Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

GENERAL PURPOSE ENTERPRISE SIMULATION WITH MASTER

Willi Bernhard
Marco C. Bettoni

Basel Institute of Technology
CIM-Zentrum Muttenz
CH - 4132 Muttenz, SWITZERLAND

ABSTRACT

This article describes the concept of a knowledge-based assistant (MASTER) for speeding up simulation projects. It is able to simulate and optimise whole enterprises of the types: discrete part-manufacturing, continuous process-manufacturing and service-companies. It consists of well proven, powerful software-packages, all of them are commercially available. These are the expert-system building tool "KEE", the simulation language "SIMSCRIPT", the animation-package "SIMGRAPHICS" and the spreadsheet-package "LOTUS-123". The packages are embedded in a UNIX environment. At present, this system is being developed and carries the name MASTER which stands for "Muttenzer Assistant for Simulation Tasks with Expert Reasoning".

1 INTRODUCTION

In general, an enterprise can be modelled as consisting of three kinds of flows and their relationships: the material-flow, the information-flow and the value-flow (money-flow). In order to simulate, analyse and optimise an Enterprise-System you must be able to model these time-dependent flows and all the appropriate relationships.
One step in a simulation-project is the use of a simulation software which is able to handle the complexity of the Enterprise-System. In order to achieve a high modelling flexibility and also a low modelling difficulty, our approach was to combine a powerful general-purpose simulation language with an expert-system. The knowledge-based interface makes it user-friendly and allows a graphical model-construction. This approach led to a design concept consisting of a) a simple and powerful modelling technique and b) the system concept of MASTER.

Today, the CIM-Zentrum Muttenz (CIM Center) has three main tasks: Consulting & Know-how transfer, the Education of students in a 1-year Post-Graduate study and Research & Developments in the area of Enterprise-Simulation.

The idea of having a model of an enterprise to train people on CIM-Concepts came up in 1989. It was clear that only software-based simulation models would be flexible enough to make the processes and relationships of an enterprise understandable. After that, several simulation packages and simulation languages were tested but no tool could meet all the requirements such as a user-friendly interface, graphical based modelling, powerful modelling capabilities almost unlimited in size and complexity, animation of the whole enterprise and it's sub-systems, avoiding the use of programming languages (Pascal, Fortran, C, ...) , software-portability, and so on. Therefore, we decided to develop a simulation assistant based on a well-known and powerful general-purpose simulation language.

2 IMPORTANT ASPECTS OF AN ENTERPRISE

An enterprise is a dynamic system, where some work has to be performed and material, information and money is involved. Therefore, our main interest is in the material-, information- and value-flows and their relationship with one another which defines the system. It does not make a difference if you deal with a part-manufacturer, process-manufacturer or a service company.

The simulation output can give you indications of:

- bottlenecks in the system
- the effects of changes in the organisation structure
- the effects of interruptions (machine, manpower, supplier, ...)

- throughput
- production time and terms of delivery
- productivity
- waiting times
- production costs
- work-in-process
- buffersize and inventory
- utilisation of resources
- synchronisation of the flow of material and information

In order to optimise the enterprise, you have to define and modify parameters based on given results.

3 THE SYSTEM CONCEPT OF MASTER

An advantage of a simulation language is it's modelling flexibility but you have to be a simulation expert in order to be able to use it. No graphical support is given when you set up models with a wordprocessor. This can be tolerable for an expert, but for a novice it is not.

MASTER consists of four basic-packages: An Expert-System, a Simulator, an Animation-Package and a Spreadsheet program (Figure 1).

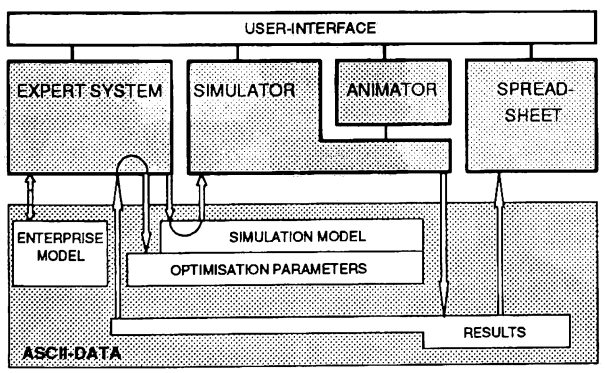

Figure 1: The system "MASTER"

The Expert-System implements our modelling techniques and allows you to create the enterprise model graphically. Then it writes the simulation model in SIMSCRIPT code like a human expert would have to do manually. All the items required will be defined including the statements for the animation.

The Expert-System knows the rules and methods as well as the syntax, exceptions and specialities needed to write a verified model.

Next, the simulator compiles the model, simulates it and drives the animation package SIMGRAPHICS which shows how objects move through the system of interest. A spreadsheet program takes charge of the results and prepares them for accounting as well as for presentation-graphics with plots, pie-charts, some statistics etc.

Another task is the optimisation of user-defined parameters. Simulation results can take a loop back to the expert-system which analyses the data and varies optimisation-parameters controlled by special rules and methods. The simulation repeats as many times as necessary in order to find an optimum.

4 THE MODELLING CONCEPT OF MASTER

One important requirement for the modelling technique is that it must be simple. This means that it is easy to understand on the one hand and easy to learn on the other. Therefore, a 3-level approach is used (Figure 2).

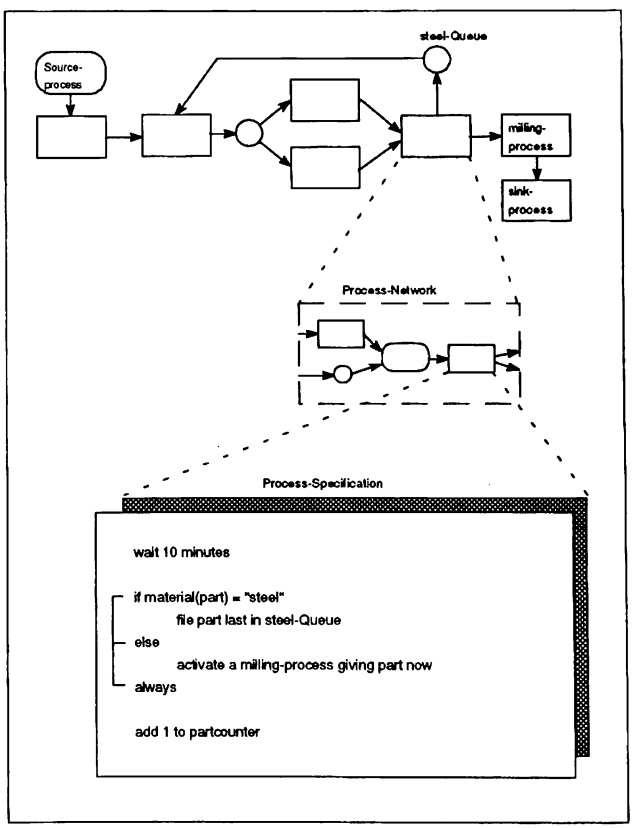

Figure 2: The 3-level modelling concept of MASTER

The first level is called Process-Network; it lets you create the system via modified DataFlow Diagrams.

The intention is to split up the system in Processes and Queues whereby several layers of detail are allowed. The purpose of the layers is for information hiding and not for pure hierarchical modelling. The elements for the construction of the Process-Network are shown in Figure 3.

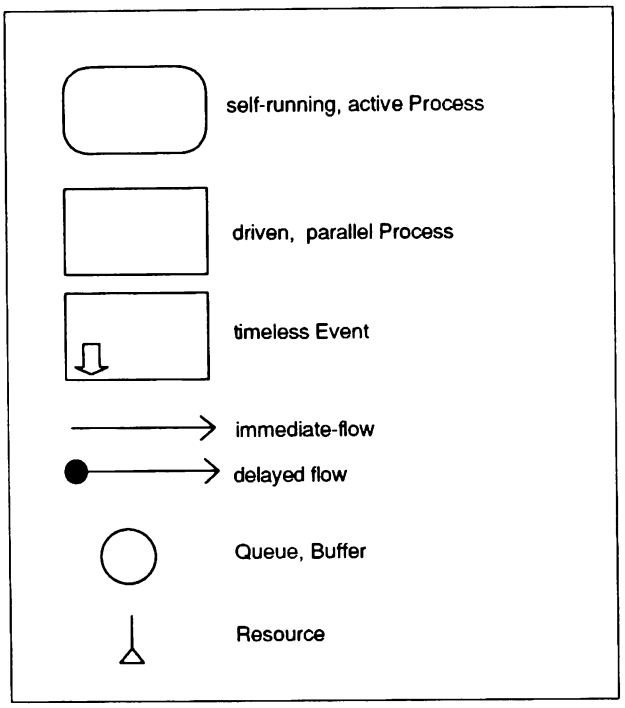

Figure 3: The seven elements for the Process-Network

There are two kinds of processes, a self-running process called "active process" and a driven process called "parallel process". For timeless tasks the "timeless event" is introduced. Further, you have the Queue or Buffer as a passive element where objects can wait for an unpredictable time. The connection always has a flow-direction and can be immediate or delayed.
And, since frequently used, you have a special symbol for Resources. There are no restrictions on how to put these elements together.

The second level is called Process-Specification. You have to describe exactly what happens in the Processes, Events and Queues. This defines the control logic and all the conditions and strategies of the modelled enterprise. The description is an "English like" language whereby the expert-system gives a choice for process-related functions, object-related functions, value-related functions, information-related functions and functions for control-structures (Figure 4). The descriptions are structured like Action-Diagrams (see also Figure 7) .

The third level is called Model-Dictionary and works as a datadictionary (Figure 5). This part helps you to define and find out all the meanings of your elements, names, variables, attributes, parameters and other data.

Process-related functions:
create, activate, suspend, interrupt, resume, wait, work, resource-request, resource-relinquish, destroy,...

Entity/Object-related functions:
create, file, remove, perform, destroy, ...

Value-related functions:
arithmetic functions, algebraic functions, boolean functions, analytical functions, statistical functions

Information-related functions:
define, let, find, compute, call, schedule, cancel, read, write, print, ...

Control structures:
condition, selection, iteration, sequence
(If-else-always, select-case, do-loop)

Figure 4: Functions for the Process-Specification

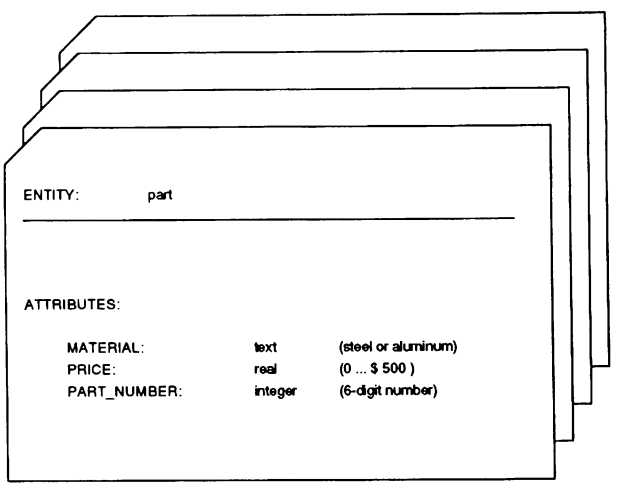

Figure 5: Example of the Model-Dictionary

There are only two requirements for a successful modelling of an enterprise within this concept:

Every single element used to model the system must be known and you must be able to describe it in detail (think of the Process-Specification)

The relationships between the processes must be known and you must be able to associated with one another (think of the Process-Network).

Normally, the objects moving through the network of processes are passive. The processes push and pull them through the network. It is possible to give the objects some attributes of type real, integer, text, pointer, array as well as queue and process. The last case is of special interest because it makes the object active. For example, if an object has to change it's priority, it's process can do this on it's own and based on it's own strategy.

There are no limits concerning size or complexity of the model - the enterprise can be as complex as you want. The only requirement is that you must be able to describe your system. You are supported by a three-level approach: Process-Network, Process-Specifications and a Model-Dictionary.

5 THE COMPONENTS OF MASTER

All the packages in MASTER are installed on the operating system UNIX and the window system X-WINDOWS.

5.1 The Expert System

The expert-system building tool KEE (IntelliCorp, USA) is used for the construction and operation of the expert-system. It is one of the most powerful and successful AI-Systems on the market. The software kit is based on the AI-language LISP.

KEE consists of two main components, the KEE Language for the construction of the knowledge base and the KEE Support Environment for the run-time support.

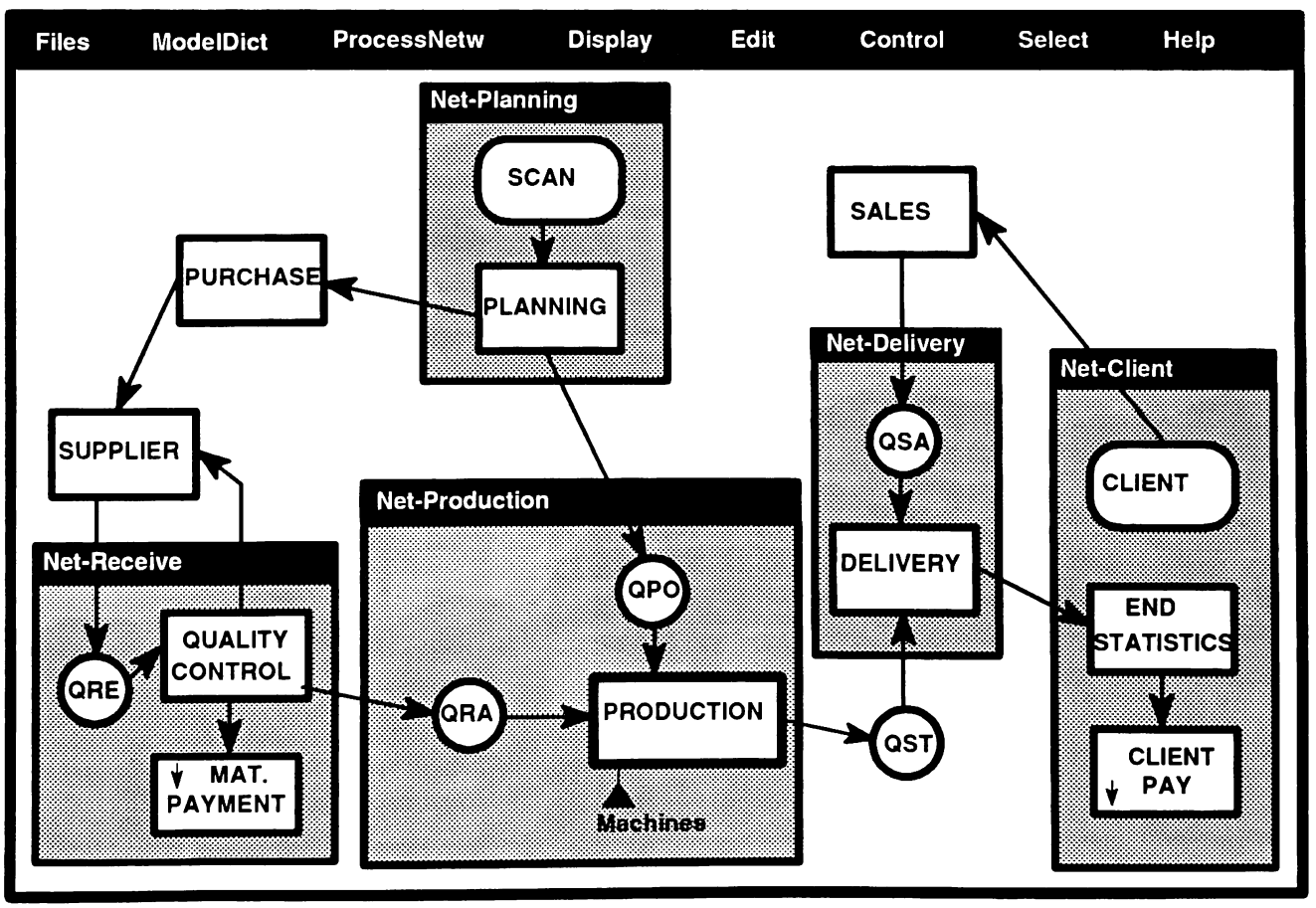

Figure 6: User-interface for the Process-Network (all Sub-Networks opened) illustrated in an example

As already mentioned, the expert-system has three main tasks: the graphical and knowledge-based support for the model construction, the source-code generation and the optimisation of model-parameters.An example of a Process-Network and a Process-Specification is given in Figure 6 and Figure 7 to illustrate the user-interface provided by KEE and LISP.

5.2 The Simulator

The general purpose simulation language SIMSCRIPT II.5 (CACI, USA) is used for the simulation task. It has an "English like" freeform syntax, is easy to read and is almost self-documented. SIMSCRIPT is well-known for its powerful data-structures and the ability to handle large and complex process-oriented models. Although it is a discrete-event simulation language, it has some features for continuous simulation.
SIMSCRIPT has a very good portability and is available on a wide range of hardware-platforms and operating systems.

The SIMSCRIPT package also contains support-tools like the DATAGRAPH for the analysis of output-data and SIMGRAPHICS for the animation and the construction of user-defined menus, input-forms, icons, layout-drawings and (dynamic) presentation graphics. A debugger and various interfaces to other programming languages and I/O devices are also provided.

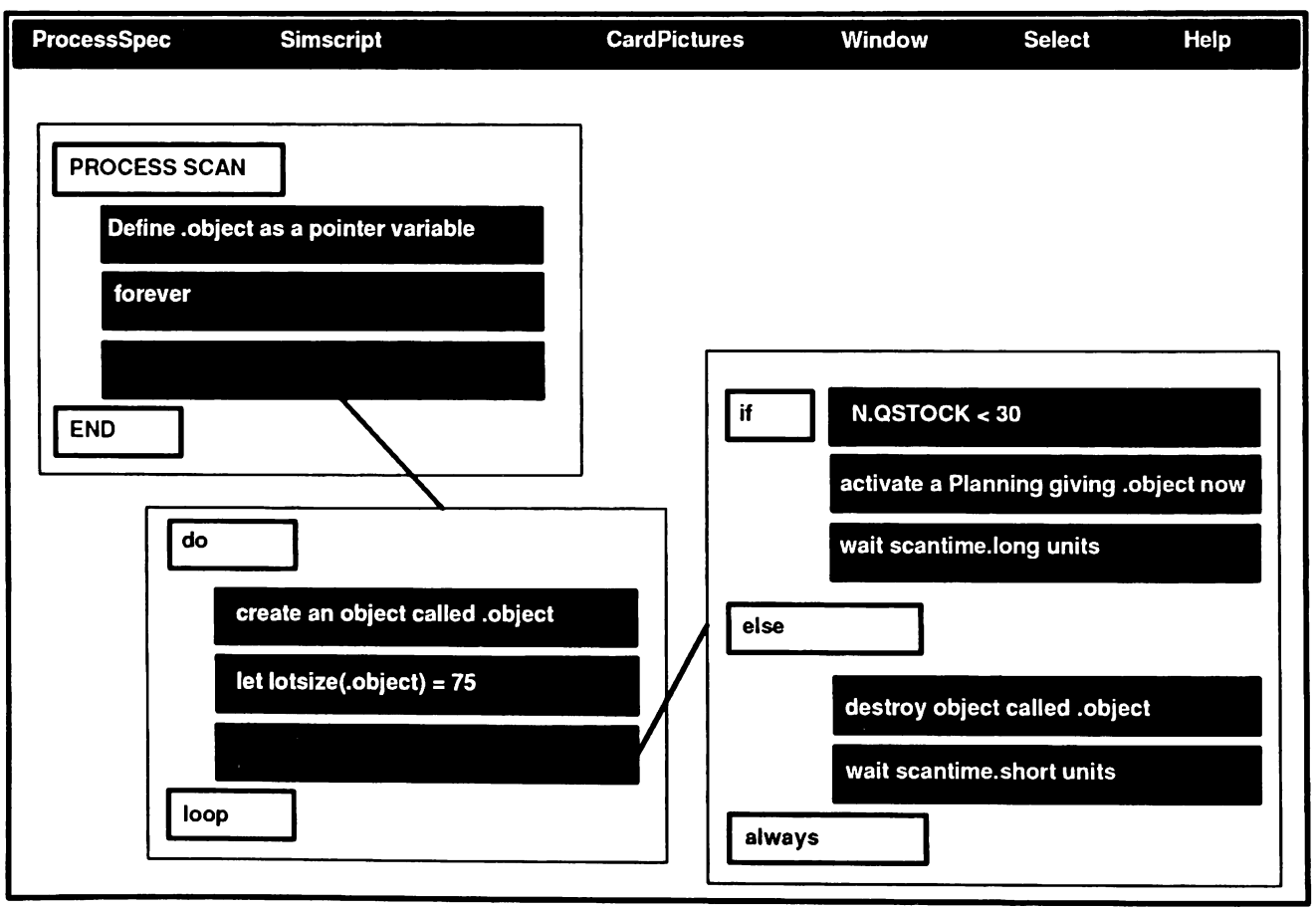

Figure 7: User-interface for the Process-Specification illustrated in an example

5.3 The Animator

The animation package SIMGRAPHICS (CACI, USA) is applied to the visualisation of the dynamic behaviour of the system. You can get moving objects as well as dynamic presentation graphics (levels, plots, pie-charts, histograms, clocks and traces). The concurrent animation is driven by SIMSCRIPT. It is also possible to record an animation and play it back by using the tool SIMREPLAY.

There is no programming effort for the construction of menus, icons, layouts and forms. Whenever needed, a printout can be produced via a postscript interface. SIMGRAPHICS has the same portability as SIMSCRIPT.

5.4 The Spreadsheet

The business-package LOTUS-123 (Lotus Corporation, USA) is mainly used for the accounting of a simulated enterprise. This package is well known on personal computers. The UNIX version does not differ in functionality from the version used on personal computers and is therefore not further described.

6 PRACTICE & EXPERIENCES

The modelling-concept of MASTER has been successfully applied to several simulation-projects within small and medium sized Enterprises in Switzerland. We often experienced the fact, that employees from enterprises described their system as a form of connected boxes even if we did not say what type of model-description they had to use. It is not a difficult task to extend such models to a Process-Network and its Process-Specification - just ask where the (forgotten) Queues are and request a detailed process-description in form of a protocol. In obstinate cases, an interview technique will help.

REFERENCES

Bernhard W., Bettoni M. 1992. Muttenzer Modellunternehmung, Konzept. CIM Zentrum Muttenz.

Charniak E., McDermott D. 1985. An Introduction to Artificial Intelligence. Addison-Wesley.

Demarco T. 1979. Structured Analysis and System Specification. Prentice Hall.

Estrine E., Directory of Simulation Software. 1991. Society for Computer Simulation.

IntelliCorpp. 1991. KEE Technical Manuals, Vols. 1-4. IntelliCorp Inc.

Martin J., McClure C. 1988. Structured Techniques. Prentice Hall.

Noche B., Wenzel S. 1991. Marktspiegel zur Simulationstechnik in Produktion und Logistik. Verlag TÜV.

Powers M., Cheney P., Crow G. 1990. Structured Systems Analysis. Boyd & Fraser.

Russel E. 1992. Building Simulation Models with SIMSCRIPT II.5. CACI Products Company

Waterman D. 1986. A Guide to Expert Systems. Addison-Wesley.

AUTHOR BIOGRAPHIES

WILLI BERNHARD is a Professor in the CIM Center at the Basel Institute of Technology. He received his Swiss Dipl.El.Ing.HTL degree in electrical engineering from the Basel Institute of Technology (Switzerland) in 1984. He teaches Simulation and Modelling, does research and consults in the area of Enterprise Simulation. He is the project leader of the nationwide research and development project "Muttenzer Modellunternehmung" which includes MASTER. His research interests are in modelling and simulation of enterprise systems, cost-optimisation and simulation systems/-environments. He is a member of SCS, ASIM and SGL.

MARCO C. BETTONI is a Professor in the CIM Center at the Basel Institute of Technology. He received an M.S. degree in mechanical engineering from the Swiss Federal Institute of Technology (ETH Zürich) in 1977. He teaches Artificial Intelligence, especially the industrial application of Knowledge Based Systems. His main research interests focus on the application of the expert system technology to graphical user interfaces, simulation optimisation and program synthesis from natural language descriptions as well as on the fundamental questions of natural intelligence.