*Proceedings of the 1989 Winter Simulation Conference*
E.A. MacNair, K.J. Musselman, P. Heidelberger (eds.)

# ARCHITECTURE FOR AN INTEGRATED SIMULATION/CIM SYSTEM

Gordon M. Clark
The Ohio State University
1971 Neil Avenue
Columbus, Ohio 43210

David H. Withers
IBM Corporation
P.O. Box 28331
Atlanta, Georgia 30358

## ABSTRACT

CIM-OSA (Computer Integrated Manufacturing - Open Systems Architecture) is the result of a joint European initiative by nineteen industrial and academic organizations to define an architecture for a CIM system. This architecture permits CIM users and vendors to evolve toward well-defined structures for CIM systems. Moreover, this architecture permits an integrated decision support system implementing simulation. We define integration criteria for this decision support system, and an architecture in response to these criteria. One of the features of this architecture is the ability to emulate the CIM control logic to perform "what-if" and "try-for-fit" analyses. We define a potential problem with simulation when emulating CIM control logic.

## 1. INTRODUCTION

The application of discrete event simulation to manufacturing problems in systems design, analysis, and operation has become an increasingly attractive methodology recently. New languages and tools for development of simulation models in the manufacturing environment continue to be developed and marketed. However, very few of these tools are integrated with the manufacturing information system and almost none are integrated with the manufacturing business process definitions and control logic. This paper describes an architecture for simulation in a Computer Integrated Manufacturing (CIM) environment that allows simulation to be integrated with the rest of the CIM system - both functionally and for information access.

The CIM-OSA (Open Systems Architecture) Reference Architecture Specification is the result of a joint European initiative by nineteen industrial and academic organizations. It defines an architecture for a CIM system (CIMOSA.). This architecture permits CIM users and vendors to evolve toward well-defined structures for CIM systems. Moreover, this architecture permits the implementation of simulation in an integrated decision support system. We define integration criteria for this decision support system, and an architecture in response to these criteria. One of the features of this architecture is the ability to emulate the CIM control logic to perform "what if" and "try for fit" analyses.

This paper is organized as follows: We first introduce the concepts from the CIM-OSA Implementation Model and identify the key features of a CIM Simulation system. We then discuss emulation of a CIM control system and develop model structures to support traditional simulation models as well as integrated CIM models. We also identify a potential problem with simulation when emulating the CIM control logic.

## 2. CIM-OSA IMPLEMENTATION MODEL

The CIM-OSA Reference Architecture Specification begins with generic constructs for modeling business processes, information, resources, and organization. As an enterprise develops particular models the implementation model provides guidance for a unified way to implement a solution. The implementation structure is shown in Figure 1.

This implementation model provides a way to develop CIM application software that is independent of the computing hardware and operating systems, independent of data storage methods and location, and independent of the context where the applications will be used. The Building Blocks in the Implementation Model are:

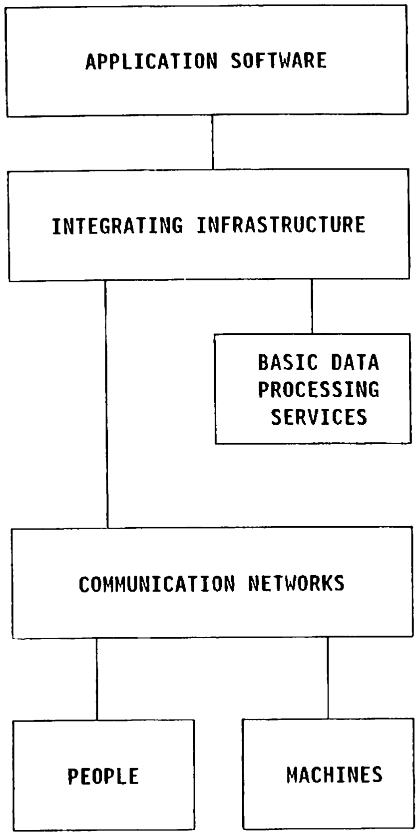


Figure 1. CIM-OSA Implementation Model

**Application Software:** Typical CIM applications might include (Jarvis(1986), Scheer(1988), and Yoemans, Chaudry, and ten Hagen (1985)):

- Computer Aided Design
- Process Planning
- Engineering Release
- Material Requirements Planning
- Work Flow Control
- Project Management

**Integrating Infrastructure:** Will be described in detail in Figure 2 and the section following the figure.

**Basic Data Process Services:** Typical data processing services include:

- Computer hardware
- Operating system software
- Database Management System
- Graphical display services

**Communications Networks:** The protocols and associated processing services for communicating with other entities such as people, devices, and other computers.

The implementation model also provides common protocols for interactions among machines, humans, and application programs across a network of CIM systems. The model shown in Figure 1 can be replicated onto several computing systems and connected via the common protocols.

## 2.1 INTEGRATING INFRASTRUCTURE

This building block is the focus for the CIM/Simulation architecture we have developed. The integrating infrastructure shown in Figure 2 enables system, data, and context independence for applications.

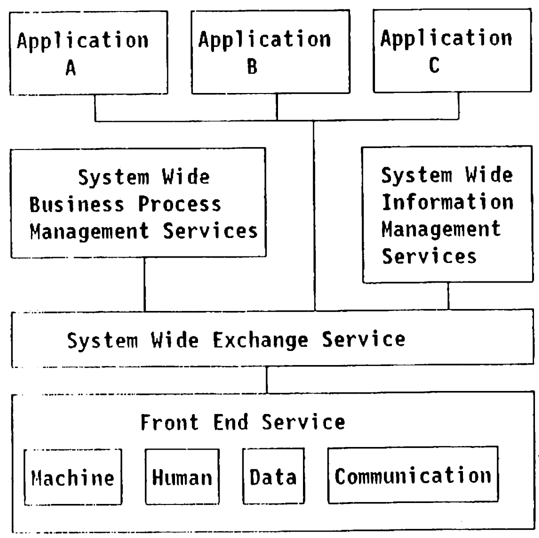


Figure 2. CIM-OSA Integrating Infrastructure

The *Front End Services* insulate the applications and other services from the differences in equipment, display terminals, communications protocols, etc.

The *System Wide Exchange Service* directs all requests for Front End Services and Applications to the correct destination. This service insulates the applications from knowledge of the services chosen for a particular implementation.

The *System Wide Business Process Management Services* and *System Wide Information Services* insure execution of the correct applications to accomplish a business process and route requests to the correct service for information exchange.

We have displayed the *Applications* in a different position than in the original descriptive reference so that the independence from execution context can be shown more easily later in this paper.

## 3. KEY FEATURES OF A CIM/SIMULATION SYSTEM

Before we can show how the CIM-OSA Integrating Infrastructure can provide an architecture for integrating function and data between Simulation and other CIM applications, we need to describe the key features of such a system. Simulation is considered to be part of the decision support system and needs to conform to the attributes for these systems, see for example (Sprague(1982)). Simulation supports a variety of users such as engineering, production planning, and plant floor operations. The application supports a variety of modeling skills from persons with no knowledge of simulation, programming, or modeling to experienced simulation model developers. Access is required to all of the information in the CIM data base(s). Simulation needs to support "try-for-fit" and "what-if" analysis that are not structured in advance and include any of the operational CIM processes. Simulation needs to be viewed from the same user interface as the user would see for normal operation of the system. Most importantly, simulation needs to replicate the processes and control that are used for normal operation of the CIM system. Additionally it must be possible to add simulation models to model activities that are external to the CIM system.

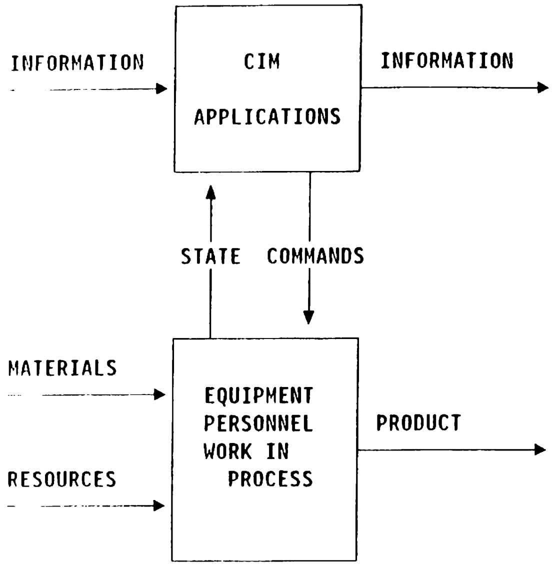


Figure 3. Manufacturing System

## 4. OPERATION, EMULATION, and SIMULATION

The manufacturing system shown in Figure 3 will be the basis for describing how the above requirements for a decision support system can be met using the CIM-OSA Integrating Infrastructure.

In this system, external information flows into the CIM applications. They also receive state information from personnel, equipment, and possibly other systems. The applications provide commands to the personnel and production equipment so that materials and resources can be transformed into product.

This same structure for a manufacturing system can be changed so that a *Global Model* can represent the same control logic and business process in emulation mode. This is shown in Figure 4.

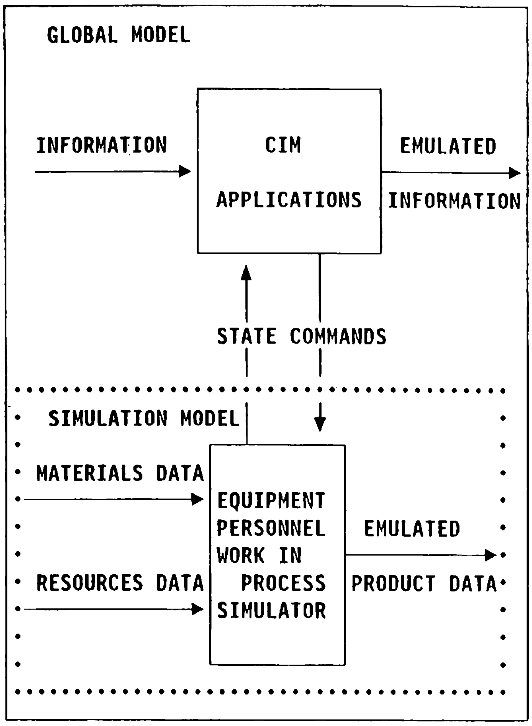


Figure 4. Emulating the Manufacturing System

A *Simulation Model* replaces the equipment and personnel that were part of operation of the manufacturing system. The Simulation Model may in fact be several models, e.g., one for an automated work cell, one for an automated materials movement system, and one for a manual work station. Emulated information is produced from the applications. Emulation of manufacturing control systems was first defined by Crookall (1985). The National Technical Institute (formerly NBS) has implemented an emulation system for hierarchical control that allows the investigation of system behavior before the production hardware is available (Furlani(1985)). Harmonosky and Barrick (1988) discuss the use of simulation logic for real time control and HEI, Inc. has a US patent for a system design that permits the same control logic to be used to operate a materials handling system or to view a graphical display of the system dynamics without operating it (HEI(1985)). Withers (1987) defined a way for integration of simulation, emulation, and operation of a manufacturing control system that is the predecessor to the work reported here.

In addition to the emulation mode defined above, there are occasions when there is no CIM application with the required control logic or business process. The simulation model becomes the global model in this case as shown in Figure 5.

In the next section we will define the classes of global models more precisely. The significant point to be made is that all of the types of simulation activity can be supported by the same architecture. These activities range from simple testing of a new software release to "what-if" analysis of plant floor rearrangements. This same architecture integrates the simulation activities with the CIM data so that redundant copies, additional input, and outdated data values are no longer issues with simulation.

## 4.1 GLOBAL MODELS

There are three classes of global models:

- Sequence of Application Functions
- Sequence of Application Functions supplemented with one or more Simulation Models
- A Simulation Model only

Each class is described below and shown as at conceptual level in a diagram.

## 4.2 SEQUENCE OF APPLICATION FUNCTIONS

The first class of global models is a sequence of application functions (A.F.x in Figure 6). In normal operation they neither receive state information from production equipment nor send commands to external devices. The only thing that makes the Global Model different from normal operation is the execution context which will be described later.

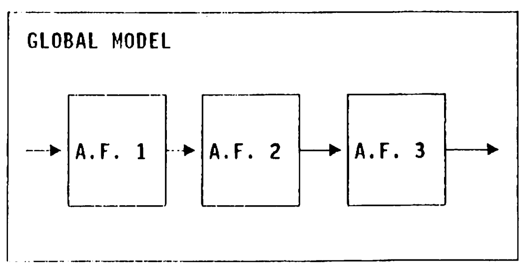


Figure 6. Global Model - Sequence of Application Functions

Examples of this class are:

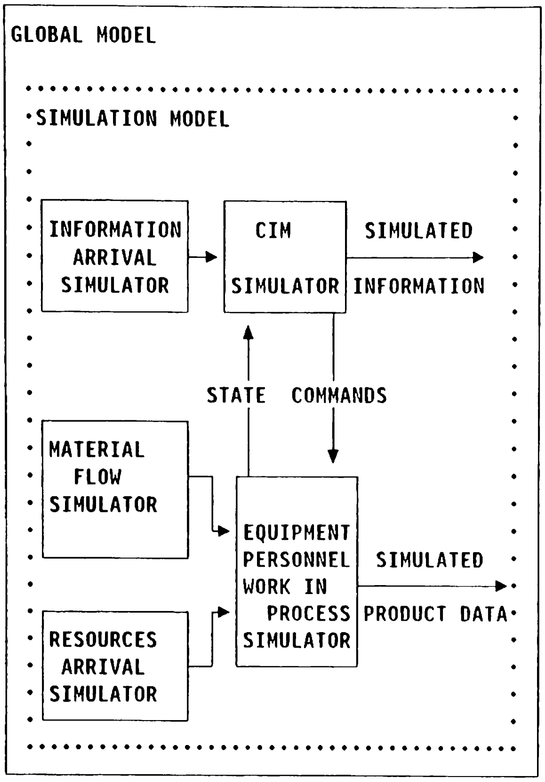


Figure 5. Traditional Method for Simulation

- Using the Cost Estimation application to estimate the cost of a new part
- Using the Purchasing application to train a new purchasing agent on the use of the application.

### 4.3 APPLICATION FUNCTIONS WITH A SIMULATION MODEL

This class of global models allows the substitution of *Simulation Model(s)* for personnel, equipment, and external systems. The application functions are also used as part of the global model as in the class above. This class is shown in Figure 7.

Examples of this class of global models are:

- Shop floor control system controlling a flexible manufacturing system with simulation models of machines and transporters
- Work Flow Control system controlling a production system with simulation models of manual work stations.

### 4.4 SIMULATION MODEL ONLY

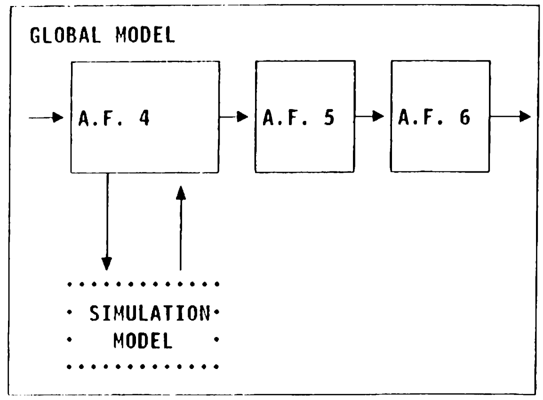


Figure 7. Global Model - Application Function(s) with Simulation Model(s)

This class of global models corresponds to traditional discrete event simulation models. Application functions are not part of the global model because they either don't exist or are not needed for the current investigation. This global model, however, is integrated with the CIM system for information exchange.

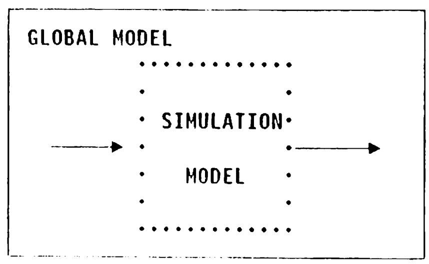


Figure 8. Global Model - Simulation Model Only

Examples for this class of global models include:

- Shop floor control system for the design of a new line for assembling a microcomputer
- Queueing analysis of trucks arriving at the receiving department.

## 5. EXECUTION CONTEXT

As we mentioned earlier, the *Execution Context* is the only difference between normal operation and some global models. The Execution Context is the definition of the information sources and destinations. It is most easily described by comparing a simple business process in normal operation and the same process in simulation context. In Figure 9 a production control application receives manufacturing routing information from a routing file, sends commands to a cell controller, receives production information from the controller and maintains a history file of production information.

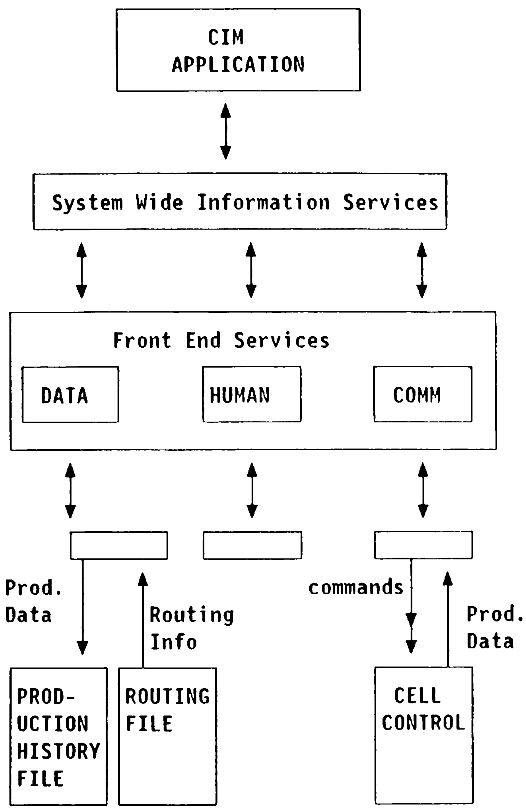


Figure 9. Example of a CIM Application in Normal Operation

In Figure 10 the application receives manufacturing routing information from a routing file (as before), sends commands to a simulation model, receives simulated production data from the simulation model and displays this information on a terminal. Note that the control logic is the same for both operation and simulation.

## 6. POTENTIAL IMPLEMENTATION PROBLEM

In the last example the same application code was used for operation of the production facility as for simulation to investigate perhaps an alternate shop order schedule. A simulation model replaced the

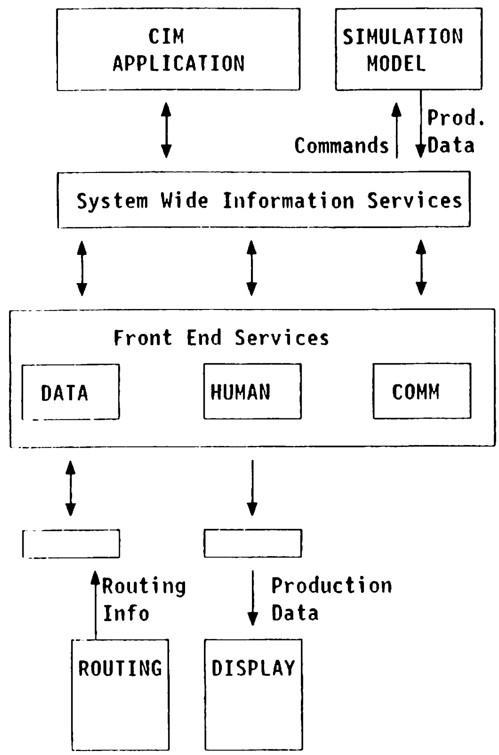


Figure 10. Example of CIM Application in Simulation Context

actual production equipment. Even if the model mimics the actual production delays, the computing delays incurred by the CIM control system may be different. The simulation system will not identically mimic the CIM control system if either:

- Transmissions arrive in a different sequence, or
- Control decisions are made in a different sequence.

This phenomena can occur even if the CIM control programs are identical for both operation and simulation. This anomaly may be more pronounced for systems that operate on distributed computing hardware for normal operation and are placed in simulation mode on a single processor. Further research is required to quantify this problem.

## 7. SUMMARY

The research reported in this paper has established that an integrated CIM/Simulation system architecture can be developed from the CIM-OSA architecture. The resulting system allows the same control logic, information bases, and user interfaces to be used for operation or simulation.

## REFERENCES

*CIM-OSA Reference Architecture Specification*, CIM-OSA/AMICE, 489 Avenue Louise, B 14-B-1050, Brussels, Belgium (undated)

Crookall, J.(1985). Simulation, Emulation and Computer Integrated Manufacturing, Technical Report, Cranfield Institute of Technology, Cranfield, UK.

Furlani, C.M. (1985). *Hierarchical Control System Emulation User's Manual*, National Bureau of Standards, Gaithersburg, MD, (order PB85-233849/XAB)

Harmonosky, C.M. and D.C. Barrick (1988). Simulation in a CIM environment: Structure for analysis and real-time control, *Proceedings of the 1988 Winter Simulation Conference*, pp 704-711.

HEI Inc. (1985). U.S. Patent Number 4,512,747.

Jarvis, John W. (1986). *Computer Integrated Manufacturing Technology and Status - An Overview for Management*, Advanced Manufacturing Information at the Institution of Electrical Engineers.

Scheer, A. W. (1988). *CIM - Computer Integrated Manufacturing - Computer Steered Industry*, Springer Verlag.

Sprague, R. H., and E. D. Carlson (1982). *Building Effective Decision Support Systems*, Prentice-Hall, Inc.

Withers, D.(1987). Integration of Simulation, Emulation, and Operation of a Manufacturing Control System, IBM TR 52.0011, Atlanta, GA.

Yoemans, R. W., A. Choudry, and P. J. W. ten Hagen (1985). *Design Rules for a CIM System*, Elsevier.

## AUTHORS' BIOGRAPHIES

GORDON M. CLARK is a professor of Industrial and Systems Engineering at the Ohio State University. During the 1988/89 academic year he was on sabbatical with the IBM Corporation in their Application Systems Division. He received the B.I.E. degree in Industrial Engineering from The Ohio State University in 1957, a M.Sc. in Industrial Engineering from the University of Southern California, and a Ph.D. from The Ohio State University in 1969. He was responsible for the development of several combat simulations at The Ohio State University while under contract with the U.S. Army (1965-1974). He was Associate Editor of *Operations Research* (1975-1978), and a member of the Board of Directors of the Military Operations Research Society (1972-1976). He has consulted in the areas of simulation modeling and analysis and manufacturing systems for industrial companies and Battelle Memorial Institute. His research interests include simulation optimization, applications in manufacturing systems, and approximations to nonstationary queuing systems. He is a member of ORSA, TIMS, IIE, SCS, and SIGMA XI.

Gordon M. Clark
Department of Industrial and Systems Engineering
The Ohio State University
Columbus, OH 43210
(614) 888-1746

DAVID H. WITHERS is a Senior Planner at the IBM Application Systems Division laboratory in Atlanta, Georgia. He received a B.S. from the U.S. Coast Guard Academy in 1962 and M.S. degrees in mathematics and computer science in 1966 and 1969 from Rensselaer Polytechnic Institute. His assignments with the U.S. Coast Guard included Commanding Officer of a Cutter and Instructor on the faculty at the Academy in New London, Connecticut. He joined IBM in 1969 and has held a variety of technical and management positions. Prior to his current assignment he was the Manager, Product and Process Analysis at the IBM T.J. Watson Research Center in Yorktown, New York. Currently he is responsible for requirements integration for software for computer integrated manufacturing products. He is a member of ACM, ORSA, TIMS, SIGSIM, and SCS.

David H. Withers
IBM Corporation
1500 RiverEdge Parkway
P.O. Box 28331
Atlanta, GA 30358
(404) 956-5289