NISTIR 6154

Production Systems Engineering: Requirements Analysis for Discrete-Event Simulation

Anna Bartolotta
Charles McLean
Y. Tina Lee
Albert Jones

U.S. DEPARTMENT OF COMMERCE
Technology Administration
Manufacturing Systems Engineering Group
Manufacturing Systems Integration Division
Manufacturing Engineering Laboratory
National Institute of Standards
and Technology
Gaithersburg, MD 20899-0001

NIST

NISTIR 6154

Production Systems Engineering: Requirements Analysis for Discrete-Event Simulation

Anna Bartolotta
Charles McLean
Y. Tina Lee
Albert Jones

U.S. DEPARTMENT OF COMMERCE
Technology Administration
Manufacturing Systems Engineering Group
Manufacturing Systems Integration Division
Manufacturing Engineering Laboratory
National Institute of Standards
and Technology
Gaithersburg, MD 20899-0001

April 1998

U.S. DEPARTMENT OF COMMERCE
William M. Daley, Secretary

TECHNOLOGY ADMINISTRATION
Gary R. Bachula, Acting Under Secretary
for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Director

Preface

The Computing, Information, and Communications (CIC) Program [3] was formally established by the High Performance Computing Act of 1991 (Public Law 102-194). The goal of this program is to accelerate the development of future generations of high performance computers and networks and the use of these resources in the government and throughout the U.S. economy. The National Institute of Standards and Technology's Systems Integration of Manufacturing Applications (SIMA) program [2] is the agency's coordinating focus for its CIC activities. SIMA is addressing the information interface needs of the U.S. manufacturing community. Specifically, the SIMA program works with U.S. industry to:

- develop information exchange and interface protocols to address manufacturing integration problems,
- establish test mechanisms for validating protocols and implementations, and
- transfer information technology solutions to manufacturing enterprises.

The primary output of the SIMA Program will be a collection of specifications called Initial Manufacturing Exchange Specifications (IMES) [4]. IMESs provide the means to improve the SIMA Program's ability to meet the needs of the U.S. industry in the area of standards and testing methods by providing a structured approach to the SIMA Program's activities in this arena. They will fill an important void in the manufacturing systems integration process as it exists today. Each IMES will be developed through an industry review and consensus process. It is expected that the manufacturing community will accept them as an authoritative specification.

Three types of IMESs have been identified: an interface specification between a human being and a software application; an interface specification between two or more software applications; and a reference information repository specification. Each IMES involves several components that define the integration aspect, specify a definitive solution to the integration problem, and demonstrate the validity of the proposed solution. It must contain a clear description of WHAT information the interface or repository MUST convey, and possibly HOW it is conveyed. The content is usually specified by an information model of all the objects and related information attributes that are covered by the specification.

To support the scope and domain specifications, the IMES shall address a particular "example scenario," identifying an actual interface/information requirement derived from a real industrial problem. The proof of IMES' value to industry will be the ability to build a prototype to the IMES, using the software applications actually used by the industrial practitioners, and solving the cited problem. To support the development of an IMES, SIMA projects will have seven phases: identify/define the industry need, conduct requirements analysis, develop proposed solution, validate proposed solution, build consensus, transfer technology, and initiate standardization. Each of these phases has a well-defined set of deliverables.

This document follows the Phase I IMES document of the Production Systems Engineering component of the Production and Product Data Management Project within SIMA [6], that identified and documented the industry need, technical specifications to be developed, potential collaborators, a proposed technical approach, and a manufacturing scenario for this project. It also described the relationships between the proposed project, the SIMA Reference Architecture, other related projects, and current standards activities.

This document describes the results of the requirement analysis phase (phase II, according to [4]), for a particular interface specification, the interface specification for discrete event simulation. This Phase II IMES document defines the context, scope and information requirements for discrete-event simulation models of manufacturing systems.

Work described in this paper was sponsored by the NIST Systems Integration for Manufacturing Applications (SIMA) and U.S. Navy Manufacturing Technology Program.

Table of contents

1. SCOPE **1**

1.1 APPLICATION ACTIVITY MODEL (AAM) 4

1.1.1 AAM definitions and abbreviations *4*

1.1.2 AAM Diagrams *8*

2. STANDARDS REVIEW **12**

2.1 STANDARDS REVIEW 12

3. DEFINITIONS AND ABBREVIATIONS **13**

3.1 TERMS DEFINED IN ISO 10303-1 13

3.2 TERMS DEFINED IN 10303-44 13

3.3 TERMS DEFINED IN 10303-203 14

3.4 TERMS DEFINED IN 10303-213 14

3.5 OTHER DEFINITIONS 14

3.6 ABBREVIATIONS 14

4. INFORMATION REQUIREMENTS **15**

4.1 UNITS OF FUNCTIONALITY (UOF) 15

4.1.1 logical elements UoF *15*

4.1.2 planning UoF *15*

4.1.3 product UoF *16*

4.1.4 process plan UoF *16*

4.1.5 resource UoF *16*

4.2 APPLICATION OBJECTS 17

4.2.1 Arrival *17*

4.2.2 Attribute *18*

4.2.3 Breakdown *19*

4.2.4 Connection *20*

4.2.5 Conveyor *20*

4.2.6 Generic resource *22*

4.2.7 Load *23*

4.2.8 Location *24*

4.2.9 Lookup table *24*

4.2.10 Maintenance *25*

4.2.11 Operation *26*

4.2.12 Operator *29*

4.2.13 Part *30*

4.2.14 Path *31*

4.2.15 Process plan *32*

4.2.16 Processor *32*

4.2.17 Queue *33*

4.2.18 Resource *34*

4.2.19 Shift *35*

4.2.20 Unit load *36*

4.2.21 Variable *36*

4.2.22 Vehicle *37*

4.2.23 Work schedule ... *39*
4.3 APPLICATION ASSERTIONS ... 39
4.3.1 Arrival to Load ... *40*
4.3.2 Connection to Resource ... *40*
4.3.3 Load to Attribute ... *40*
4.3.4 Load to Process plan ... *40*
4.3.5 Load to Part ... *40*
4.3.6 Operation to Process plan ... *40*
4.3.7 Operation to Resource ... *40*
4.3.8 Resource to Breakdown ... *40*
4.3.9 Resource to Maintenance ... *40*
4.3.10 Resource to Shift ... *41*
4.3.11 Resource to Work schedule ... *41*
5. REFERENCES ... 42

1. SCOPE

Production System Engineering is one of the focus areas within the Systems Integration of Manufacturing Applications (SIMA) Production and Product Data Management project [2]. The project focuses on providing the models, integrated framework, operating environment, common databases, and interface standards for a wide variety of emerging tools and techniques for designing manufacturing processes, equipment, and enterprises. The Phase I IMES document [6] identified and documented the industry need, technical specifications to be developed, potential collaborators, a proposed technical approach, and a manufacturing scenario for the Production System Engineering project.

This Phase II IMES document specifies an interface specification for the exchange, archiving, and sharing of discrete-event simulation models of manufacturing systems. The objective is to provide a neutral mechanism capable of describing input data to a discrete-event simulation system for manufacturing systems, independent from any particular commercial simulation system. The nature of this description makes it suitable not only for neutral file exchange between dissimilar discrete-event simulation systems, but also as a basis for implementing and sharing databases and archiving. In fact the existence of this interface specification for input data to discrete-event simulation enables the automatic transmission of data from existing knowledge-based systems: information on resources, products, and process can be either extracted from existing repository systems through links or imported from external sources.

The interface specification should provide the ability to exchange simulation models of production systems from one enterprise to another by utilizing data exchange standards. This could be particularly suitable for emerging virtual enterprises conducting collaborative design and manufacturing. It could be used to simulate and prove out the manufacturing cycle of a product prior to launching production ramp-up.

Simulation software has been used to model different processes and systems within the manufacturing enterprise for many years. The most important concept in manufacturing simulation is that of a state variable. State variables describe what is happening in the process or system at any point in time. Continuous simulation models are used for state variables that change continuously over time. The models are mainly mathematical, differential, or difference equations that represent the evolution of some physical phenomena which changes continuously over time. They are used primarily during product design and process selection. Examples include fluid and structural dynamics, stress analysis, heat transfer, and machine tool program verification.

In the case of discrete-event simulation, state variables change at discrete points in time. Examples of such variables include the number of jobs waiting in the queue in front of a machine, the status of each machine on the shop floor, and the location of each job in the factory. The simulation models, which are built using these methodologies, are mainly flow models that track the flow of entities through the factory. The tracking is done using times at which the various events occur. This interface specification deals only with discrete-event simulation. Furthermore, the viewpoint in this interface specification is that of a production system designer responsible for plant design. He must ensure that there is sufficient production capability and capacity to manufacture products. He must decide on which material processing machines, storage devices, and transportation systems to buy, and the proper physical layout. The long-run performance of production systems hinges critically on these policy decisions. Intuition is frequently a poor guide because of the complexity and stochastic nature of the interactions among these various components. Furthermore, there are usually many performance objectives, some of which can be negatively correlated. Minimizing the total system cost and maximizing system flexibility are examples of negatively correlated objectives, because flexible, highly reliable equipment is very expensive. As a result, analysis tools such as simulation are critical to capturing the trade-offs and helping the system designer make decisions. The simulation models that are used to make these types of decisions generally represent the flow of materials to and from processing machines and the operations of the machines themselves. The critical issues that are assessed include throughput, location of bottlenecks, machine reliability, and cost.

Figure 1 contains the data planning model that provides a high-level description of the requirements for this interface specification, as well as the relationships between the basic data components. The data planning model shows the *resource* class; all the plant components (processor, storage, and transporters) and the auxiliary resources (such as unit load, tool, and fixture) within a manufacturing system are subclasses of the *resource* class. Four elements are associated with the *resource* class: *shifts*, *breakdown*, *maintenance,* and *work schedule.* These new classes model the unavailability of the resources due to scheduled events (maintenance, work schedule, and shifts) or unscheduled events (like breakdowns). The plant components (a subset of the resources) are connected through the *connection* class. The *connection* class enables us to represent a physical link between two plant components (for example, a queue and a processor). In the top part of the picture, the *arrival* class allows the simulation of the behavior of a demand. The *arrival* class periodically generates new loads and releases them in the system. The *load* represents the real physical entity that flows within the manufacturing system. Generally, it is composed of parts of different types and it travels on a *unit load* (which represents the physical means employed for transporting the parts among the different resources). A *process plan* is associated to each load. A *process plan* is composed of a sequence of single steps, called *operations.* The *operation* class permits us to allocate the *resources* to the *loads* flowing within the system.

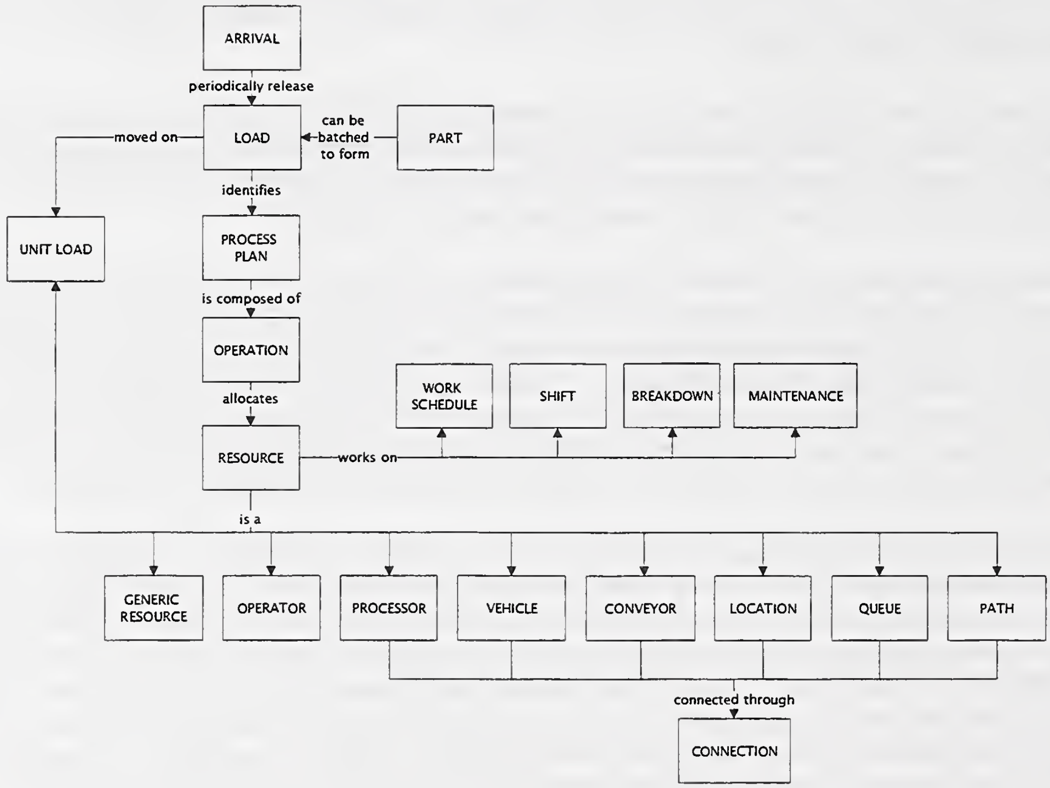

Figure 1 – Data planning model

The following are within the scope of this IMES:
- information required to describe the physical resources in a discrete-event simulation model;
- information required to describe the process plan in a discrete-event simulation model;
- information required to describe the flow of the parts inside the manufacturing system.

The following are outside the scope of this IMES:

- modeling of the production planning and control system;
- representation of alternative process plans;
- continuous production process;
- output data from the discrete-event simulation system;
- layout information of the facility;
- information modeling of area-restricted transporters, like cranes and robots;
- modeling of a complex storage system, like an automated storage and retrieval system (ASRS);
- information required to animate the simulation model.

A typical interface specification contains: 1) scope and domain of the specification intended to serve, 2) application objects required for the application domain, 3) application information model that specifies all information necessary to represent the application domain, and 4) conformance requirements of the specification. This Phase II IMES document deals with the first two items.

The scope of this interface specification will be better clarified in the remainder of section 1: an application activity model is provided in the next section as basis for the definition of the scope. Sections 2 and 3 of this document present standards and definitions related to the designation of these requirements. Finally the information requirements of the interface specification, decomposed into units of functionality, application objects, and application assertions, are described in detail in section 4.

1.1 Application Activity Model (AAM)

The Application Activity Model (AAM) is provided to aid the understanding of the scope and information requirements defined in this IMES. The model is presented as a set of definitions for the activities and data flow between these activities. These activity flows are represented as level diagrams using Integrated Definition Method IDEF0 modeling techniques and the Meta Software Design/IDEF ™ tool [7]. Activities/functions are represented by "boxes," data/objects such as inputs, controls, outputs, and mechanisms are represented by "arrows" and boxes and arrows are labeled. If an activity at one level can been decomposed into sub-activities a separate level is included.

As with any IDEF0 model, the application activity model is dependent of a particular viewpoint and purpose. The viewpoint of the AAM is that of the system engineer and production manager responsible for designing or re-designing a production system. The purpose of the AAM is to identify a process for using simulation tools to evaluate a proposed production system design, hence to aid the understanding of the context and scope of this IMES.

1.1.1 AAM definitions and abbreviations

The following terms are used in the application activity model. The activities and data types are explained in alphabetical order.

1.1.1.1 Budget reports (output: A0, A5)

Estimated cost data required to support the implementation of the designed production system. The budget report, for each project, usually specifies: labor, tooling, capital equipment, projected maintenance, information and control system, operations, training, licensing and inspection, construction, installation, material (components, consumables), overhead (utilities, labor multipliers, area usage), and rental costs.

1.1.1.2 Build and validate simulation model (activity A44)

Construct the production system model using the selected simulation tool(s). The model contains input parameters, workcenter layout and activities, transportation and storage devices, and methods for computing performance objectives. After the model has been created, it must be tested to ensure that it faithfully represents the real world system.

1.1.1.3 Completed model (output: A44, input: A45)

The final validated model to be used in the evaluation of the current production system design. The model captures the system layout, dynamics, inputs, procedures, and processes. It also includes the statistical design of experiments to be used in the evaluation.

1.1.1.4 Define the production engineering problem (activity A1)

Delimit the production engineering problem to be solved; identify the part mix for which the facility is to be designed and the physical plant or area to be used. Define the work elements to be undertaken and the (external) constraints on the (re)design. For a complex product/part mix, this activity may produce a hierarchy of production engineering problem definitions for which all of the A3 activities are separately or jointly performed.

1.1.1.5 Define the simulation modeling objectives (activity A41)

Determine the subset of the user-defined performance objectives that will be analyzed in the simulation model(s). Some examples include expected throughput, expected time in the system, and expected delays at the various workcenters. Values must be specified for each of these objectives.

1.1.1.6 Design the production system (activity A3)

Design the physical processing systems, material storage and delivery systems, automated control systems, and information management systems for the production facility. This includes selecting major equipment items, tooling, controllers, information systems, and networking equipment. It also includes developing the facility layout and specifying physical plant requirements.

1.1.1.7 Develop simulation model requirements (activity A42)

Specify model characteristics considered essential for successful completion of the analysis. Those characteristics include: the type statistical support; the kind of simulation model - e.g., discrete, continuous, process, ergonomic; the current and future computing environment; integration capabilities - file transfer, sockets, database, common object request broker architecture (CORBA); and animation needs.

1.1.1.8 Engineer the production system (activity A0)

Design new or modified production facilities for the manufacture of a particular collection of parts. A "facility" may be a plant, a shop, a line, a manufacturing cell, or a group of manufacturing cells. This activity encompasses both design-from-the-walls and re-engineering of all or part of such a facility to improve the production of certain products. It includes identification of parts, products, and processes for which the production system is to be tailored, identification of the equipment to be installed or replaced, (re)design of the floor layout, and development of an implementation for the (re)designed production system.

1.1.1.9 Engineering decisions (control: A0)

Those decisions which a human design engineer brings to bear on a design problem.

1.1.1.10 Engineering tool kit environment (resource: A0)

A set of software packages that provides an integrated set of functions and shares data to serve a common business purpose, e.g., manufacturing engineering.

1.1.1.11 Evaluate model (activity A45)

Develop the statistical experimental design for the analysis to determine the number of runs and appropriate methods for varying model parameters. Run the model, complete the statistical analysis, and compare the results of the analysis to the stated performance objectives.

1.1.1.12 Implementation report (output: A0, A5)

Project management data required to support the implementation of the designed production system. The implementation report specifies the phases, tasks, resources, and timing data (early/late start and end dates, estimated task duration, slack and float, and lead times).

1.1.1.13 Manufacturing resources (input: A0, A3, A4, A5, A41)

The major physical resources used for production activities: machinery, test systems, material handling devices and carriers, tooling, work holding devices, materials, and staff.

1.1.1.14 Model and evaluate the system (activity A4)

Using both simulation and actual performance measurements, analyze the dynamics of the proposed system. Test the facility design against the expected production demand for the selected part mix. Identify quality concerns, performance bottlenecks, etc., and feedback results to the process specification and production system design activities.

1.1.1.15 Modeling requirements (output: A42, input: A43)

A complete, ranked-ordered list of requirements. Alternatives for individual items should be specified where they are acceptable. For example, the user may want the ability to integrate with a database, but he will accept file transfer capability. Those items deemed absolutely essential, should be labeled.

1.1.1.16 Performance evaluation (output: A4, input: A3)

This is a report that details the quality of the current system design, as computed by the simulation model, relative to the current performance objectives. The report will say which objectives were met and which objectives were not met, and by how much. For example, one objective may be that average throughput be 20 per day. The simulation results estimate that the throughput is only 18 per day.

1.1.1.17 Performance objectives (input: A0, A4, A41)

The quantitative measures used to evaluate the quality of current the production system design.

1.1.1.18 Prepare management information (activity A5)

When a satisfactory design for the production system has been found, activities like development of project plans and schedules, preparation of budgets, establishment of configuration management controls, and generation of reports are performed.

1.1.1.19 Problem definition templates (output: A1: control: A2, A3, A4, A5, A41, A42, A43, A44)

Data entry forms that collect all the data gathered by the team of engineers during the definition of the production engineering problem. Critical data that must be identified to initiate the engineering process includes: product data and key product attributes, production system and engineering project type, manufacturing constraints and issues, critical milestone dates and schedules, expected or estimated costs, and manufacturing data for related products.

1.1.1.20 Process specifications (output: A0, A2; control: A3, A4, A41, A42, A43, A44)

The high-level engineering specifications for the manufacture of the product: the principal stock materials or components from which it is to be made, the major processes to be employed, and the principal equipment and/or human resources required to perform those processes. This represents ongoing interchanges among process and production engineering activities, with the consequence that the process definition is in varying degrees of detail at different times. In later stages, it will include routings, operation sequences, special processing notes, quality control specifications, process control specifications, process measurement specifications and process tracking/audit requirements.

1.1.1.21 Product specifications (input: A0, A1, A2, A3)

The product specifications include functional specifications, performance specifications, appearance specifications, and other engineering specifications for the product to be manufactured.

1.1.1.22 Production requirements (input: A0, A1, A2, A3)

Requirements and constraints that characterize the production system that is to be designed. Examples of critical data that must be identified to initiate the engineering process are: product and data key attributes, production systems and engineering project type, manufacturing constraints and issues, critical milestone dates and schedules, expected or estimated costs, and manufacturing data for related products.

1.1.1.23 Quality, time, and cost constraints (input: A0, A1, A2, A3, A4, A5, A41)

Limitation imposed by product (price) planning and other corporate decisions on acceptable costs, time, and quality for the production system that is to be designed.

1.1.1.24 Select appropriate simulation tools (activity A43)

Using the specified requirements, choose the best available commercial simulation tool(s).

1.1.1.25 Selected simulation tools (output: A43, input: A44)

A list of the acceptable simulation tool(s), vendor, cost, and summary of capabilities.

1.1.1.26 Simulation objectives (output: A41, input: A2, A43, A44, A45)

This is a subset of the overall system performance measures. These measures, which are usually stated as averages such as average throughput or average delay, are compared against the simulation results to determine the quality of the production system design.

1.1.1.27 Specify production and support processes (activity A2)

Develop a specification for the production and support processes required to manufacture the target part mix. This activity concentrates on identifying processes common to multiple parts in the mix or to other parts manufactured in the same facility. In addition, this activity specifies the necessary support operations for the production system – materials management, tooling preparation and management, material flows, equipment maintenance, and workspace requirements.

1.1.1.28 System specifications (output: A0, A3, A45; control: A4, A5, A41, A42, A43, A44)

Output of the production system engineering activity, ultimately captured in the form of layouts that define the logical and physical location of systems, their orientation, and the paths by which material and information flows between them.

1.1.2 AAM Diagrams

The application activity model is given in the following diagrams. The graphical form of the application activity model is presented in the IDEF0 activity-modeling format [7], and taken from the Phase I IMES document [6] and the SIMA Reference Architecture [1].

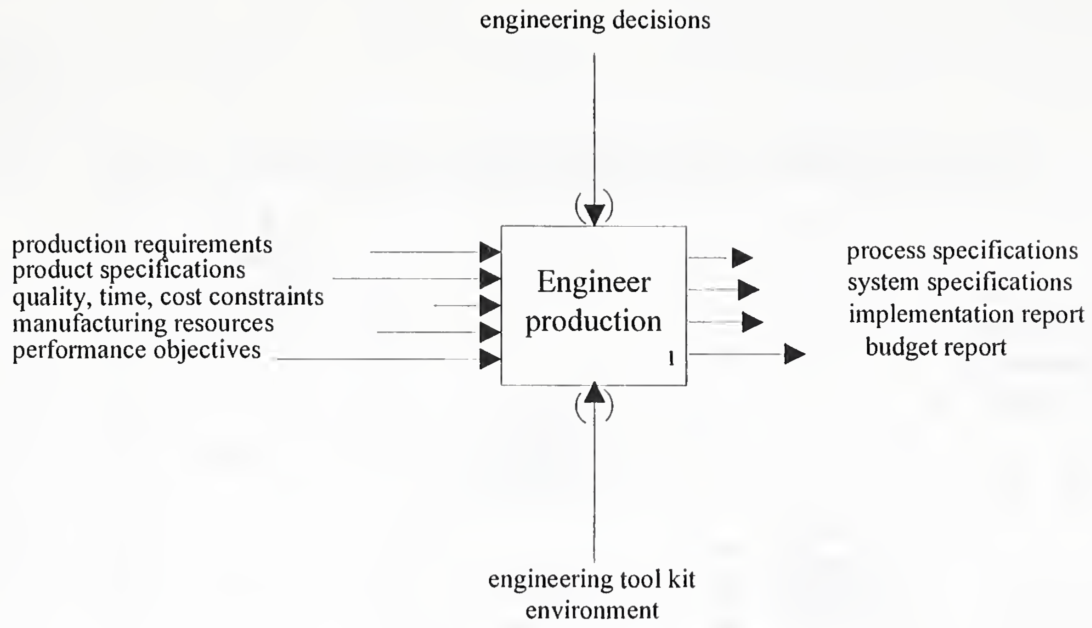

Figure 2- Top Level of the IDEF0 Model for Production System Engineering

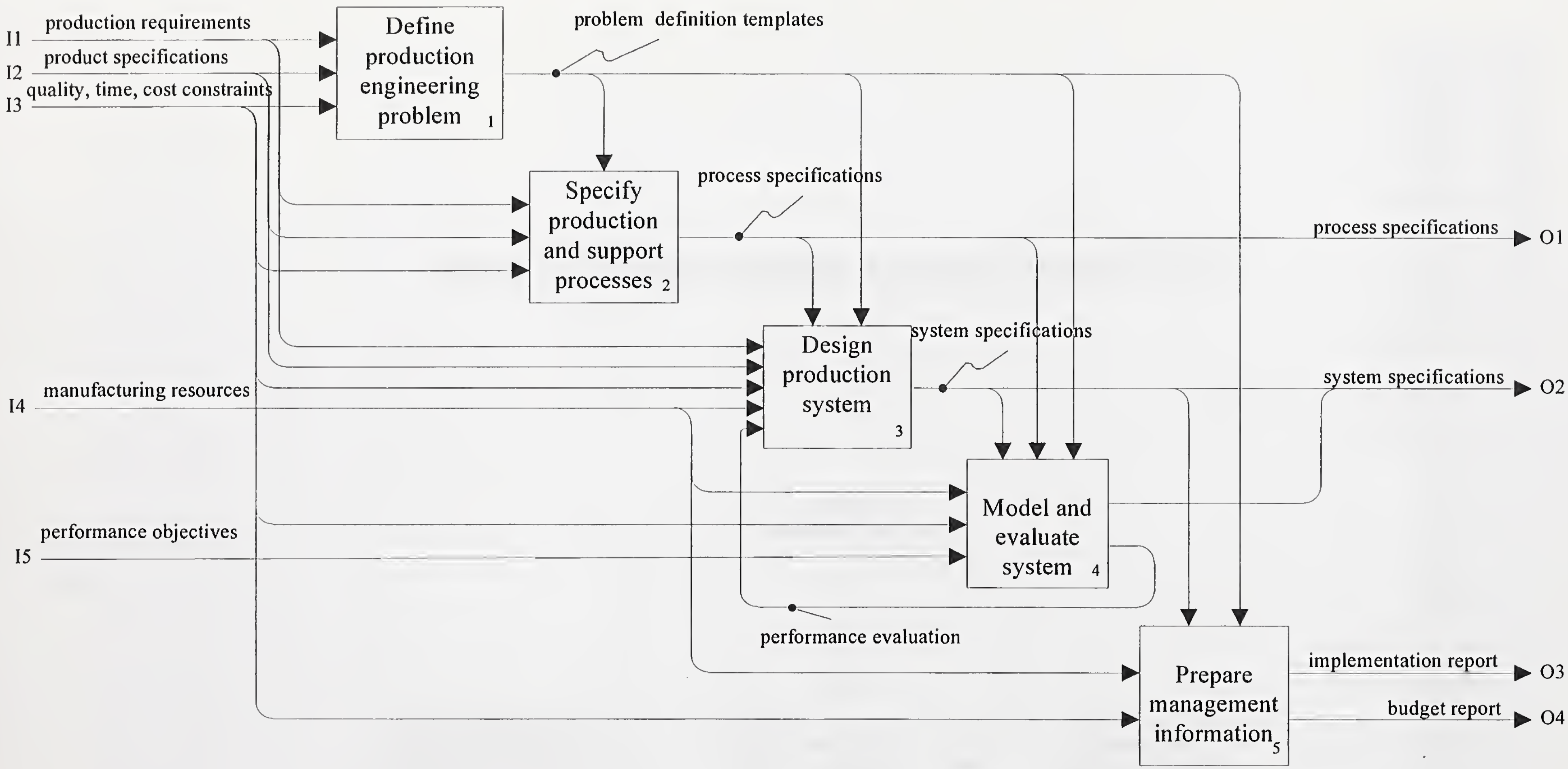

Figure 3 - Decomposition of the A0 Activity: Engineer Production System

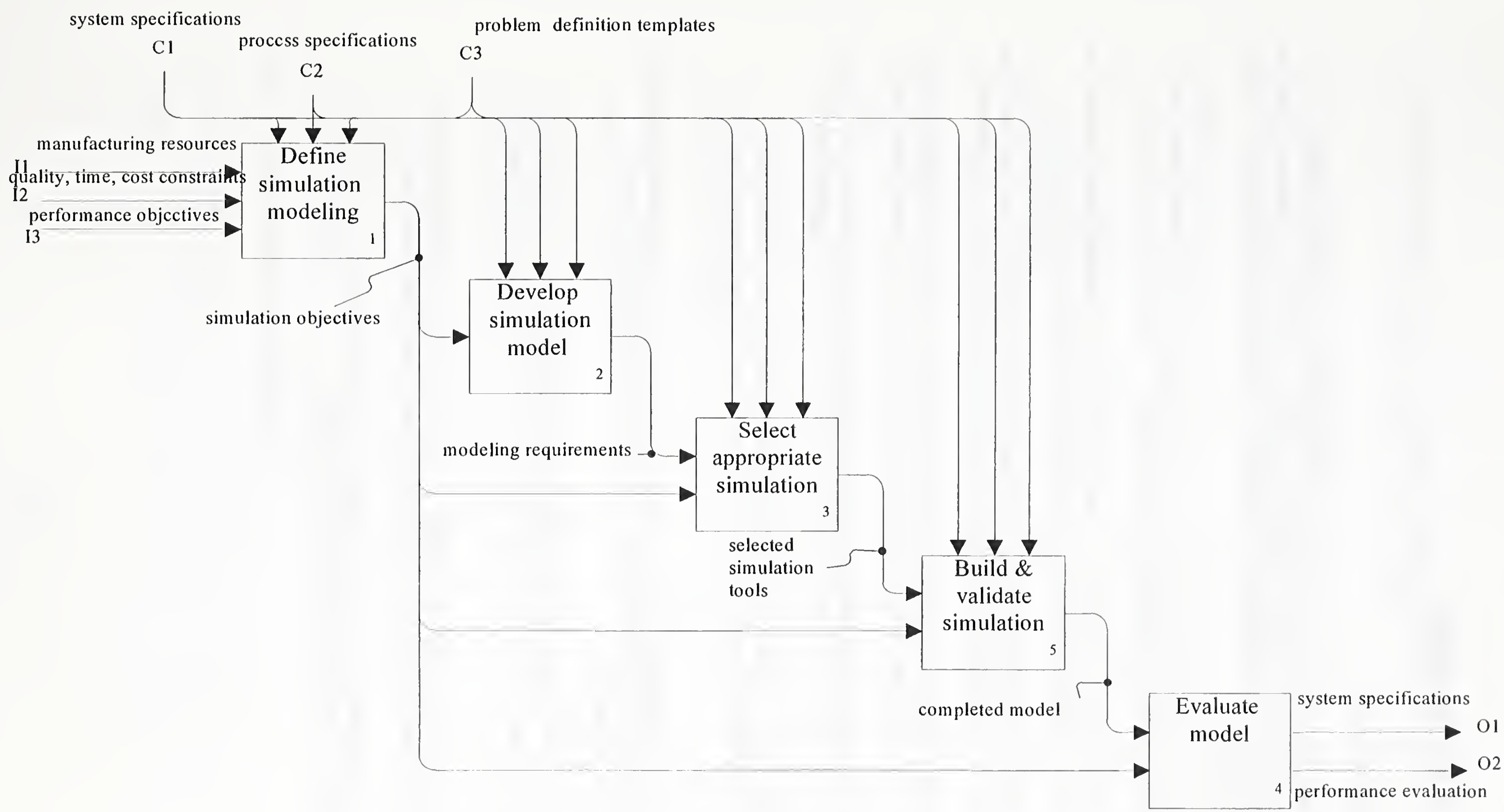

Figure 4 - Decomposition of the A4 Activity: Model and Evaluate System

2. STANDARDS REVIEW

2.1 Standards Review

The following standards contain provisions that, through reference in this text, constitute provisions of this IMES. All standards are subject to revision, and parties to agreements based on this IMES are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

ISO 10303-1: 1994, Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles.

ISO 10303-11: 1994, Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual.

ISO 10303-21: 1994, Industrial automation systems and integration - Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure.

ISO 10303-42: 1994, Industrial automation systems and integration - Product data representation and exchange - Part 42: Integrated generic resources: Geometrical and topological representation

ISO 10303-44: 1994, Industrial automation systems and integration - Product data representation and exchange - Part 42: Integrated generic resources: Product structure configuration.

ISO/DIS 10303-49: 1995, Industrial automation systems and integration - Product data representation and exchange - Part 49: Integrated generic resources: Process structure and properties.

ISO 10303-203: 1994, Industrial automation systems and integration - Product data representation and exchange - Part 203: Application protocol: Configuration controlled 3D designs of mechanical parts and assemblies.

ISO/DIS 10303-213: 1996, Industrial automation systems and integration - Product data representation and exchange - Part 213: Application protocol: Numerical process plans for machined parts.

ISO/CD 10303-227: 1995, Industrial automation systems and integration - Product data representation and exchange - Part 227: Application protocol: Plant spatial configuration.

ISO 8824-1: 1994, Information technology - Open systems interconnection - Abstract syntax notation one (ASN.1) - Part 1: Specification of basic notation.

3. DEFINITIONS AND ABBREVIATIONS

3.1 Terms Defined in ISO 10303-1

This IMES document makes use of the following terms defined in ISO 10303-1:

- application;
- application activity model;
- application context;
- application object;
- assembly;
- component;
- data;
- data exchange;
- exchange structure;
- implementation method;
- information;
- information model;
- product data;
- product information;
- resource construct;
- structure
- unit of functionality.

3.2 Terms Defined in 10303-44

- bill-of-material structure;
- link;
- tree.

3.3 Terms Defined in 10303-203

- process specification;
- sub-assembly.

3.4 Terms Defined in 10303-213

- activity;
- machine.

3.5 Other Definitions

For the purposes of this IMES, the following definition applies:

- discrete-event simulation: technique used for analyzing the performance of a system through the employment of a model that reproduces on computer the behavior of the real system.

3.6 Abbreviations

For the purposes of this IMES, the following abbreviations apply:

AAM	application activity model
AGV	automatic guided vehicle
BOM	bill-of-material
id	identifier
ICAM	integrated computer-aided manufacturing
IDEF0	ICAM definition language 0
IMES	Initial Manufacturing Exchange Specifications
ISO	International Organization for Standardization
NIST	National Institute of Standards and Technology
MHS	Material handling system
SIMA	System Integration of Manufacturing Applications Program

STEP	STandard for the Exchange of Product model data
UoF	Unit of Functionality

4. INFORMATION REQUIREMENTS

This clause specifies the information required for the exchange of status information between status database and shop floor data collection. The information requirements are specified as a set of units of functionality, application objects, and application assertions. These assertions pertain to individual application objects and to relationships between application objects. The information requirements are defined using terminology of the subject area of this IMES.

4.1 Units of Functionality (UoF)

This subsection specifies the units of functionality for the interface specification for discrete-event simulation. This IMES specifies the following units of functionality:

- logical elements UoF
- planning UoF
- process plan UoF
- product UoF
- resource UoF

The units of functionality and a description of the functions that each UoF supports are given below. The application objects included in the UoFs are defined in section 4.2.

4.1.1 Logical elements UoF

The Logical Elements UoF contains the information required for the management of logical elements usually employed in a simulation model. The following application objects are used by the Logical Elements UoF:

- attribute
- lookup table
- variable

4.1.2 Planning UoF

The Planning UoF contains the information that describes how to run the processes for the manufacturing and/or assembling of the products. This includes the description of arrival mechanism and the elements that affect the functioning of the production system. The following application objects are used by the Processing UoF:

- arrival
- breakdown
- maintenance
- shifts
- work schedule

4.1.3 Product UoF

The Product UoF contains the information that describes the product. This includes the description of all the parts composing it, and the description of the product structure. The following application objects are used by the Product UoF:

- load
- part

4.1.4 Process plan UoF

The Process Plan UoF contains the information that describes the process necessary to manufacture and/or assemble a product. This includes the description of all the steps needed to obtain the finished product from the raw material. The following application objects are used by the Process Plan UoF:

- process plan
- operation

4.1.5 Resource UoF

The Resource UoF contains the information that describes the physical equipment used to accomplish the fabrication and/or assembly of a product. This includes the processing system (machines), the material handling system, the storage system, and all the auxiliary equipment such as unit loads, tools, and fixtures. The following application objects are used by the Resource UoF:

- connection
- conveyor
- generic resource
- location
- operator
- path
- processor
- queue
- resource

- unit load
- vehicle

4.2 Application Objects

This subsection specifies the application objects for the discrete-event simulation interface specification. Each application object is an atomic element that embodies a unique application concept and contains attributes specifying the data elements of the object. The application objects and their definitions are given below.

4.2.1 Arrival

An Arrival cyclically releases a stream of one or more loads into the manufacturing system. The number of loads released can be limited, and the time between two successive releases can be a constant or a sample from a statistical distribution.

The data associated with an Arrival are the following:

- description
- first release time
- interarrival time
- load name
- max load count
- name

4.2.1.1 description

The description uses the word or group of words to describe the arrival.

4.2.1.2 first release time

The first release time is an expression indicating the time of the first release of a load with a demand (relative to the simulation start time).

4.2.1.3 interarrival time

The interarrival time is an expression (usually a distribution) indicating the time in hours between releases of loads according this arrival.

4.2.1.4 load name

The name of the load created by this arrival.

4.2.1.5 max load count

The max load count is the maximum number of loads to generate using this demand.

4.2.1.6 name

The name uniquely identifies the arrival.

4.2.2 Attribute

An Attribute is a place-holder attached to a load and usually contains information about that load. An Attribute is identified by its name and may be assigned a value.

The data associated with an Attribute are the following:

- id
- initial value
- name
- part
- type

4.2.2.1 id

The id uniquely identifies an attribute.

4.2.2.2 initial value

The initial value is an expression indicating the initial value of the attribute.

4.2.2.3 name

The name uniquely identifies the type of attribute.

4.2.2.4 part

The part is an option indicating which loads will be associated with this attribute. A specific load can be associated with the attribute (in this case the name-index number of the load is specified), or a class of loads (in this case the load name is specified), or a type of part (in this case, the part name is specified).

4.2.2.5 type

The type can be either a string (of one or more characters) or a number (real or integer).

4.2.3 Breakdown

A Breakdown is used to represent periods of time that one or more resources are down. Examples of down periods are: an automatic guided vehicle breaks down after spending 180 hours (on average) operating, operators call in sick about once every two months and they are absent between 1 and 3 days.

The data associated with a Breakdown are the following:

- application basis
- first breakdown value
- name
- repair time
- required resource
- required resource unit
- value between breakdowns

4.2.3.1 application basis

The application basis is an option that controls whether a breakdown event will be applied to units of a multi-unit resource on an individual basis or to all units together. It can be either *all* or *individual. All* indicates it will happen to all units of a multi-unit resource at the same time. *Individual* indicates the breakdown event will be scheduled for members of a multi-unit resource on an individual basis. The first breakdown, the time between breakdowns, and the repair time will be evaluated for each unit.

4.2.3.2 first breakdown value

The first breakdown value is an expression indicating the time (in hours) from simulation start that the first breakdown is to occur.

4.2.3.3 name

The name uniquely identifies the breakdown.

4.2.3.4 repair time

The repair time is an expression indicating the duration of the repair.

4.2.3.5 required resource

The required resource is the name of the resource required for repairing the breakdown.

4.2.3.6 required resource unit

The required resource unit is the number of units of the resource required for repairing the breakdown.

4.2.3.7 value between breakdowns

The value between breakdowns is an expression used to schedule periodic breakdowns.

4.2.4 Connection

The Connection enables a representation of the linking between the physical resources present in a manufacturing system: wherever a flow of parts is possible, a connection should be specified. The Connection can be mono or bi-directional since the flow of parts can be in one or both directions. Each resource should have at least one bi-directional connection or two mono-directional connections.

The data associated with the Connection are the following:

- resource 1
- resource 2
- direction
- name

4.2.4.1 resource 1

The resource 1 specifies the name of one of the resources connected.

4.2.4.2 resource 2

The resource 2 specifies the name of the second resource connected.

4.2.4.3 direction

The direction specifies if the connection is mono-directional or bi-directional (for mono-directional connection the direction is from resource 1 to resource 2).

4.2.4.4 name

The name uniquely identifies the connection.

4.2.5 Conveyor

A Conveyor is a material handling system able to transport parts along a fixed path. The Conveyor can be continuous loading, discrete loading, accumulating, or non-accumulating.

A continuous loading conveyor does not have fixed positions where loads can be loaded, therefore loads are not necessarily evenly spaced. A discrete loading conveyor has fixed positions where loads can be processed, and loads are evenly spaced on the conveyor. The distinction between continuous loading and discrete loading conveyors can be represented as an attribute named ***spacing***, which specifies the minimum distance between two adjacent parts.

In an *accumulating* conveyor, if the load in front of the conveyor is blocked, then loads will move forward until their progress is impeded by other loads that have accumulated. In a *non-accumulating* conveyor, if the load in front of the conveyor is blocked, all loads on the conveyor will stop at the exact position they occupied at the time of blocking.

The data associated with a Conveyor are the following:

- capacity
- length
- loading (continuous, discrete)
- part orientation
- spacing
- speed
- stop at load (yes, no)
- stop at unload (yes, no)
- type (accumulating, nonaccumulating)
- unit load

4.2.5.1 capacity

The capacity is a number that defines the number of loads which can be present on the conveyor at the same time (if defaulted the capacity is calculated by dividing the conveyor length for the spacing or the length of the part).

4.2.5.2 length

The length specifies the length of the conveyor.

4.2.5.3 loading (continuous, discrete)

The loading can be either continuous (there are not specific positions where loads can be loaded) or discrete parts can be loaded only in specific positions and loads are evenly spaced on the conveyor.

4.2.5.4 part orientation

The part orientation specifies the disposition of the part on the conveyor, it can have two values: length-wise or width-wise depending on whether the entity is traveling on the conveyor in the direction of the entity length or in the direction of its width.

4.2.5.5 spacing

The spacing is the minimum distance between fixed positions of the continuous loading conveyor.

4.2.5.6 speed

The speed specifies the speed of the conveyor.

4.2.5.7 stop at load (yes, no)

The stop at load specifies if the conveyor stops during the loading time.

4.2.5.8 stop at unload (yes, no)

The stop at unload specifies if the conveyor stops during the unloading time.

4.2.5.9 Type (accumulating, nonaccumulating)

The type specifies the behavior of the conveyor when a load is blocked at the front of a conveyor: the discrete loading conveyor can only block (stop moving), while the continuous loading can block or have loads accumulate, according to the type. When the conveyor is not accumulating, loads in the conveyor will stop exactly at the exact position they occupied at the time of blocking.

4.2.5.10 unit load

The unit load specifies the name(s) of the unit load required for the transporting of entities.

4.2.6 Generic Resource

A Generic Resource is any physical resource required for processing/transporting/storing a load. In order to perform their activities, processors, storage devices, and transporters have to be equipped with special tools and fixtures. A tool is a device used for processing a workpiece at or on a resource, for example, the probe on a coordinated machine. A fixture is a device used for positioning, holding, supporting, locating, or clamping a workpiece in the three-dimensional workspace of the processor. In this context, we consider only those fixtures that permanently remain with a workpiece. The traveling fixtures, or pallets, belong to the unit load class. Tools and fixtures can be associated with defined parts and defined operations. When fixtures are associated with defined parts, they must conform to part geometry, size, and shape. When they are associated with defined operations, tools and fixtures occur in specific combinations.

The basic idea is that tools and fixtures belong to pools of resources. When an operation requires a specific tool, the corresponding pool will provide that tool, if it is available. At the end of the operation, the tool might stay where it is, or it might come back to the pool. Sometimes, tools and fixtures are handled and transported using the same material handling system (MHS) employed for the part movement. Sometimes there is a separate system for each. For simplicity, we will not consider a separate MHS.

4.2.7 Load

The Load is the real physical entity that flows within the manufacturing system. Two different options to introduce loads within the manufacturing system are available: through a specific release (for each load the release time and date are specified), or through the arrival class (a generator of loads).

The data associated with a Load are the following:

- id
- name
- part name
- part quantity
- priority
- process plan name
- release date
- release time
- unit

4.2.7.1 id

The id specifies the unique identification of a load.

4.2.7.2 name

The name uniquely identifies the type of load.

4.2.7.3 part name

The part name specifies the name of the part(s) to be produced by this load (it can be composed of more parts, so this field can be an array).

4.2.7.4 part quantity

The part quantity specifies the number of parts, of each type, in a whole load (it will be an array if the part name is always an array).

4.2.7.5 priority

The priority is a numeric weighting factor giving this load a priority value in relation to other loads.

4.2.7.6 process plan name

The process plan name identifies the process plan associated with this load.

4.2.7.7 release date

The release date specifies the date when the load will be released into production.

4.2.7.8 release time

The release time specifies the time when the load will be released into production.

4.2.7.9 unit

The unit specifies the number of identical loads to be released in the manufacturing system.

4.2.8 Location

A Location is a specific point where loads can be loaded/unloaded in the material handling system (interface points between the material handling system and the rest of the system) and where branching decisions are made. For example, in an automatic guided vehicle (AGV) system, locations are specific points where the vehicle can stop for loading/unloading parts or for making branching decisions (merge or diverge points). For locations where branching decisions are made, a routing rule has to be specified. The data associated with a Location is the parking location: the parking location is an option specifying whether this location is a valid parking location for a vehicle.

4.2.9 Lookup table

A Lookup Table is a two-dimensional matrix. One or two indices are defined, and, during the simulation, whenever a particular pair of indices is encountered, the associated value is looked up in the table and used.

The data associated with a Lookup Table are the following:

- left index
- top index
- name
- value

4.2.9.1 left index

The left index is the name of the first index (on the left) associated with this lookup table.

4.2.9.2 top index

The top index is the name of the second index (in the top row) associated with this lookup table.

4.2.9.3 name

The name uniquely identifies the table.

4.2.9.4 value

The value is the number indicating the value associated with a particular index.

4.2.10 Maintenance

A Maintenance represents a period of time that one or more resources are down. Examples of these down periods are monthly preventative maintenance on a milling machine causes it to be unavailable for 2 hours, and automatic guided vehicle (AGV) battery changes performed after every 100 hours of vehicle operation. The difference between breakdowns and maintenance is that the maintenance is scheduled to happen at a specific time.

The data associated with a Maintenance are the following:

- application basis
- first maintenance date
- first maintenance time
- maintenance time
- name
- required resource
- required resource unit
- time between maintenance events

4.2.10.1 application basis

The application basis is an option that controls whether a maintenance event will be applied to units of a multi-unit resource on an individual basis or to all units together. It can assume two values: all, it will happen to all units of a multi-unit resource at the same time; individual, the maintenance event will be scheduled for members of a multi-unit resource on an individual basis.

4.2.10.2 first maintenance date

The first maintenance date is the date that the first maintenance event will start.

4.2.10.3 first maintenance time

The first maintenance time is the time that the first maintenance event will start.

4.2.10.4 maintenance time

The maintenance time is an expression used to schedule the end of maintenance.

4.2.10.5 name

The name uniquely identifies the maintenance.

4.2.10.6 required resource

The required resource specifies the name of the resource required for the maintenance.

4.2.10.7 required resource unit

The required resource unit specifies the number of units of the resource required for the maintenance.

4.2.10.8 time between maintenance events

The time between maintenance events is an expression used to schedule periodic maintenance events.

4.2.11 Operation

An Operation is a discrete step that consists of a set of actions which leads to a well defined state change, physical or not, in the part. An Operation might require one or more resources, such as a processor and/or an operator, so that the operation is performed on the part. The objective in the definition of the attributes for the operation class is to have a class flexible enough to represent both rough process plans and detailed process plans. In order to represent all the described kinds of operations, the operation class must have different attributes, but some of the attributes are optional (for example, the rejection rate attribute will not be meaningful if the operation is not an inspection operation). In the most complex case, an operation can be decomposed in four different sub-operations: setup, loading, processing, and unloading. Each sub-operation might have different times and require different resources; in order to have an extremely flexible operation class, attributes related to the four suboperations are specified.

The data associated with an Operation are the following

- input part quantity
- input part type
- input parts
- loading required resource
- loading required resource unit
- loading time
- name
- output part quantity

- output part type
- output parts
- processing required resource
- processing required resource unit
- processing time
- rejection rate
- resource name
- rework operation
- scrap rate
- setup basis
- setup required resource
- setup required resource unit
- setup time
- unloading required resource
- unloading required resource unit
- unloading time

4.2.11.1 input part quantity

The input part quantity specifies, for each type of part (according to the name(s) specified in input part), the quantity of loads or single parts (according to the input part type) to be accumulated before starting the operation.

4.2.11.2 input part type

The input part type specifies if the value of the input part quantity is referred to parts or loads.

4.2.11.3 input parts

The input part specifies the name of the part(s) required in order to start the operation (it is possible that more types of parts are required to perform this operation: in this case, it is possible to input an array of names)

4.2.11.4 loading required resource

The loading required resource is the name of the resource(s) required for loading (for example, an operator is needed to perform the loading).

4.2.11.5 loading required resource unit

The loading required resource unit is the number of units of the resource(s) required for loading.

4.2.11.6 loading time

The loading time indicates the time(s) required for loading the part/load(s).

4.2.11.7 name

The name uniquely identifies the operation (it can also be an operation number).

4.2.11.8 output part quantity

The output part quantity specifies the number of loads or the number of parts for each load (according to the output part type) in output to the operation.

4.2.11.9 output part type

The output part type specifies if the output part quantity is referring to parts or loads.

4.2.11.10 output parts

The output part specifies the name of the part(s) in output to the operation.

4.2.11.11 processing required resource

The processing required resource is the name of the resource(s) required for processing (for example, an operator is needed to perform the processing).

4.2.11.12 processing required resource unit

The processing required resource unit is the number of units of the resource(s) required for processing.

4.2.11.13 processing time

The processing time indicates the time(s) required to process the part.

4.2.11.14 rejection rate

The rejection rate indicates the probability that the load is rejected.

4.2.11.15 resource name

The resource name specifies the name of the resource(s) that performs the operation on the part.

4.2.11.16 rework operation

The rework operation specifies the name of the operation used to rework the rejected load.

4.2.11.17 scrap rate

The scrap rate specifies the probability that the load is defective.

4.2.11.18 setup basis

The setup basis specifies on what setup is based. It can be performed always (prior to processing a load), load (based on change from the last load to the current load), or part (it is based on changes in part name).

4.2.11.19 setup required resource

The setup required resource is the name of the resource(s) required for setup (for example, an operator is needed to perform the setup).

4.2.11.20 setup required resource unit

The setup required resource unit is the number of units of the resource(s) required for setup.

4.2.11.21 setup time

The setup time indicates the time(s) required to setup the resource.

4.2.11.22 unloading required resource

The unloading required resource is the name of the resource(s) required for the unloading (for example, an operator is needed to perform the unloading).

4.2.11.23 unloading required resource unit

The unloading required resource unit is the number of units of the resource(s) required for unloading.

4.2.11.24 unloading time

The unloading time indicates the time(s) required to unload the part from the resource.

4.2.12 Operator

An Operator is a person performing the following functions within a manufacturing system: transporting entities, assisting in performing manual operations on entities (for example, loading and unloading the part on a processor), performing maintenance of other resources, and so on. The data associated with an Operator is the speed: it defines the speed to be used for any of the operator's movements along a path network.

4.2.13 Part

A Part is an individual item that is to be produced by the manufacturing system; it may be an assembly composed of other parts or a single item. It is an abstraction of product, subassembly and item, present in the *bill of material* (BOM).

A *product* consists of *subassemblies* and/or *components*. In turn a subassembly, in similar fashion to a product, is composed of subassemblies and/or components. Typically the product structure is represented through the BOM, i.e., a tree showing the subassemblies and components of a product (see Fig. 1).

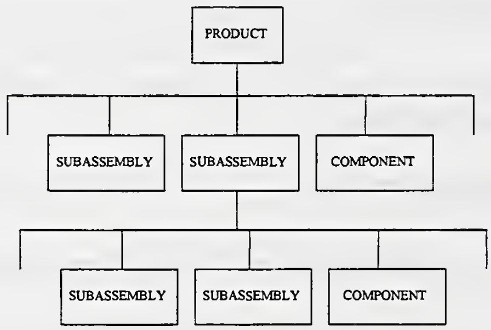

Fig. 5- Bill of material (BOM) tree

The following attributes can be defined for the class Part:

- icon
- length
- name
- type (product, subassembly, component)
- width

4.2.13.1 icon

The icon is the name of the icon used to graphically represent the part.

4.2.13.2 length

The length is the length of the part.

4.2.13.3 name

The name uniquely identifies the part.

4.2.13.4 Type (product, subassembly, component)

The type can be the final product, the subassembly or the component.

4.2.13.5 *width*[1]

The width is the width of the part.

4.2.14 Path

A Path is a portion of the transporter layout (or track); a path connects two different locations.

The data associated with a Path are the following:

- capacity
- direction
- length
- maximum speed
- shape
- type

4.2.14.1 *capacity*

The capacity specifies the number of vehicles that can be present on the path at the same time (it is employed when the number of vehicles in a path has to be less than that calculated by dividing the segment length by the vehicle length).

4.2.14.2 *direction*

The direction specifies if the path can be traversed in one or both directions.

4.2.14.3 *length*

The length specifies the length of the path.

4.2.14.4 *maximum speed*

The maximum speed is the maximum vehicle speed for that path.

4.2.14.5 *shape*

The shape specifies if the path is straight or curved.

---

[1] The width and the length are used to determine the number of entities that can fit on a conveyor. Which side a user chooses to call the length or width is unimportant as long as the proper side is referenced when defining a conveyor.

4.2.14.6 type

The type can either be passing or non-passing. Passing indicates that entities and resources are free to overtake one another; non-passing networks consist of single tracks such as those used for AGVs where vehicles are not able to pass.

4.2.15 Process plan

A Process Plan specifies the sequence of operations, which must be performed on a load. A Process Plan can be expressed at different levels of detail: in the first stages of the manufacturing systems engineering process, the designer deals with a 'rough process plan', which specifies the operations to be performed in a very rough way (with average times). The process plan, as the process evolves, is specified at a more detailed level: the 'detailed process plan', which in some cases can coincide with the routing, specifies all the single movements of the part within the manufacturing systems, like the various batching/unbatching operations performed within the manufacturing system. In this document the idea is to deal with a detailed process plan that describes in detail the operations to be performed on a load.

The data associated with a Process Plan are the following:

- name
- operation list

4.2.15.1 name

The name uniquely identifies a process plan.

4.2.15.2 operation list

The operation list is the set of one or more operations comprising the process plan.

4.2.16 Processor

A Processor is a resource, like a machining center, an inspection device, and assembly machine. It is also a location dedicated to the function of processing a part, where processing means any value-added activity required to transform raw materials into finished products. Processors are the physical elements of the manufacturing system used to manufacture parts. Different kinds of processors can be present in a manufacturing system, like inspection machines, assembly processors, or machining processors. From the simulation viewpoint, it's not important to distinguish between these different kinds of processors, as all of them have the same behavior: they take in one or more parts and, after a time lag, they produce one or more parts.

The following data can be associated with the Processor:

- initial setup

- max number of processes
- min number of processes
- unit load

4.2.16.1 Initial setup

The initial setup is the name of the part for which this processor is set up at simulation start.

4.2.16.2 max number of processes

The maximum number of processes is the maximum number of loads that can be processed at the same time.

4.2.16.3 min number of processes

The minimum number of processes specifies the minimum number of loads that can be processed at the same time.

4.2.16.4 unit load

The unit load specifies the name of the unit load(s) required for the processing of entities.

4.2.17 Queue

A Queue is a single buffer able to house material for staging or building inventory. Usually the commercial off-the-shelf simulation packages available for the modeling of manufacturing systems do not provide *ad hoc* constructs for the modeling of a combined buffer. For this reason, the combined buffer has not been considered as a simulation class. The queue class is just a proxy of the combined buffer: the withdrawal time attribute specifies the average time required for the withdrawal of a unit load from the storage system.

The data associated to a Queue are the following:

- capacity
- initial stock
- queue rule
- unit load
- withdrawal time

4.2.17.1 capacity

The capacity is the maximum amount of loads that can be stored in the material storage area.

4.2.17.2 initial stock

The initial stock is the number of loads already present in the queue at the beginning of the simulation.

4.2.17.3 queue rule

The queue rule specifies the withdrawal rule. It can be last in -first out, first in-first out, short processing time, long processing time, random, or user specified.

4.2.17.4 unit load

The unit load specifies the name of the unit load(s) required for the storing of entities in the queue.

4.2.17.5 withdrawal time

The withdrawal time is the average time required for the withdrawal of the unit load from the storage.

4.2.18 Resource

A Resource is a physical component present in a manufacturing system. The Resources are allocated to the loads through the operations. Each resource is a conveyor (see 4.2.4), fixture (see 4.2.5), location (see 4.2.8), operator (see 4.2.12), path (see 4.2.14), queue (see 4.2.16), tool (see 4.2.20), unit load (see 4.2.21), or a vehicle (see 4.2.23).

The data associated with a Resource are the following:

- icon
- id
- name
- unit

4.2.18.1 icon

The icon specifies the name of the icon used to visualize the resource.

4.2.18.2 id

The id specifies the unique identification of a resource.

4.2.18.3 name

The name uniquely identifies the type of resource.

4.2.18.4 unit

The unit specifies the number of identical resources.

4.2.19 Shift

A Shift specifies the set of shift intervals (time periods) that are used to determine when the resources and material-handling systems are available to perform operations. If no shifts are specified, the resources are available 24 hours a day, 7 days a week. The shift is specified in terms of a week. This weekly pattern is repeated for all weeks in the simulation time-frame. A week starts Sunday morning at 00:00:00 and runs until Saturday at midnight. All periods within a week that are not specified within an interval period are assigned to be down. When a shift reaches the end of an 'up' interval, the resources identified as being on that shift are made unavailable.

The data associated with the Shift are the following:

- ending day
- ending time
- name
- starting day
- starting time

4.2.19.1 ending day

The ending day is the day of the week at which the 'up' period ends. It can assume the following values: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday.

4.2.19.2 ending time

The ending time is the time for the end of the shift interval (in terms of hh, mm, ss).

4.2.19.3 name

The name identifies uniquely the shift.

4.2.19.4 starting day

The starting day is the day of the week at which the 'up' period starts. It can assume the following values: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday.

4.2.19.5 starting time

The starting time is the time for the start of the shift interval (in terms of hh, mm, ss).

4.2.20 Unit load

The Unit Load, or pallet, represents the physical means employed for transporting the parts (both single parts and loads) among the different resources. Unit loads are made in different sizes and from different materials determined by the size, weight, geometry, environmental requirements, etc. of the goods handled. They can be of different types, the most common in the manufacturing systems are bins, boxes, baskets, tot pans, disposable and reusable pallets. Sometimes transporters, processors, and storage equipment require specific unit loads (for example, an automatic storage and retrieval system can store only certain kinds of unit loads). In this case the part has to be loaded in the specific unit load in order to be processed. For this reason the attribute unit load has been added to processors, transporters, and queues. This attribute specifies the name of the specific unit load required for processing/transporting/storing the load.

The data associated with the Unit Load are the following:

- capacity
- length
- width

4.2.20.1 capacity

The capacity is the maximum number of parts that can be contained in the unit load.

4.2.20.2 length

The length is the length of the unit load.

4.2.20.3 width[2]

The width is the width of the unit load.

4.2.21 Variable

A Variable is a place-holder defined by the user to represent changing numeric values.

The data associated with a Variable are the following:

- Initial value
- Name

[2] The width and the length are used to determine the number of unit loads that can fit on a conveyor. It is unimportant which side a user chooses to call the length or width as long as the proper side is referenced when defining a conveyor.

- type

4.2.21.1 initial value

The initial value is an expression indicating the initial value of the variable.

4.2.21.2 name

The name uniquely identifies the variable.

4.2.21.3 type

The type of attribute can be either a string (of one or more characters) or a number (real or integer).

4.2.22 Vehicle

The Vehicle is used to represent all the kinds of trucks, both manually operated and computer controlled, present in a manufacturing system; they can transport multiple parts from one location to another. The vehicle always travels along a specific *path* or route. Sometimes the path can be *guided*, like in the case of AGVs or cart-on-track, or it can be *free*, like in the case of a hand cart. In simulation, it is not meaningful to distinguish between *free path* and *guided path*, because in order to simulate the behavior of the vehicle, in both cases, the path followed has to be specified. The distinction between *fixed route* and *semi-fixed route* guided path vehicle, we can consider the fixed route guided path vehicle as a special case of the semi-fixed route guided path vehicle, in which the vehicle can go through only one path. Sometimes the vehicle can present lifting capability, for example the counterbalanced truck. The distinction between *non-lifting* and *lifting* free path vehicle, an attribute can specify if the vehicle can lift the load, and additional attributes can specify the lifting load, and so on.

The data associated with a Vehicle are the following:

- acceleration
- capacity
- curve speed
- deceleration
- default position
- length
- lifting speed
- loaded speed
- maximum speed
- type
- unit load

4.2.22.1 acceleration

The acceleration specifies the acceleration of the vehicle.

4.2.22.2 capacity

The capacity specifies the number of loads the vehicle can transport at one time.

4.2.22.3 curve speed

The curve speed is the speed when the vehicle is traversing a curved path.

4.2.22.4 deceleration

The deceleration specifies the deceleration of the vehicle.

4.2.22.5 default position

The default position specifies the name of the location(s) where the vehicle is located at the start of the simulation.

4.2.22.6 length

The length specifies the length of the vehicle.

4.2.22.7 lifting speed

The lifting speed specifies the lifting speed of the vehicle.

4.2.22.8 loaded speed

The loaded speed specifies the speed of the vehicle when the vehicle is loaded.

4.2.22.9 maximum speed

The maximum speed specifies the speed of the vehicle when it is empty.

4.2.22.10 type

The type specifies if the vehicle requires the presence of an operator in order to perform a transportation mission.

4.2.22.11 unit load

The unit load is the name of the unit load(s) required for the transporting of entities.

4.2.23 Work schedule

The Work Schedule component is used to specify periods of scheduled down time for all resources. It addresses production down time such as holidays, vacations and plant shutdowns for inventory and maintenance. It should be used to specify all major holidays and other periods where the plant is not scheduled to operate.

The data associated with the Work Schedule are the following:

- ending day
- ending time
- name
- shutdown name
- starting day
- starting time

4.2.23.1 ending day

The ending day is the date when the production shutdown ends.

4.2.23.2 ending time

The ending time is the time when the production shutdown ends.

4.2.23.3 name

The name identifies uniquely the work schedule.

4.2.23.4 shutdown name

The shutdown name is a name which identifies the specific shutdown (generally there is a list of shutdowns for each work schedule).

4.2.23.5 starting day

The starting day is the date when the production shutdown starts.

4.2.23.6 starting time

The starting time is the time when the production shutdown starts.

4.3 Application Assertions

This subsection specifies the application assertions for the interface specification for discrete event simulation. Application assertions specify all relationships among application objects, the cardinality of the relationships and the rules required for the integrity and validity of the application objects, and UoFs. The application assertions and their definitions are given below.

4.3.1 Arrival to Load

Each Arrival creates one or more Loads. Each Load is created by zero or one Arrival.

4.3.2 Connection to Resource

Each connection connects exactly two Resources. Each Resource is connected to other resources by at least one connection.

4.3.3 Load to Attribute

Each Load is associated to zero or more Attributes. Each Attribute is associated with exactly one Load.

4.3.4 Load to Process plan

Each Load can follow different Process Plans during its life cycle. Each Process Plan can be associated to one or more Loads.

4.3.5 Load to Part

Each Load is composed of one or more Parts. Each Part belongs to zero, one, or more Loads.

4.3.6 Operation to Process plan

Each Operation is part of one or more Process Plans. Each Process Plan is composed of one or more Operation.

4.3.7 Operation to Resource

Each Operation is related to at least one Resource. Each Resource is associated with zero, one, or more Operation.

4.3.8 Resource to Breakdown

Each Resource can be related to zero or one Breakdown. Each Breakdown is associated with one or more Resources.

4.3.9 Resource to Maintenance

Each Resource can be related to zero or one Maintenance. Each Maintenance is associated with one or more Resources.

4.3.10 Resource to Shift

Each Resource can be related to zero or one Shift. Each Shift is associated with one or more Resources.

4.3.11 Resource to Work schedule

Each Resource can be related to zero or one Work schedule. Each Work schedule is associated with one or more Resources.

5. REFERENCES

[1] Barkmeyer, E. (ed.), "SIMA Reference Architecture, Part 1: Activity Models", NISTIR 5939, National Institute of Standards and Technology, Gaithersburg, MD, December 1996.

[2] Barkmeyer, E., Hopp, T., Pratt, M., and Rinaudot, G., "Background Study -Requisite Elements, Rationale, and Technology Overview for the System Integration for Manufacturing Applications (SIMA) Program, NISTIR 5662, National Institute of Standards and Technology, Gaithersburg, MD, September, 1995.

[3] Howe, S., *High Performance Computing and Communications: Toward a National Information Infrastructure*, Government Printing Office, Washington, DC, USA, 1994.

[4] Kemmerer, S. and Fowler J., "Initial Manufacturing Exchange Specification (IMES): IMES Concept Document For Manufacturing Systems Integration", NISTIR 5978, National Institute of Standards and Technology, Gaithersburg, MD, February 1997.

[5] LeCapitaine C., Riddick F., and Jones A., "Production Management Standards: Requirements Analysis for Shop Floor Status", NISTIR 6123, National Institute of Standards and Technology, Gaithersburg, MD, March 1998.

[6] McLean, C. and Leong, S., "Industrial need: Production Systems Engineering Integration Standards", NISTIR 6019, National Institute of Standards and Technology, Gaithersburg, MD, May 1997.

[7] Meta Software Corp., Design/IDEF User's Manual and Tutorial for Microsoft Windows, Meta Software Corp., 1994.