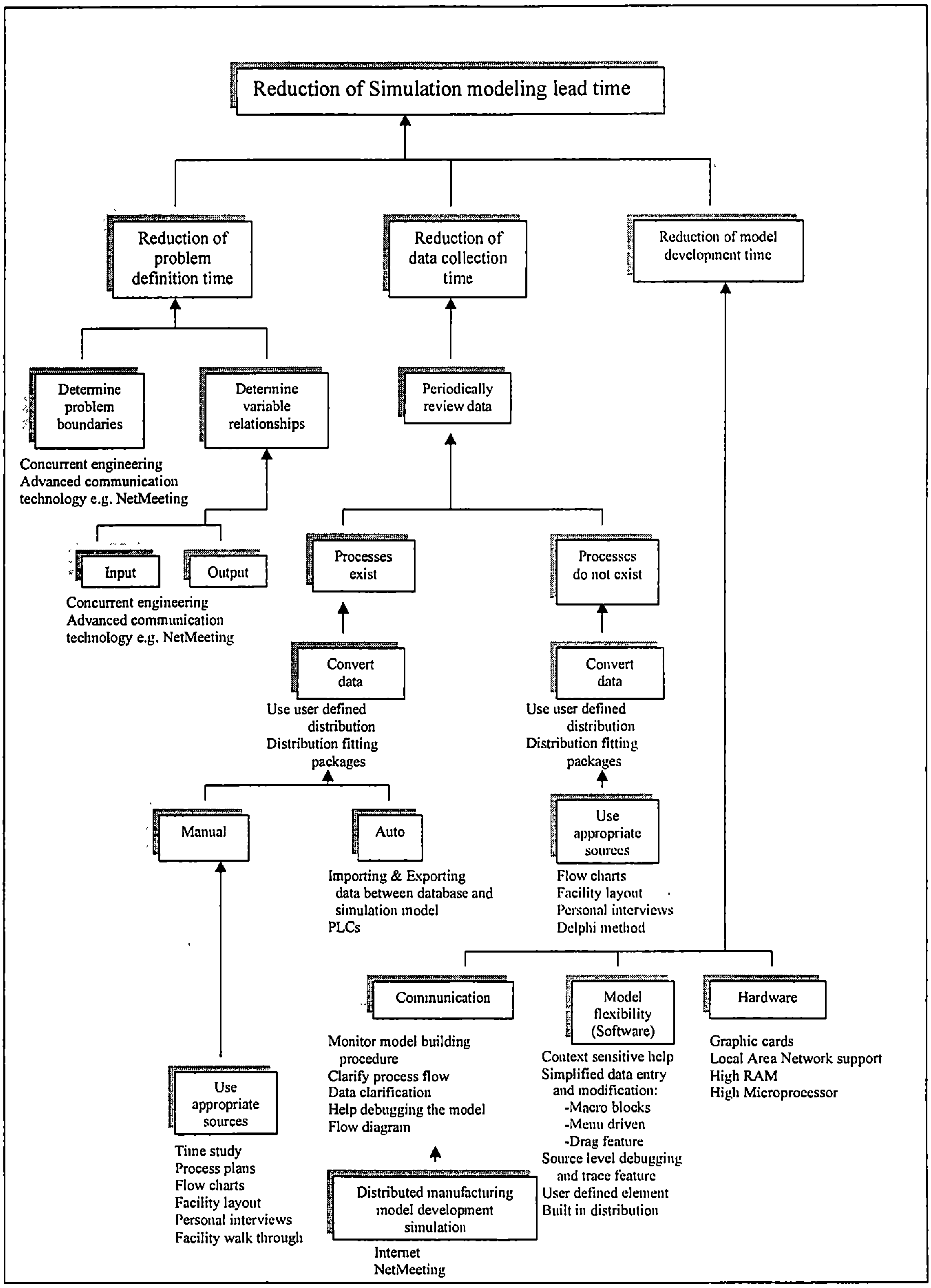

Figure 7: Framework for reducing simulation-modeling lead-time (Continued on next page).

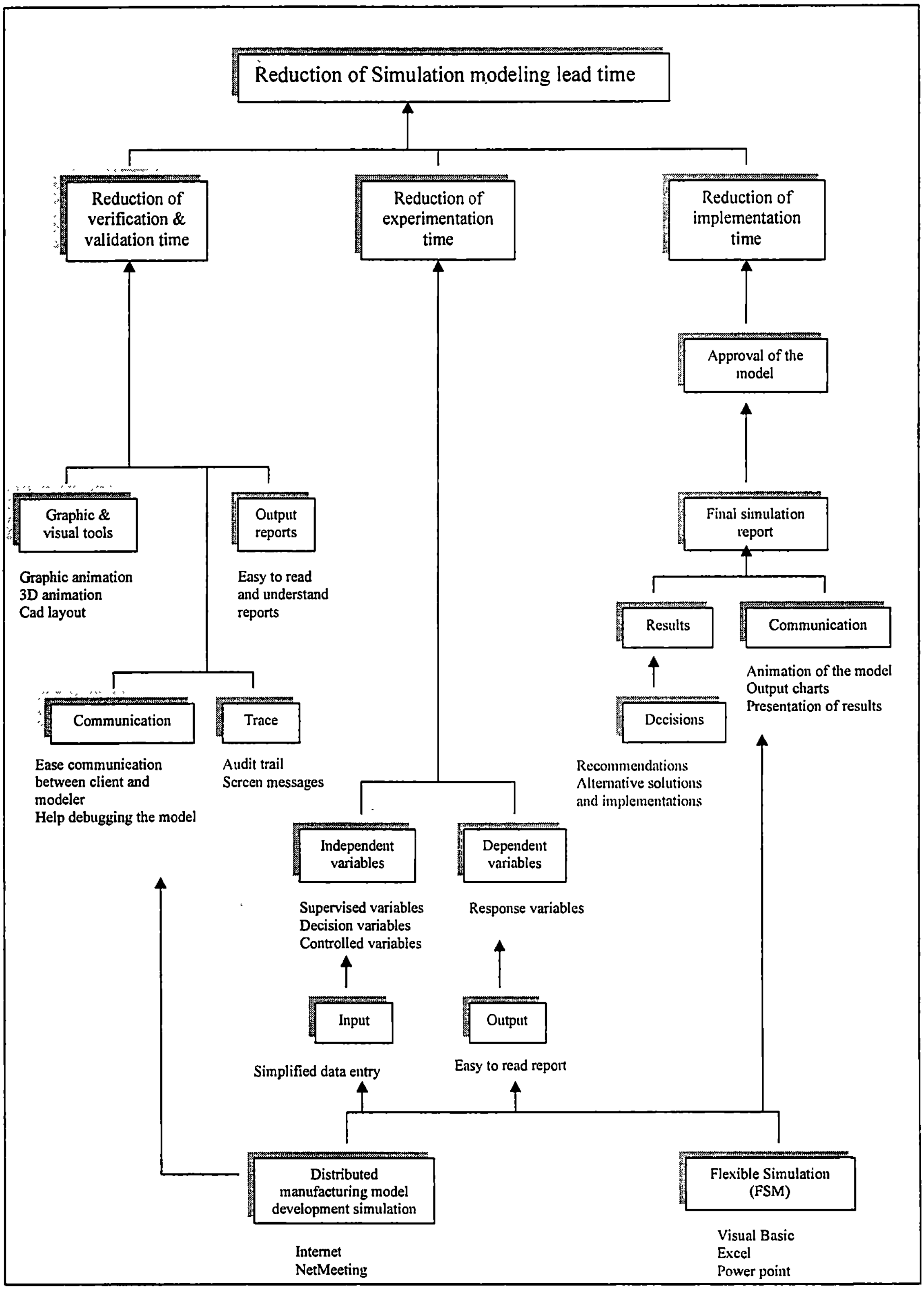

Figure 7 (Continued): Framework for reducing simulation-modeling lead-time.

by applying advanced communication technology such as conference callings or video conferencing (NetMeeting). Consequently, time would be only spent on value added processes (discussion time), and a large portion of the nonvalue-added (communication waste) activities would be eliminated.

B. Data Collection

Data collection is an actual step in the modeling processes. Results of modeling are based on how accurate the input data is (Garbage in Garbage out). Data or information collection represents an estimated 25 percent of the total simulation-modeling process lead-time (see Figure 3). This percentage is composed of the following components (see Figure 8):

- Time spent gathering data from appropriate sources
- Time spent converting data
- Time spent reviewing data periodically

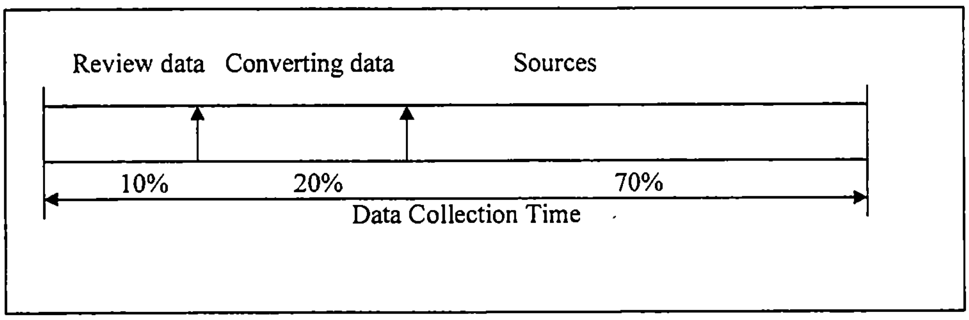

Figure 8: Estimated time percentage of Data collection

Typically, an estimated 10 percent of the data collection time is spent reviewing what data is available and updating it periodically. To avoid outdated data, appropriate sources of data should always be used. Data collection is divided into two scenarios. The first scenario is when processes exist. This is divided into manual data collection and automatic data collection. The second scenario is when processes do not exist, which makes it harder to gather data. Manual data gathering is a time-consuming step because it involves performing time studies, process plans, flow charts, layouts, personal interviews, and a factory walk through. Meanwhile, auto data gathering saves tremendous time because it eliminates the need for performing time studies. For instance, Programmable Logic Control (PLC) serves as a controlling system. It provides the ability to create sophisticated process control for highly automated systems. PLC is a powerful tool in system integrating, but it only functions as well as it is programmed. Facilities data collection is one of the features of PLC. Most manufacturing companies have databases with routing or scheduling data and many simulation products allow data from these databases to be imported via spreadsheet applications or a subroutine written in C or Pascal that would be helpful to have linked to the model logic [34]. On the other hand, if processes do not exist, gathering data should be accomplished through flow charts, layout, interviews, Delphi method, and historical data. After gathering all the required data, it needs to be converted to distributions. Converting data could take a lot of time, but wasted time could be avoided by utilizing database fitting packages such as Jump in, SAS, Input Analyzer in some Simulation software, ASI software. These softwares provide applied statistical analysis, which is going to be used as the input to the simulation model.

C. Model Development

Model development is the step in which a modeler should always design the model to answer the relevant questions and imitate the actual system. The model should not be so detailed that it becomes confusing. At the same time, it should not oversimplify the system to the point that it does not solve any problem. Moreover, model development is the step where experts of the modeling language transfer process flow to code and blocks in anticipation of complete flexibility. This process is also time-consuming; it represents an estimated 20 percent of the total time spent on simulation modeling lead-time (see Figure 3). If we break down the time that is spent in this process, it would be related to the following components:

- *Communication time:* Time is spent monitoring the model building procedures, clarifying process flow, clarifying meaning of some data, and helping in debugging the model.
- *Time spent on complicated software*: The rest of the time is spent transferring the processes to codes so that the computer can understand them. The more flexible the software to adopt the following elements, the more time saving it is:

- Help menus describes how to use the software and serve as a reference that guides a modeler through various operations and supply examples of the modeling.
- Simplified data entry and modification requires an excessive amount of tedious effort to complete. Input effort could be minimized through using the following features that allow each segment of logic or series of simulator action to be defined only once:
    - Macro blocks -- common blocks
    - Menu driven
    - Drag feature
- Source level debugging and trace features generate a detailed history of many things going on in the simulation, which is used primarily for debugging the model.
- User defined elements have the capability to define the needed variables and logic elements for representing special situation that should be provided by the software.
- Built-in-distribution software should be provided with statistical distribution to achieve the quickness of modeling.

- *Time spent on Hardware problems:* Hardware plays a big role in speeding up the process of running the model. The better hardware requirements, the faster model execution. To assure reduction of time in the process of model execution, Hardware should be supplied by the following:

- Graphic cards: to better represent the model via animation, an appropriate graphic card should be installed
- High RAM: the higher the RAM, the faster the execution of simulation models.
- High Microprocessor: it is advisable to run the simulation models on a high performance PC because large complex models can sometimes run very slowly on a low performance PC.

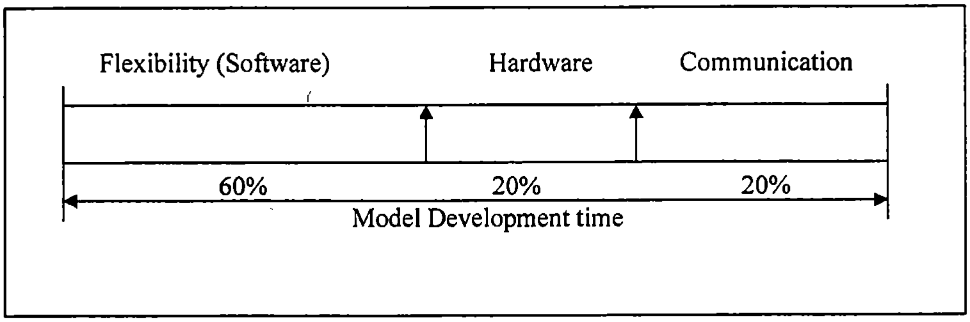

Figure 9: Estimated time percentage of model development

Figure 9 illustrates that twenty percent of the model development time is spent in communication processes. This percentage of communication could be reduced significantly through better interaction between modelers and clients. Advanced communication would speed up the process of clearing the questions between both sides. At the same time, it would build a strong base of modeler's knowledge for the process through having immediate answers to the questions. Also, sharing the modeling process with the end user or another modeler will significantly help debug the model and create control over the model. Therefore, having implemented the concept of distributed manufacturing model development simulation via the Internet, a significant reduction of the time spent on communication in this step is anticipated.

D. Verification and Validation

It is necessary to understand the meaning of verification and validation. Verification is the process that confirms whether the computer program performs as expected and intended in terms of the logical representation of the model. Meanwhile, validation is a confirmation of whether the computer logic mimics the actual system. Validation is concerned with three basic questions [46]:

- Does the model adequately represent the real world system?
- Are the models generated behavioral data characteristics of the real system's behavioral data?
- Does the simulation model user have confidence in the model results?

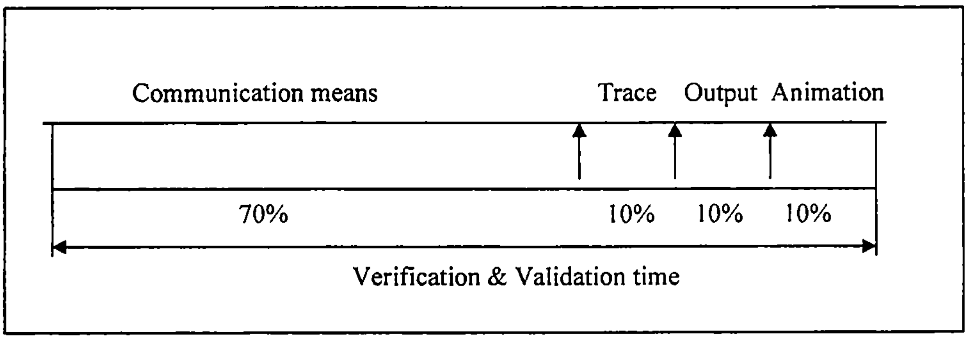

Figure 10: Estimated time percentage of verification & validation

Figure 10 illustrates that verification and validation are mainly communication processes requiring the model builder to convey the basis for confidence in a model to target audiences through periodic review to ensure validity (e.g. frequent traveling), involving the ultimate user in the entire simulation processes makes validation easier. One conclusion of the previous definition of verification and validation is that communication plays a big role in the process of verification and validation. On the other hand, the other estimated 30 percent of the time is spent in the value-added process of verification and validation, which is composed of the actual discussion time for verification and validation. These value-added steps could also be reduced through implementing the following elements:

- *Trace entity movements via the trace feature:* trace is one way to verify the model accuracy. It helps digging deeper in entity movement or debugs any problem during the verification processes.
- *Graphic and visual tools*: animation shows what the model logic is doing, and creates efficient verification and validation through presenting graphic animation, 3D animation, and CAD layout.
- *Output reports:* verification and validation time could be reduced if understandable, easy-to-read reports existed. These would make it easier to make quick decisions as to whether the models are verified and validated or not.

Meanwhile, reduction of communication could be easily accomplished by finding faster communication methods. A method should allow the modeler and the end user to communicate the previous three elements from geographically distributed sites; communication method time could go down tremendously via distributed manufacturing development simulation. For instance, implementing NetMeeting would not require the modeler to frequently travel to meet with personnel to verify and validate the model. He/she could communicate the previous discussed elements remotely.

E. Experimentation and Analysis

The experimentation and analysis step involves the actual running of the experiments and the analysis of the results. Complicated statistical issues must be addressed at this step so that the computer model can correctly generate model behavioral data through planned experimentation and data analysis. Figure 11 shows that an estimated 60 percent of the experimentation time is spent inputting independent variable data (supervised variables, decision variables, and controlled variables) to perform "what if" scenario tests; it takes much more time if the user is not experienced with the model. Also, it takes an estimated 35 percent of the experimenting and analyzing time to put the output dependent variable data (response variables) in terms of understandable data to non-molders. The value-added processes of this step take the least amount of

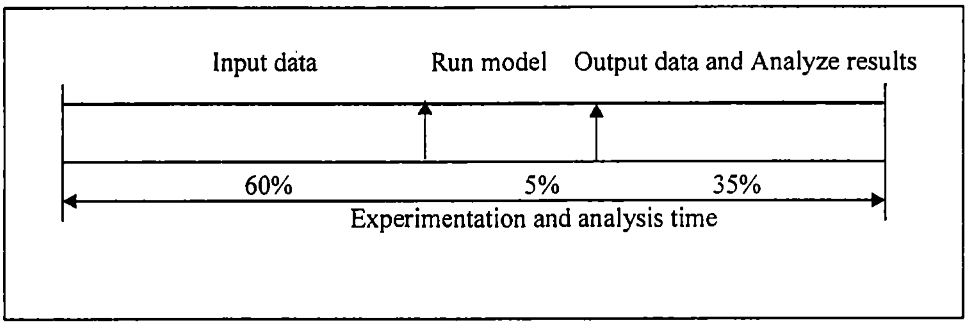

Figure 11: Estimated time percentage of experimentation and analysis

time; only 5 percent of the step is spent running the model. Therefore, the greatest impact of the reduction should be focused on the nonvalue-added activities of inputting the data and analyzing results. Experimentation is a value-added process in itself, but the traditional way of utilizing it is a nonvalue-added process. One of the major mechanism in reducing simulation modeling lead-time is Flexible Simulation Modeling (FSM). FSM is a new concept within the realm of simulation that allows the user the flexibility to design a simulation model via simple methods to input values for variables defined in the simulation model. There are three primary advantages of FSM. First, the life cycles of the models significantly increase, since the same simulation model can be utilized for a variety of scenarios rather than developing a separate model for each scenario. Thus, the longer life of a simulation model results in cost reduction over the life of the model. Second, decision makers have the output available to them much quicker than in static model, since they are not dependent on a programmer to make the required changes. Third, FSM is more conductive to error detection, since syntax and logic errors are eliminated. Figure 12 represents a sample structure for FSM. It illustrates the coupling of simulation software with other software. Visual Basic based menus are coupled to the front end of a simulation model to allow the user to either redesign the processes defined in the simulation model or to change the production characteristics of the model. Excel-based reports are generated at the back of the simulation model.

Based on actual projects conducted by the University of Tennessee, flexible simulation modeling is anticipated to reduce the inputting and outputting data time (Reprogramming time) by at least 50 percent [43]. More productive interaction between the modeler and the end user could make the step of experimentation more time efficient. Simulation-modeling time could also be reduced by adding remote monitoring by personnel via the easy communication of the Internet (Distributed manufacturing model development simulation) to perform the experimentation.

F. Implementation

Implementation is the step in which all stages of the simulation modeling have been completed and the results are understood, accepted, and used. This last step is to make recommendations for inputting in the actual system based on the results of simulation [34]. To reduce the experimentation time, recommendations and alternative solutions should be communicated to executives through the animation of the model and charts that explain output results so the executives

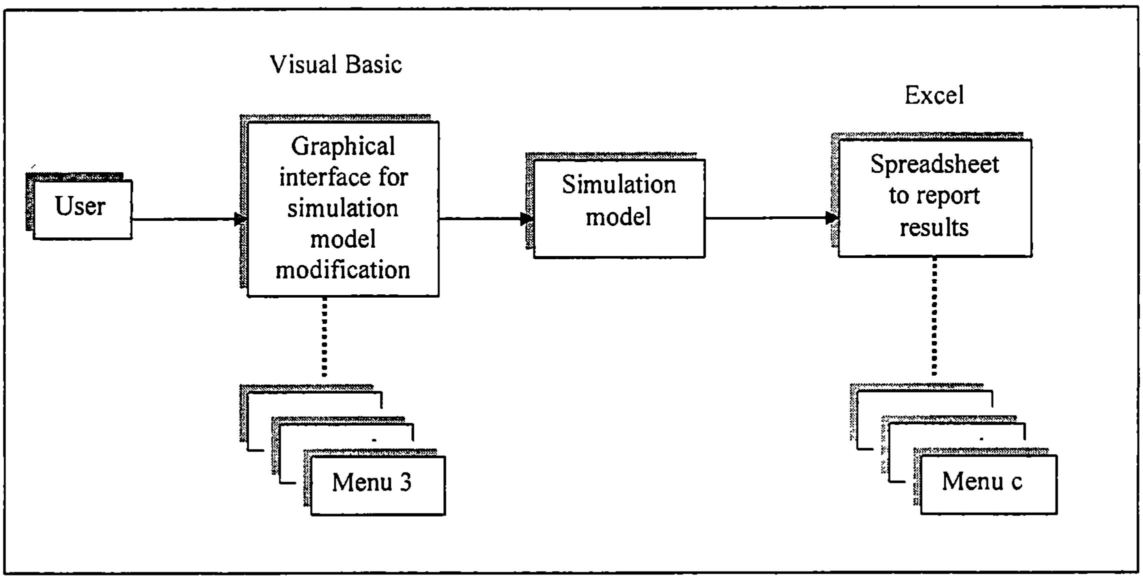

Figure 12: Flexible Simulation Model structure [43].

can make final decisions on the model. In fact, flexible simulation is also a key to implement simulation pervasively throughout the organization. It tremendously reduces the time of implementation in terms of the model being user friendly to inexperienced users who have less programming knowledge. FSM requires less than 50 percent of the time that the static model required. Table 2 presents a comparison of programming time for a study that was conducted on Printed Wire Board (PWB) by the University of Tennessee.

Table 2: Reprogramming Times for Static modeling Vs FSM [43].

Processes Name	Time to change models (min.)		
	Static model	FSM	(%) Difference
Non- Conveyorized Tin Palladium	20.4	8.25	59.6
Non- Conveyorized Organic Palladium	22.5	9.33	58.5
Conveyorized Tin Palladium	68.5	7.5	89.1
Conveyorized Organic Palladium	72.33	8.25	88.6

Since the end-user is no longer dependent on the programmer to make changes on the model, he/she can modify the processes more quickly during the implementation.

In addition, utilizing the distributed model development simulation could significantly reduce the communication traffic by remotely sharing and discussing the graphs and spreadsheets report.

CHAPTER IV

METHODOLOGY

Most of the wasted time in developing a traditional simulation model is due to the communication between clients and modelers; consequently, we have been persuaded to seek better and faster communication tools to create the collaboration. We have found software that integrates the concept of geographical distributed simulation and fast communication; this software is called NetMeeting. Microsoft introduced Windows NetMeeting to provide powerful conferencing and collaborative functions in a complete integrated package for Internet or corporate Intranet for information can be shared with two or more meeting participants in real-time, from one or more applications on a user's computer (e.g. simulation modeling). Participants can exchange graphics or draw diagrams with the electronic whiteboard, send messages, or record meeting notes and action items with the text-based chat program, and send files to other meeting participants using binary file transfer capability (e.g. results of simulation model). With a video capture card and video camera, video images can be sent and received over the Internet or corporate Intranet for face-to-face communication during a meeting. Video can be received even if the user does not have a camera connected to his/her computer.

Windows NetMeeting is a collaborative tool that lets the user:

- Do "Internet Telephony" with one other person

- "Videoconferencing" with one other person.
- Draw or work with images on a shared whiteboard with eight other people.
- "Text chat" with eight other people.
- Send files up to eight other people at one time (e.g. spreadsheets).
- Share an application with eight other people (e.g. simulation).

This software is limited to the number of people that can share the information. It is limited to:

1. Eight people, but if a user "branches off" from different computers, more are possible;
2. Bandwidth; and
3. Processor speed.

NetMeeting's features can be efficiently utilized as a communication tool during the processes of simulation model development, verification & validation, and the experimentation & implementation.

A. Audio and Video

Users can utilize simulation modeling even better by using audio and video enhancements to see other people, share ideas, and converse with audio and video enhancements, the modeler can

- Send and receive real-time video images using Windows-compatible equipment.

- Send video and audio to a user who doesn't have video hardware.
- Use a video camera to instantly view items, such as hardware devices, that are displayed in front of the lens.
- Ensure that people hear each other by adjusting the automatic microphone sensitivity level setting.
- Change the size of the video window sent to another user during a video conferencing session.
- Remotely make the trade-off between faster video performance and better image quality.

B. Whiteboard

During a simulation model development, the whiteboard could be utilized to perform the following:

- Review, create, and update graphic information (e.g. process flow diagram, layout).
- Manipulate contents by clicking, dragging, and dropping information on the whiteboard with the mouse.
- Cut, copy, and paste information from any Windows-based application onto the Whiteboard.
- Use different-colored pointers to easily differentiate participants' comments on the model.

- Save the Whiteboard contents for future reference.
- Load saved Whiteboard pages, enabling the participation of information before a conference, then dragging and dropping it onto the Whiteboard during a discussion of a simulation model. Figure 13 gives an illustration of a whiteboard:

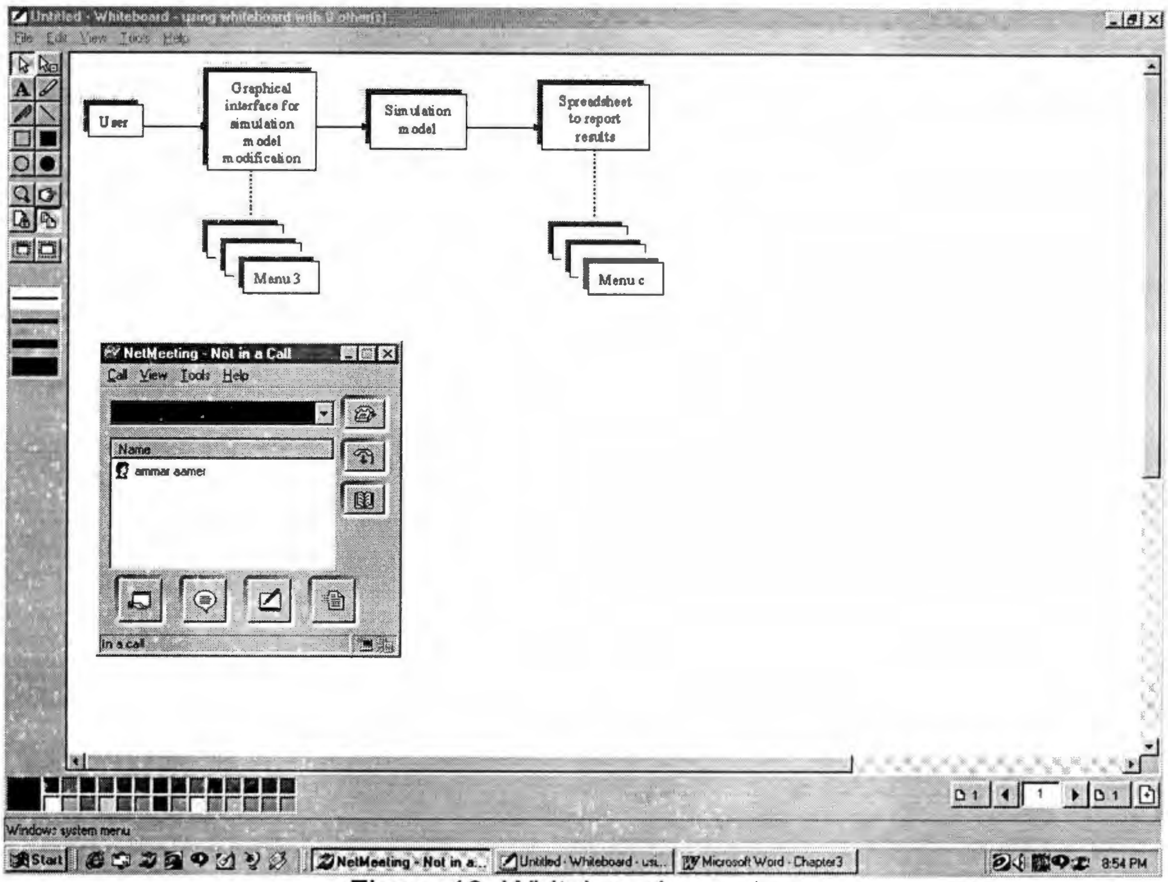

Figure 13: Whiteboard sample

C. Text Chat

This feature enables modelers to:

- Type text messages to communicate with other people during a simulation discussion.
- "Chat" with one person or a group of people across multiple computers.
- Use "Whisper" mode to send private messages with another person during a group Chat session.
- Save the contents from the Chat session to a file for future reference.

For example, a modeler can receive instantaneous comments on, or on modifying the model while performing the model development step.

D. Files Transfer

If necessary, the modeler could send a copy of the simulation model worked on to allow others (i.e. supervisors) to modify it. Using this feature, the simulation modeler could:

- Send a file in the background to conference participants. It could be a database or output files (plots, tables, etc)
- Send the file to everyone in the conference, or to one or more selected participants.

- Accept or reject transferred files.

For example, this feature could be used to transfer results of the model instantaneously to participants. Data collected could be transferred via file transfer as well.

E. Program Sharing

The most beneficial feature for simulation modeling in NetMeeting is program sharing. This feature allows the simulation modeler to share programs (e.g. Animation) in a frame, which makes it easy to distinguish between shared and local applications on user's desktop. For instance, a modeler can present an animation to verify or experiment with the model by presenting it remotely through this feature. The same animation would be displayed on each participant's screen, no matter where the participants are as long as they are logged on the Internet and allowed to participate in this collaboration by the modeler. The shared program frame can be minimized to do other work if the participant does not need to work within the current conference program because it is easy to switch between shared programs using the shared program taskbar. The modeler has the ability to allow just one person to work in the shared program at a time, approve conference participants' requests to work in the program that would be introduced, and allow or prevent others from working in a program using the sharing dialog box. Figure 14 presents an animation of simulation modeling that is shared between more than two geographically distributed users.

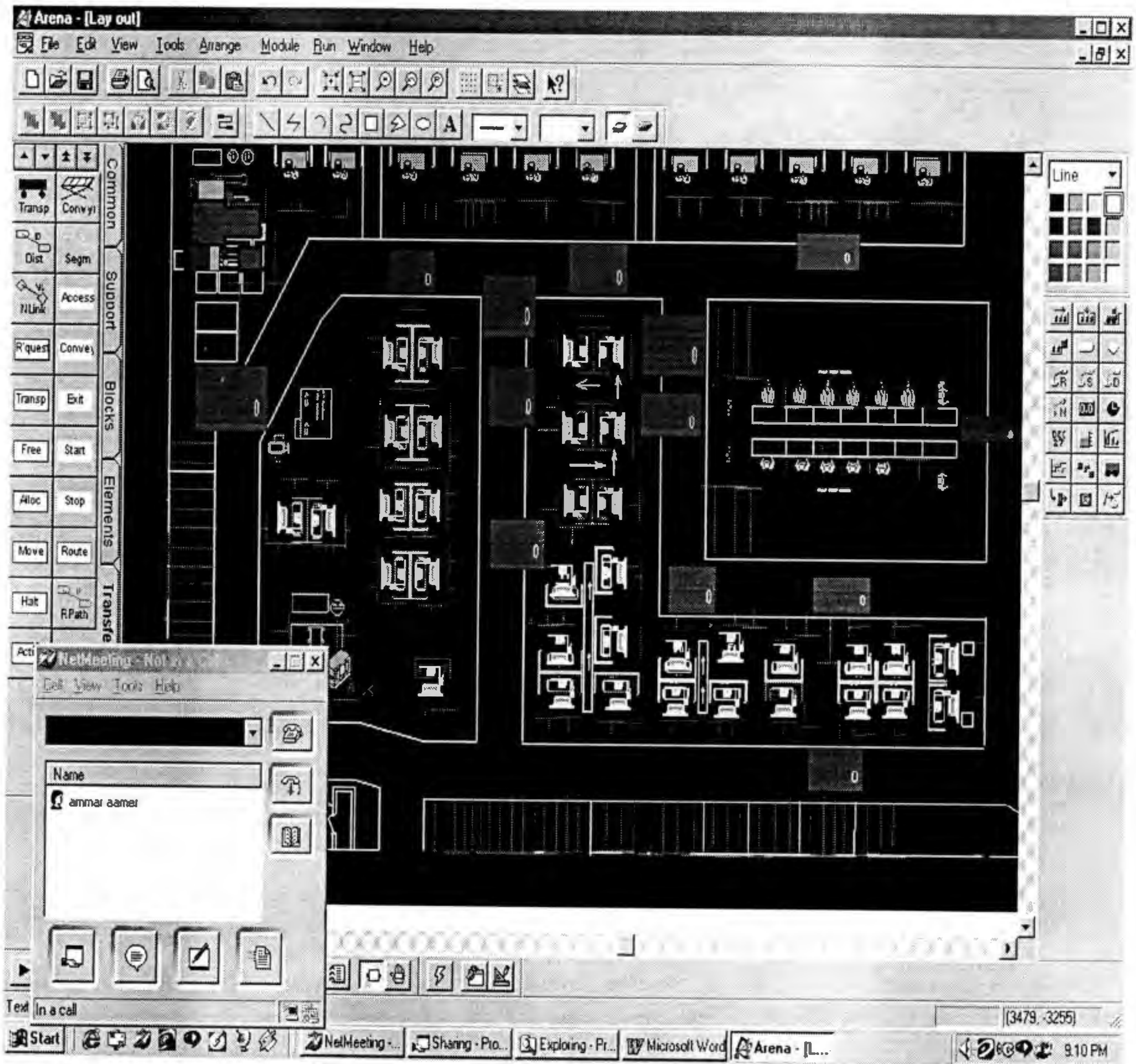

Figure 14: Shared Animation sample

CHAPTER V

A. TRADITIONAL SIMULATION MODELING CASE STUDY

A traditional simulation-modeling project was initiated in December, 1998. This project took an estimated total lead-time of 25 weeks as it is illustrated in Figure 15.

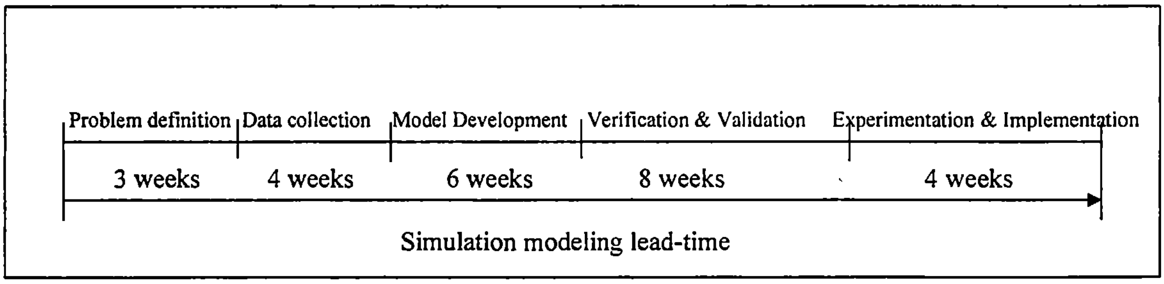

Figure15: Estimation of unexpected total simulation modeling lead-time for the traditional simulation modeling case study.

Figure 15 presents the unexpected lead-time for simulation modeling development. This unexpected lead-time is due to several issues that are discussed in this chapter (As expressed in chapter 1, expected lead-time is the time that is spent for developing a traditional simulation models based on the nine steps of simulation model development.) Based on different simulation projects conducted by the University of Tennessee and our literature search, traditional simulation modeling time (Figure 16) was estimated. Figure 16 presents the overall estimate time of 14.5:

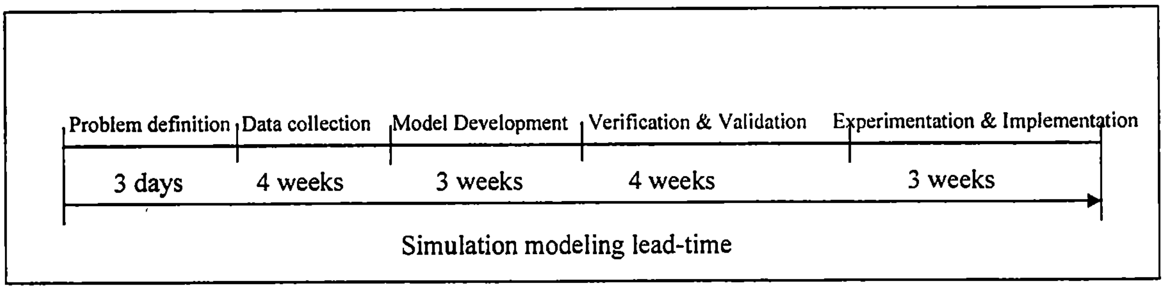

Figure 16: Estimation of expected total simulation modeling lead-time for traditional simulation modeling.

A.1 Problem Definition

Traditionally, the purpose of a simulation model is defined at the very beginning of a model development. Then, based on the output requirements at this step, input requirement is clarified. These requirements should be a result of an extensive discussion with personnel; in other words, applying the concept of concurrent engineering. Unfortunately, the problem definition step was not applied correctly during the performance of the traditional simulation-modeling project. The need of the purpose of simulation model was not defined and variables were not identified clearly at the beginning. Consequently, time was wasted on modeling meaningless variables. There was no simulation team who should have discussed the purpose of the simulation, objectives, and constraints of the model; consequently, this step was a big modeling time factor. Therefore, this step took much longer than what it was expected: It took 3 weeks instead of 3 days.

A.2 Data Collection

Because of the collection of accurate data is very important, this step is estimated to take total time of four weeks. Most of the time is spent on collecting data from appropriate sources. In our case, the production line was not in place yet and processes did not exist; therefore, historical and estimated data were used. Most of the time was spent on reviewing drawings and designs of the production lines through various meetings with a contact person. Process times and routing times were estimated by the company based on the anticipated daily average demand. Tremendous time and effort by both sides were spent in clarifying the understanding of the layout, process flow, bill of material, job descriptions, etc. The data could have been more accurate if people of different expertise had met to discuss the requirements. Even so, the time spent in this step was very close to the expected lead-time of data collection. Our data collection was broken into:

- Communication time: the modeler had to travel to the plant site every time he needed clarification.
- Meeting time: due to busy schedules, meetings did not last more than an hour on the average to discuss problems and issues.

A.3 Model Development

Upon the understanding of processes and completion of gathering necessary data, the model development was performed. Traditionally, model development takes an estimated lead-time of three weeks. In our project, model development was the most time-consuming step, not because of the difficulty in transferring the process flow to codes and blocks, but because of the complexity of the manufacturing processes; many questions were raised about the processes that required the presence of personnel during the modeling. Due to the increasing number of questions that were being asked, the model development process was performed on-site in order to have a quick and easy access to personnel. This increased the simulation modeling lead-time due to the frequent traveling. In addition, difficulties were faced due to hardware problems. The modeling on site was performed on a notebook computer with very low RAM and speed. The model was so complex that it took about an hour just to run the model to check its accuracy. The actual value-added processes (coding) in model development step took a total of almost four weeks, while the nonvalue-added processes (communication and Hardware problems) took a total of almost two weeks.

A.4 Verification and Validation

During the verification and validation step, the model is checked to see whether it represents an actual system. This is usually done through presenting an animation of the model to verify it, and comparing the output of the simulation model with the estimated production line output to validate it. The expected estimated total lead-time for this step is four weeks. Unfortunately, one of the engineers, who was the contact and the validation source, did not have enough time to discuss each step of the modeling that was performed. Periodical reviews to ensure the validity of the model were performed too: these took much time. This forced the modeler to be present whenever the simulation model was verified or validated. The unexpected verification and validation process time that was spent could be broken into the following:

- Total of two weeks of value-added discussion time
- Total of six weeks of nonvalue-added communication time and setting up meetings with the contact.

A.5 Experimentation and Implementation

After verifying and validating the model, the processes of experimentation and implementation take place. Some statistical issues are addressed at this stage to perform a plan of experimentation analysis. Changing data in the codes and blocks to perform tests requires full knowledge of where to input these data. An estimation of an expected total lead-time to the step of experimentation and implementation is three weeks. In our project, the end-user lacked the knowledge of inputting data, which required the presence of the modeler during every attempt to perform the "what if" scenario. Also, due to the hardware problems, it took much time to run the model and it took much more time to change the input data; output data had to be converted to some understandable information for personnel. A total of two weeks was spent on performing experimentation. It took almost a total of the same time whenever results were implemented after performing the experimentation. The previous mentioned issues were obstacles in developing a traditional simulation model. They could have been avoided if the proposed framework had been followed. Furthermore, implementing the concept of distributed simulation could further reduce the expected traditional lead-time of the nine steps of simulation model development as it is illustrated in Figure 7.

B. Distributed Manufacturing Modeling Development Study

This section discusses the feasibility of reducing the expected lead-time via utilizing the concept of distributed simulation. The previous mentioned geographical distributed communication tool (NetMeeting) is applied here. Tasks in simulation model development that were impacted by the concept of distributed simulation are presented and illustrated in terms of lead-time reduction.

B.1 Model Development

It was addressed in the previous case study that the model development step took a total of almost six weeks of modeling time. Two weeks of the lead-time were wasted on communication, software, and hardware problems. The same modeler could have spent much less time developing the simulation model by remote monitoring and remote helping in debugging the model by the simulation team and other modelers. For example, a model could have been shared with the simulation team remotely to debug the problems and ensure the flow of the entities was correct. Comments and suggestions could have been sent instantaneously via the text chat feature. Additionally, the modeler would have utilized the state-of-the-art equipment from a simulation lab, which would have run the model faster and clearer. Therefore, the expected lead-time of the step of model development (three weeks) would have been greatly impacted; it would have been reduced to an estimated total of two weeks (reduction of 33 percent).

B.2 Verification and Validation

The verification and validation step is one of the most impacted tasks of developing a simulation model by utilizing NetMeeting. This step in the previous traditional case study took a total of eight weeks lead-time. Six out of the eight weeks were nonvalue-added process of communication time between clients and modelers; this huge waste could be eliminated completely by applying the right communication tool (NetMeeting). Verification and validation is usually accomplished through two steps:

1. Animation; and
2. Output data (Plots, Histograms, etc).

For instance, a modeler could share the animation shown on the screen with different geographically-distributed participants. Each participant (up to eight) could visualize the modeler's screen remotely. At the same time, comments and suggestions about the validation of the model could be sent directly after running the model through the text chat and the audio features in NetMeeting. Sharing would allow the team to quickly trace the problems if they exist, and modeler would not have to frequently travel to geographically-distributed locations to present outputs or animation every time the model needed to be verified or validated. Therefore, the waste of communications traffic would be eliminated and reduction of the expected lead-time in this step is feasible.

B.3 Experimentation and Implementation

Flexible Simulation Modeling (FSM) is the main key in reducing the simulation modeling lead-time in the steps of experimentation and implementation. Data could be easily inputted or outputted from the simulation modeling either by the modeler or by the end-users through pre-formatted menus. Simplified data entry via Visual Basic forms as menus would be a good example of easy data entry; this would allow the end users to perform "what if" scenario and input the input variable easily. Excel-based reports would allow the end-users easy access to end results of the dependent variables during the experimentation, which would ease the decision making. These results could be communicated geographically and shared through NetMeeting with every individual in the simulation team, which would allow the simulation team to make decisions faster and report them to the management via a final simulation report. At the same time, implementing a geographically-distributed tool would eliminate the necessity of having a modeler present during the performance of the experimentation and implementation. Thus, applying the features of NetMeeting during the processes of experimentation and implementation would decrease the simulation modeling lead-time significantly. A combination of FSM and

NetMeeting would reduce the experimentation and implementation time to an estimated total time of three days modeling time. Figure17 illustrates the total estimated simulation-modeling lead-time based on the distributed simulation study.

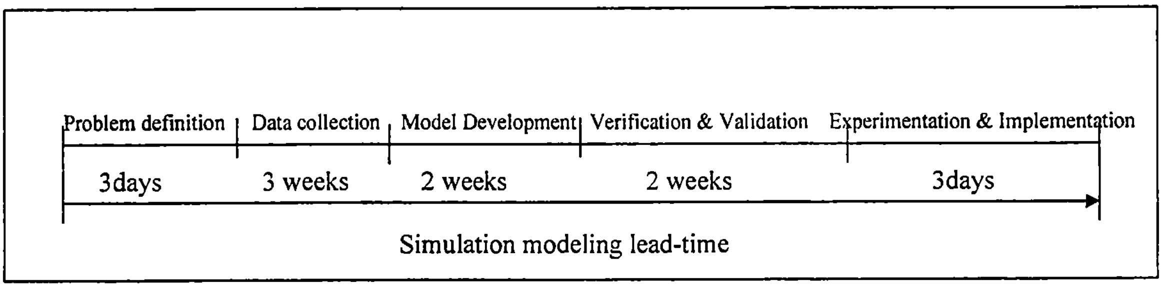

Figure17: Estimated total simulation modeling time based on the distributed simulation study.

If we compare both case studies in terms of expected, unexpected, and reduced lead-time, we find a significant time reduction in the simulation modeling lead-time. This reduction is due to following the steps of the new framework.

Table 3 presents a comparison between the expected and unexpected traditional simulation modeling lead-time and the distributed simulation study. This implies that developing a simulation modeling unsystematically could lead to longer lead-time than the expected time. The expected lead-time is based on following every step of the simulation model development correctly as illustrated in the new framework (see Figure 7) with the exception of distributed manufacturing model development simulation. Thus, applying the concept of distributed simulation through a right communication tool (NetMeeting) could tremendously reduce the expected lead-time by 44.8 percent.

Table3: Comparison between traditional simulation modeling lead-time and distributed simulation study.

	Expected Traditional Simulation Model	Unexpected Traditional simulation model (Case study)	Distributed simulation study
Problem Definition	3 days	3 Weeks	3 days
Data Collection	4 weeks	4 Weeks	3 Weeks
Model Development	3 weeks	6 Weeks	2 Weeks
Verification & Validation	4 weeks	8 Weeks	2 Weeks
Experimentation & Implementation	3 weeks	4 Weeks	3 Days
TOTAL	**14.5**	**25 weeks**	**8 weeks**

CHAPTER VI

CONCLUSION

Many of the constraints that have kept simulation from being used pervasively have been eliminated through advancement in technology and education which has made the software available today more user-friendly by requiring less programming skill than ever before. This advantage has allowed for the results of change to the modeled systems to be visually seen on the screen. This ease of programming has also dramatically reduced the long lead-time required to perform a simulation model. Obtaining and analyzing results from the simulated systems is much quicker due to the increased speed of the computers. Still, in spite of the elimination of the constraints, simulation is not used in many companies to improve decision-making. The modeling of complex systems is still a formidable task, requiring the functional communication and some knowledge of statistical analysis.

Functional communication and statistical analysis are the main concerns in the simulation modeling development in terms of long lead-time. Therefore, a framework that helps in reducing the previous two main concerns of statistical analysis and communications issues in simulation model was proposed in this study. The two key mechanisms to the previous issues are Flexible Simulation Modeling (FSM) and Distributed Manufacturing Modeling Development Simulation (DMMDS). FSM is one mechanism that promotes, enhances, and encourages the use of simulation modeling pervasively. It overcomes the long lead-time of modeling by allowing decision-makers to have the statistical analysis output (e.g. Visual Basic and Excel) available to them quicker and easier. Further, Distributed Manufacturing Model Development Simulation (DMMDS) is the other key to solve the long lead-time that is represented in communication waste. DMMDS application, NetMeeting, was utilized in a hypothetical case study in this paper in combination with FSM. They had a significant impact on the simulation model development lead-time. A hypothetical reduction of 14.5 weeks was concluded from the hypothetical study that was presented in this paper, which represents 58 percent of the total simulation model development lead-time traditionally. This reduction in the lead-time will definitely help increase the use of simulation modeling pervasively throughout organizations, which is the main goal of this paper.

Future studies should be conducted to continuously decrease the simulation lead-time. One of the areas that should be studied is the feasibility of data collection time reduction. More research and studies should concentrate on the feasibility of adopting user-friendly tools and integrating simulation and production lines to be able to reduce the long time of data collection. There is also a high potential in reducing model development time. This reduction could be accomplished through developing new software that is even user-friendlier than what is available today.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1]. Kirby, Ken, Rupy, Sawhney. "Simulation: Shifting the Competitive Edge for the Next Generation." MDC UPDATE Vol. 6, summer 1997.

[2]. Bassiouni, Mostafa, M., Chiu, M., Loper, M., Garnsey, J., Williams. "Performance and Reliability Analysis of Relevance Filtering for Scalable Distributed Interactive Simulation." Annual Review Report for the Advanced Distributed Simulation Research Consortium, pp. 60-61, 1997.

[3]. Bassiouni, Mostafa, M., Chiu. "Semantic-based Bandwidth Reduction in Wide Area Training Networks." Report for the Advanced Distributed Simulation Research Consortium, pp. 65-66, 1997.

[4]. ATV-DSD. " Why Distributed Simulation?" AEROSPATIALE, DLR, 1997.

[5]. Dahman, Judith, R., Fujimoto, and R., Weatherly. "THE DoD HIGH LEVEL ARCHITECTURE: AN UPDATE." International Workshop on Distributed Interactive Simulation and real-time Application, July 1998.

[6]. Barber, Suzane, and C., Chase. "Distributed Simulation Project." Report 1998.

[7]. Sapaty, Peter, and Steve, Seidensticker. "Mobile Intelligence in Distributed Simulation".

[8]. Bagrodia, R., K., Chandy, and Wen, Liao. "A unifying framework for distributed simulation." ACM Transaction on Modeling and Computer Simualtion, Vol. 1, No. 4,Oct. 1991, pp. 348-385.

[9]. Ferenci, Steve, R. Fujimoto. "RTI Performance on Shared Memory and Message Passing Architectures."

[10]. Tacic, Ivan, R. Fujimoto. "SYNCHRONIZED DATA DISTRIBUTION MANAGEMENT IN DISTRIBUTED SIMULATIONS."

[11]. Krajewski, L., L.,Ritzman. "Operation Management Strategy and Analysis." ADDISON-Wesely Publishing Company, Inc. 1993.

[12]. Fujimoto, Richard, Richard, Weatherly. "HLA TIME MANAGEMENT AND DIS."

[13]. Fujimoto, Richard. "Exploiting Temporal Uncertainty in Parallel and Distributed Simulation."

[14]. Foster, G., J., Glover, K., Warwick. "Flexible Distributed Control of Manufacturing Systems Using Local Operating Networks."

[15]. Glover, J., G., Foster, and K., Warwlck. "Parallel Discrete Event Simulation of Production Lines Using Local Operation Networks (LONS)."

[16]. Parsons, David, L., Krause. "Tactical Logistics and Distribution Systems (TLODAS) Simulation."

[17]. Lucas, Thomas, R., kerchner, J., Friel, and D., Jones. " A Guide for Analysis Using Advanced Distributed Simulation (ADS)." Rand, 1997.

[18]. Page, Ernest, R., Moose, and S., Griffin. " Web-Based Simulation in SIMJAVA Using Remote Method Invocation." Computer Based Learning Unit, University of Leeds, 1996.
[19]. Pesce, Mark. "Distributed Behavior in Distributed Simulations." Virtual Reality Systems Conference, New York City, October 1993.
[20]. Weatherly, R., A., Wilson, B., Canova, E., Page, A., Zabek. "Advanced Distributed Simulation through the Aggregate Level Simulation Protocol." International Conference on System Sciences, Hawaii, January 1996.
[21]. Narayanan, D., A., Bodner, U., Sreekanth, T., Govindaraj, L., McGinnis, and C. Mitchell. "Research in Object-Oriented Manufacturing Simulations: An Assessment of the State of the Art." IIE Transaction.
[22]. Perumalla, K., R., Fujimoto. " Interactive Parallel Simulations with the JANE Framework." College of Computing, Georgia Institute of Technology, http://www.cc.gatech.edu/computing/pads/ted.html.
[23]. "Multiresolutional Visualization of Evolving Distributed Simulations." http://www.erc.msstate.edu/~xsdu/ResearchSec1.html
[24]. "Distributed Simulation." NRL Code 5585:The Visualization and Computing Systems Section http://www.ait.nrl.navy.mil/5585/ADS
[25]. Weisshuhn, Peter. Innovation Gateway. http://newmedia.pira.co.uk/pubtech/oig/index.html
[26]. " Distributed Simulation." http://www.os.kcp.com/home/amnii/ds.html
[27]. Fujimoto, R., R., Weatherly. "HLA Time Management and DIS."
[28]. Ferenci, Steve, R., Fujimoto. "RTI Performance on Shared Memory and Message Passing Architectures." College of Computing, Georgia Institute of Technology, http://www.cc.gatech.edu/computing/pads/ted.html.
[29]. Parsons, D., L., Krause. "Tactical Logistics and Distribution Systems (TLODAS) Simulation." Simulation Dynamics, Inc.
[30]. Glover, J., g., Foster, and K., Warwick. "Parallel Discrete Event Simulation of Production Lines Using Local Operating Networks (LONS)." Department of Cybernetics, The University of Reading.
[31]. Fujimoto, Richard. "Exploiting Temporal Uncertainty in Parallel and Distributed Simulation." College of Computing, Georgia Institute of Technology, http://www.cc.gatech.edu/computing/pads/ted.html.
[32]. Carothers, Christopher, K., Perumalla., and R. Fujimoto. "Efficient Optimistic Parallel Simulation using Reverse Computation." College of Computing, Georgia Institute of Technology, http://www.cc.gatech.edu/computing/pads/ted.html
[33]. Tacic, Ivan, R. Fujimoto. "Synchronized Data Distribution Management in Distributed Simulation." College of Computing, Georgia Institute of Technology, http://www.cc.gatech.edu/computing/pads/ted.html
[34]. Harrell, Charles, K. Tumay. "Simulation Made Easy." Industrial Engineering and Management Press, Institute of Industrial Engineers, Georgia 1995.

[35]. Page, Ernest. "The Rise of Web-Based Simulation: Implication for the High Level Architecture." Proceedings of the 1998 Winter Simulation Conference.

[36]. Page, Ernest. "Web-Based Simulation in SIMJAVA Using Remote Method Invocation." Proceedings of the 1998 Winter Simulation Conference.

[37]. Fujii, S., Y., Kidani, A., Ogita, T., Kaihara. "Synchronization Mechanism for Integration of Distributed Manufacturing Simulation Systems." Simulation, 1999,pp. 1-11.

[38]. Foster, G., K., Wawick. "Flexible Distributed Control of Manufacturing Systems Using Local Operating Networks." Department of Cybernetics, The University of Reading.

[39]. Page, E., B., Arnold, P., Fishwick, K., Healy, R., Nance, and R., Paul. "Web-Based Simulation: Revolution or Evolution."

[40]. Page, E., J., Opper. "Investigating the Application of Web-Based Simulation Principles within the Architecture for Next-Generation Computer Generated Forces Model."

[41]. Page, Ernest. "Beyond Speedup: PADS, the HLA and Web-Based Simulation."

[42]. Page, E., S., Griffin, and L., Rother. "Providing Conceptual Framework for Distributed Web-Based Simulation within High Level Architecture."

[43]. Sawhney, Rapinder, Mark, Upton, and Ammar, Aamer. "Flexible Simulation Modeling as a Mechanism to Pervasively Implement Simulation in the Automotive Industry. Society of Automotive Engineering, Inc. 1997.

[44]. Monden, Yasuhiro. "Toyota Production System." Engineering & Management Press, Georgia, 1998.

[45]. Kelton, W., R., Sadowski, and D., Sadowski. "Simulation with Arena." WCB McGraw-Hill, Inc. 1998.

[46]. Pedgen, C., R., Shannon, and R., Sadowski. "Introduction to Simulation Using SIMAN." McGraw-Hill, Inc. 1995.

[47]. MICROSOFT site. http://www.microsoft.com/windows/netmeeting/

[48]. McLean, Charles, F., Riddick, and Al Jones. "An Architecture for Distributed Manufacturing Simulation." National Institute of Standard and Technology, Manufacturing Systems Integration Division, Gaithersburg, MD, October 1999.

VITA

Ammar Mohamed Aamer was born in Sana'a, Yemen on May 11, 1974. He attended schools in Sana'a, Yemen, where he was graduated from the Alahlia High School in June 1992. In August 1994, he entered Maryville College in Maryville, TN; he then transferred to Pellissippi State Community College in Knoxville, TN, and attended The University of Tennessee in Knoxville in 1996. He received his Bachelor of Science in Industrial Engineering in May 1998 with Cum Laude honors. In June 1998, he started his Master's of Science with a manufacturing concentration and worked as a research assistant for Dr. Rapinder Sawhney. He has been working on different simulation projects locally and nationally under the supervision of Dr. Sawhney. The Master's degree was received in December, 1999.

He is presently pursuing his Doctorate of Engineering Science with an Industrial Engineering concentration.