Proceedings of the 1987 Winter Simulation Conference
A. Thesen, H. Grant, W. David Kelton (eds.)

AN INTELLIGENT MODELING SYSTEM FOR SIMULATING MANUFACTURING PROCESSES

Donnie R. Ford
Kenneth E. Johnson Research Center
University of Alabama in Huntsville
Huntsville, Alabama 35899

Bernard J. Schroer
Kenneth E. Johnson Research Center
University of Alabama in Huntsville
Huntsville, Alabama 35899

Rodney Daughtrey
Kenneth E. Johnson Research Center
University of Alabama in Huntsville
Huntsville, Alabama 35899

ABSTRACT

This paper presents the development of an intelligent system for simulating manufacturing processes. The emphasis of this paper is placed on the following elements of the intelligent system: the natural language interface, especially the parser, the simulation analyzer, and the simulation writer. An application of the intelligent manufacturing simulation system in an industrial environment is also presented.

1. INTRODUCTION

An elusive goal of the simulationist has been to communicate directly with the computer and then have the computer generate the appropriate simulation model. Furthermore, through additional interfaces the model would then perform various "what-if" scenarios and the corresponding statistical analyses. Many simulationists have found this goal nothing more than a dream. Recent developments, however, have moved this goal closer to reality.

One not-so-recent development has been the introduction of simulation languages such as GPSS (Gordon 1975), SLAM (Prisker and Pegden 1986) and SIMAN (Pegden 1985). These languages are relatively easy to use. Recently, many of these languages have been programmed for the microcomputer, such as GPSS/PC by Minuteman (1986) and SIMAN by Systems Modeling Corporation. Also, these languages have been expanded to provide graphical animation capabilities, such as SIMAN's CINEMA. The languages make simulation easier to use especially by decision makers. For example, these languages have enhanced the management and analysis aspects of simulation modeling. Notwithstanding, the simulationist still must construct the simulation model.

Another development has been the introduction of interactive software that translates the logic of a model, described in relatively general symbolism, into the corresponding simulation language code. A recent article by Haddock (1987) discussed some of these developments which he called simulation program generators. For example Medeiros and Sadowski (1983) developed a system for constructing simulation models by selecting various generators from a library of available programs. Then by linking these programs generators together and executing the software, the system automatically generates the simulation code.

In addition to these developments, the excitement generated by Artificial Intelligence, or AI, as a simulation assist has resulted in numerous research activities in intelligent simulation systems. In an ideal intelligent simulation system, the system is able to interpret and understand the user's request and then determine what is needed in terms of data input, techniques to process the information and the type of information to be output (Shannon 1985). With the recent development of the new AI processors and the supporting AI languages and development toolkits, it is now conceivable to begin visualizing intelligent simulation systems. For example, Intellicorp's SIMKIT (1986) and Carnegie Group's SIMULATIONCRAFT (1986) are two recently introduced commercial AI toolkits for use in manufacturing simulations.

This paper presents the development of an intelligent system coupled with a commercial simulation language for simulating manufacturing processes. The research goal is to provide a natural language interface so the decision maker will not have to learn the simulation language. A second research goal is to embed an expert system in the system that will assist the decision maker in analyzing the model output.

2. INTELLIGENT MANUFACTURING SIMULATION SYSTEM

The elements of the Intelligent Manufacturing Simulation System (IMSS) are:

- o Natural language interface
  - Transformer
  - Understander
- o Simulation writer
- o Simulation analyzer

The interfacing of these elements are described pictorially in Figure 1.

2.1 Natural Language Interface

A major element of the Natural Language Interface (NLI) (Ford and Schroer 1987) is the parser. In the Intelligent Manufacturing Simulation System, the parser is divided into a Transformer and an Understander. In general, a parser converts input into basic structures according to prescribed rules. These basic structures may be grammatical or semantical and represent the basic concepts of the language. The prescribed rules directly impact the robustness of the parser. If there are any rules missing, the conversion of the input could be incomplete or a failure of the parser could occur. This problem becomes more manageable when the domain of the parser is narrowly defined. In the case of the IMSS, the domain is very specific: an electronics manufacturing facility in Huntsville, Alabama.

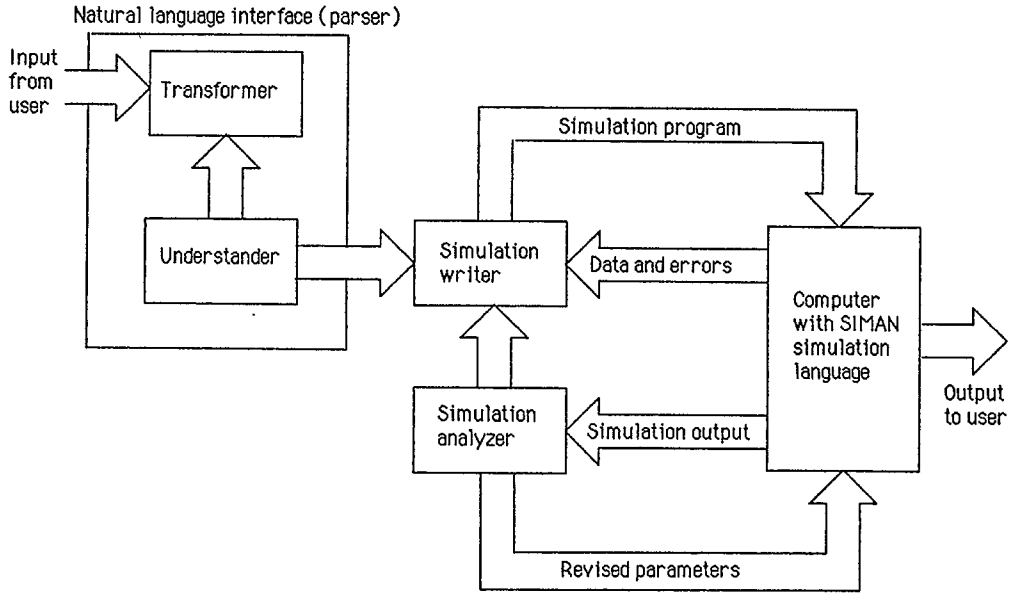

Figure 1. Intelligent manufacturing simulation system

Transformer. The transformer converts the natural language input into an internal representation. This internal representation is called conceptual dependency and was developed from a theory of language understanding based on semantics rather than syntax (Schank 1975). Because of its structure and rules, inferences can be made based on the input that allow a more complete understanding of the meaning of the input. Also, this allows the Transformer to function similar to a human in processing input. The input is processed according to the algorithm developed by Riesbeck (1974).

During this conversion, if any errors are encountered during the interpretation phase, the system will restart the process taking note of the procedures that have been used previously. Thus, backtracking to a previous decision point is not used. Instead the system utilizes a memory system for what has been used thus far in the conversion and begins from the start of the sentence (Bobrow 1977). This simulates more closely the behavior exhibited by humans during the process of natural language understanding.

The Transformer is designed and developed with emphasis on the kinds of concepts people use in dealing with other people; i.e., beliefs, communication, intention, reasons, and results. The process of converting natural language into conceptual dependency allows the Transformer to find an interpretation of the sentence input. The system looks for the ideas that the user is trying to communicate. Therefore, decisions about concepts and conceptual relationships are made while the sentence is being read.

Understander. Once the entire story has been converted into conceptual dependency format and the Transformer has made an interpretation, control is passed to the Understander. The Understander's function is to make all the necessary inferences to complete the meaning of the story and to prepare the necessary input for the Simulation Writer. In completing its objectives, the Understander uses rule-based knowledge about conceptual dependency, goals, plans, themes, and their relations. The Understander uses the algorithm developed by Wilensky (1980).

While performing this process on the story, the Understander uses both forward and backward chaining. The Understander initially starts with backward chaining. In using this combination of control procedures, the system is able to control the number of inferences that are generated. Also, this more closely resembles the way humans perform the same process. These inferences are used to establish the connection between events in the story. The events themselves are identified by explanations from the algorithmic process. Therefore, the story understanding is explanation-driven.

The utility of this model is in providing a way of organizing knowledge about actions so that connections between events can be inferred that are not "canned" in the program's memory structure (Winograd 1974). This organization makes it easy to integrate new knowloedge into the system and to interact with previous knowledge.

After all the sentences are processed through the explanation algorithm and the inferences are made, then the Understander is said to understand

the story. That is, the Understander knows what the user desires to be explained or in our case simulated. Next, the Understander begins preparing the input for the Simulation Writer. These commands are passed one at a time to the Simulation Writer.

2.2 Simulation Writer

The Simulation Writer (SW) is basically a code generator. The SW takes the commands from the Understander, converts them to SIMAN code, writes the code to a file, and sends the code to the computer for execution. When the Simulation Writer receives the input, the code is first checked to see if there is any previously generated code that can be used to solve the current problem. If there is, the code is removed; modified, if necessary; and used for the solution. New code may or may not be generated depending on the magnitude of the modifications that need to be made to the existing library code. If there is not any code that can be used, the SW begins the process of building the needed SIMAN code.

2.3 Simulation Analyzer

The Simulation Analyzer (SA) is a rule-based expert system and contains knowledge on improving system efficiency by manipulating utilizations and system design. Any recommendations are passed to the Simulation Writer which modifies the existing code and re-runs the simulation. The user is asked for input prior to sending the corrections to the Simulation Writer. The user can change, modify, or accept the proposed corrections. Also, the user can perform "what-if" scenarios. Once the user is satisfied or there are no further efficiency improvements, the system stops processing the current problem and prepares for a new problem. The Simulation Analyzer is in the conceptual stage of development.

3. SIMULATION APPLICATION

The Intelligent Manufacturing Simulation System is being developed in LISP on a Symbolic 3670 which is interfaced to a VAX 785. The IMSS is being developed domain specific for an electronics manufacturing plant. Furthermore the dictionary of words and phrases are plant specific for the Motorola/UDS facility in Huntsville, Alabama. This dictionary contains over 100 words and phrases and is being added to constantly as the IMSS development continues. Figure 2 contains a representative list of these words and phrases.

3.1 Experimental Results

A person very knowledgeable in simulation and familiar with the Motorola/UDS manufacturing plant was asked to describe four simple simulation scenes and to write these descriptions in text form. Prior to writing the text, the person was briefed by the IMSS developer on the words and expressions in the parser's dictionary. Each of these examples illustrates the use of the IMSS in generating SIMAN code directly from text. These examples will also indicate some of the current deficiencies in the IMSS and areas for further refinement and development.

```
(enter      def (ptrans actor nil
                        object *
                        from * ))

(process    def (do actor *
                    object * ))

(second     def (time name (minute)
                      base-units (1)))

(minute     def (time name (minute)
                      base-units (60)))

(uniform    def (dist-type name (uniform)
                           min *
                           max * ))

(mean       def (statistic name (mean)
                           measure * ))

(box        def (phys-obj class (container)
                          name (box)))

(to         def (prep is (to)))

(is         def (be-verb name (is)))

((ai work center)
            def (complex name (automatic-insertion)))

((dip machine)
            def (process-actor class (station)
                               name (dip-machine)))

((normal distribution)
            def (dist-type name (normal)
                           mean *
                           sd * ))

((printed circuit board)
            def (process-object name (printed-circuit-board)))
```

Figure 2. Partial list of words and phrases

Experiment One. The first scene describing a relatively simple process was:

> PCBs arrive at the automatic insertion center according to a uniform distribution with a minimum of 8 seconds and a maximum of 12 seconds. PCBs are processed at the automatic insertion center according to a uniform distribution with a minimum of 9 seconds and a maximum of 11 seconds. The PCBs then exit the AI center.

The output from the parser for this scene ise given in Figure 3 and serves as input to the Simulation Writer. The output from the Simulation Writer is the following SIMAN code:

```
BEGIN;
        CREATE,1:UN(1,1);
        DELAY;
        QUEUE,1,;
        SEIZE:RES1;
        DELAY:UN(2,2);
        RELEASE:RES1;
END;
```

```
(PTRANS ACTOR NIL
        OBJECT (PROCESS-OBJECT NAME (PRINTED-CIRCUIT-BOARD)
                               QUANTITY (1))
        TO (COMPLEX NAME (AUTOMATIC-INSERTION)
                    PREPOBJ (PREP IS (AT)))
        VOICE (ACTIVE)
        SENT-NUM (1)
        DIST (DIST-TYPE NAME (UNIFORM)
                        MIN (STATISTIC NAME (MIN)
                                       MEASURE (TIME NAME (SECOND)
                                                      BASE-UNITS (1)
                                                      QUANTITY (8)))
                        MAX (STATISTIC NAME (MAX)
                                       MEASURE (TIME NAME (SECOND)
                                                      BASE-UNITS (1)
                                                      QUANTITY (12)))))

(DO ACTOR (COMPLEX NAME (AUTOMATIC-INSERTION)
                   PREPOBJ (PREP IS (AT)))
    OBJECT (PROCESS-OBJECT NAME (PRINTED-CIRCUIT-BOARD)
                           QUANTITY (1))
    VOICE (PASSIVE)
    SENT-NUM (2)
    DIST (DIST-TYPE NAME (UNIFORM)
                    MIN (STATISTIC NAME (MIN)
                                   MEASURE (TIME NAME (SECOND)
                                                 BASE-UNITS (1)
                                                 QUANTITY (9)))
                    MAX (STATISTIC NAME (MAX)
                                   MEASURE (TIME NAME (SECOND)
                                                 BASE-UNITS (1)
                                                 QUANTITY (11)))))

(PTRANS ACTOR NIL
        OBJECT (PROCESS-OBJECT NAME (PRINTED-CIRCUIT-BOARD)
                               QUANTITY (1))
        FROM (COMPLEX NAME (AUTOMATIC-INSERTION))
        VOICE (ACTIVE)
        SENT-NUM (3))
```

Figure 3. Output from the parser

Experiment Two. The second scene is similar to the first scene. The following second paragraph was added:

> The PCB's then arrive at the dip machine. At the dip machine PCB's are processed according to a normal distribution with a mean of 5 seconds and a standard deviation of 1 second.

The parser had difficulty with the prepositional phrase " at the dip machine" at the start of the second sentence. This sentence was rewritten as:

> The PCBs are processed at the dip machines according to a normal distribution....

By making this change the IMSS was successful in generating the following correct SIMAN code:

```
BEGIN;
        CREATE,1:UN(1,1);
        DELAY;
        QUEUE,1,;
        SEIZE:RES1,1;
        DELAY:RN(2,2);
        RELEASE:RES1,1;
END;
```

Experiment Three. The third scene was:

> PCBs arrive at the automatic insertion center at a rate of 10 per minute. The PCBs are processed at the dip machine at a rate of 75 per minute. The PCBs are processed at the VCD machine at a rate of 6 per minute. The PCBs then exit the AI center.

The system could not interpret the word "rate." These sentences had to be changed to:

> PCBs arrive at the automatic insertion center according to a uniform distribution with a minimum of 6 seconds and a maximum of 6 seconds. The PCBs are processed at the dip machine according to a uniform distribution with a minimum of 8 seconds and a maximum of 8 seconds. The PCBs are processed at the VCD machine according to a uniform distribution with a minimum of 10 seconds and a maximum of 10 seconds. The PCBs then exit the AI center.

The correct SIMAN code after making this change is:

```
BEGIN;
        CREATE,1:UN(1,1);
        QUEUE,1;
        SEIZE:RES1,1;
        DELAY:UN(2,2);
        RELEASE:RES1,1;
        QUEUE,2,;
        SEIZE:RES2,1;
        DELAY:UN(3,3);
        RELEASE:RES2,1;
END;
```

Experiment Four. The fourth scene was a more complex problem, although still relatively simple. The problem included some elementary branching based on a part reject rate. The following paragraph was added to the third scene:

> The PCBs then proceed to a quality control station where the PCB's are processed (inspected) at a rate of 20 per minute. Ten percent of the PCBs are rejected. Accepted PCBs proceed to the test and assembly center. The rejected PCBs then proceed to a repair station where they are repaired at a rate of 5 per minute. The repaired PCBs then proceed to the test and assembly center. The PCBs are processed at the test and assembly center at a rate of 10 per minute. Ten percent of the PCBs are rejected at the test and assembly center. The rejected PCBs proceed to scrap. The accepted PCBs exit the AI center.

The parser presently cannot process branching logic. This capability is currently being added to IMSS.

4. CONCLUSIONS

The IMSS development began in January 1986 by a research grant from UDS Corporation in Huntsville, Alabama. UDS is a division of Motorola and a manufacturer of modems and related electronics equipment. In summary, progress in developing a robust system capable of being transported to UDS for implementation has been slow. However, progress has been made.

The following conclusions can be made to date in the development of the IMSS system:

- o Development of a relatively large system such as the IMSS dictates the requirement for a symbolics processor rather than a PC based system.

- o A relatively complex system such as the IMSS has to be developed using an AI language such as LISP rather than a commercial AI development tool kit. That is, the requirements of a real world application generally requires functions and procedures to be developed outside a commercial tool kit domain.

- o The organizational structure of the SIMAN simulation language is ideal for coupling with an expert system.

REFERENCES

Bobrow, D. et al, "GUS, A Frame-Driven Dialog System, Artificial Intelligence, 1977.

Bullers, W. "Artificial Intelligence in Manufacturing Planning and Control," AIIE Transactions, December 1980.

Ford, D. R. and B. J. Schroer,"An Expert Manufacturing Simulation System," Simulation, Vol. 48, Nr. 5, May 1987.

Fox, M. A., The Intelligent Management System: An Overview, Carnegie-Mellon University Report CMU-RI-TR-81-4, December 1982.

Gordon, G., The Application of GPSSV to Discrete System Simulation, Prentice Hall Inc., Englewood Cliffs, NJ, 1975.

Lebnert, W. and M. Burstein, The Role of Object Primitives in Natural Language Processing, Tech Rep. 162, Yale University, 1979.

Lebowitz, M., Generalization and Memory in an Integrated System, Tech Rep. 187, Yale University 1980.

Lebowitz, M., "RESEARCHER: An Overview," AAAI-83 Proceedings, Washington, 1983.

Lo, H. and J. Sheeler, "A Computer Simulation Methodology by LISP and SANS," Proceedings Summer Computer Simulation Conference, Vol I, 1983

Medeiros, D. J. and R. P. Sadowski, "Simulation of Robotic Manufacturing Cells: A Modular Approach," Simulation, Vol 40, Nr 1, pp. 3-12, 1983.

Miller, R., "Artificial Intelligence: A New Tool for Manufacturing," Manufacturing Engineering, April, 1985.

Pegden, C. Dennis, Introduction to SIMAN, Systems Modeling Corp., 1985.

Pritsker, A.A.B. and C. D. Pegden, Introducction to Simulation and SLAM, Wiley and Sons, New York, NY, 1986.

Riesbeck, C., Computational Understanding: Analysis of Sentences and Context, PhD Thesis, Stanford University, 1974.

Schank, R., Conceptual Information Processing, North Holland Press, Amsterdam, 1975.

Shannon, R., R. Mayer and H. Adelberger, "Expert Systems and Simulation," Simulation, June 1985.

Stefic, M., "The Organization of Expert Systems: A Perspective Tutorial," Artificial Intelligence, March 1982.

Sussman, G., A Computer Model of Skill Acquisition, American Elsevier Publishing.

Wilensky, R., Understanding Goal-Based Stories, Garland Publishing, NY, 1980.

Winograd, J., Representation and Understanding: Studies in Cognitive Science, Academic Press, New York, 1975.

Winograd, J., Understanding Natural Language, Academic Press, New York, 1972.

Winston, P. H., Artificial Intelligence, Addison-Wesley Publishing Inc., 1984.

Ziegler, B. P., Multifaceted Modeling Methodology and Discrete Event Simulation, Academic Press, London, 1984.

_____ Simkit System Knowledge Based Simulation Tools in KEE, No. 1.0-USK-3, Intellicorp, Inc., February 1986.

_____ GPSS/PC Reference Manual, Minuteman Software, Stow, MA, 1986.

_____ Simulationcraft, Carnegie Group Ind., Cambridge, MA, 1987.

AUTHORS' BIOGRAPHIES

DONNIE R. FORD is a research instructor in the Department of Management Science at the University of Alabama in Huntsville. He is currently completing his PhD in business at the University of Alabama. Mr. Ford is a member of AAAI.

BERNARD J. SCHROER is director of the Johnson Research Center and a research professor in the Department of Management Science at the University of Alabama in Huntsville. He has a PhD in industrial engineering from Oklahoma State University and is a registered professional engineer. Dr. Schroer is a member of IIE, SME/RI, SCS, AAAI, NSPE, and Sigma Xi.

RODNEY DAUGHTREY is a research associate in the Johnson Research Center. He has a BS in computer engineering from Vanderbilt and is pursuing a masters in computer science at the University of Alabama in Huntsville.