Achieving Autonomy Through Design

Jerzy W. Rozenblit

Department of Electrical and Computer Engineering
The University of Arizona
Tucson, Arizona 85721, U.S.A

Abstract

This article discusses a high level design methodology and its support of high autonomy systems synthesis. Requirements for high autonomy as well as the design framework are briefly summarized. Then, model-based techniques which unify autonomous system design are presented.

1 Introduction

Autonomy as a design goal can be defined as the ability of a system to function independently, subject to its own laws and control principles. Quick strides are being undertaken to achieve high autonomy in engineering designs as evidenced by recent research and development of high autonomy systems [1, 3, 4, 7, 8, 18]. Work in *high autonomy* stems, to a large extent, from NASA's Space Station program and its Systems Autonomy Demonstration Project [3, 8]. This project focuses on research in artificial intelligence (AI), human factors, and dynamic control systems in support of Space Station automation and robotics technology [3, 8].

Most AI and expert systems (ES) tools and methods which have been successfully applied to planning, scheduling, diagnosis, and control, the applications treat the above functions as separate entities. A salient requirement in a highly autonomous system is that such, and similar, functions can be integrated to support the operation of a complete system. Architectures that foster the integration have been proposed in the literature by a number of authors [1, 7, 17, 19]. The common basis for the proposed designs is *automatic intelligent control.*

New approaches to design of high autonomy systems emerge which take a distinct, simulation modeling approach. While the majority of new developments are application oriented, they do provide building blocks that are eventually evolving into a comprehensive methodology for autonomous systems design [10].

In this article, we briefly summarize the definition of an intelligent autonomous system and requirements for achieving high autonomy. Then, model-based techniques which unify autonomous system design are presented.

2 Intelligent Autonomous Architecture

Erickson and Cheeseman [3] stipulate that the following behavioral requirements be met in an intelligent autonomous system:

- the system must plan and re-plan to realize its goals
- the system must be able to execute its plans
- the system must monitor its environment
- it must have cognitive capabilities
- it must have diagnostic capabilities

A generic architecture is postulated in which such requirements are addressed by the ***system interface***, ***the planning, scheduling, and reasoning layer***, and ***the control and sensing layer.*** The architecture is depicted in Figure 1.

The flow of information and control in the above architecture is defined in detail in [3, 10, 18].

Antsaklis et. al. [1] refine the architecture by breaking up its functions into ***management and organization level*** that determines the overall system's goals and supports interaction with the system's external environment through the interface unit, ***coordination level*** that supports decision making, planning,

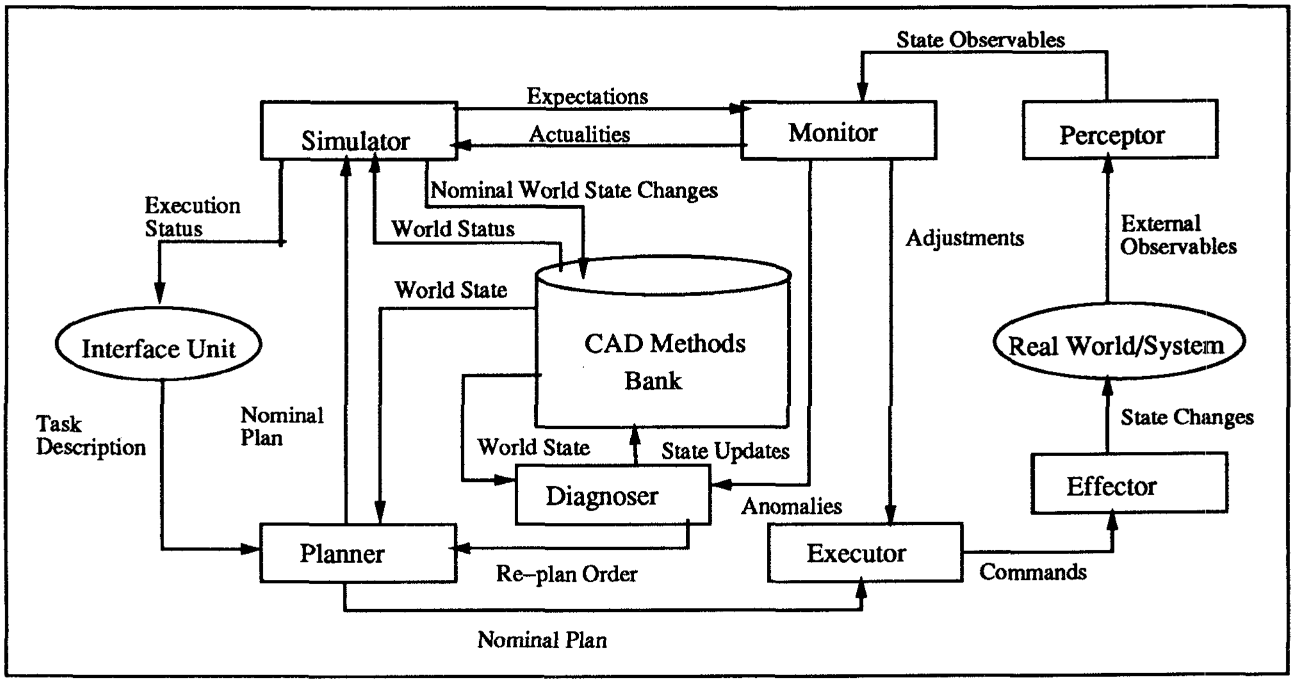

Figure 1: Intelligent Autonomous Architecture (adopted from [3] and [18])

and scheduling, and *execution level* that carries out control actions determined at higher levels through automatic controllers and actuators.

Zeigler [18] encapsulates knowledge in the form of models that can be employed at different levels of control in an autonomous system to support it's objectives. The resulting structure is termed ***model-based architecture***. The key distinguishing feature of Zeigler's approach in the use of partial models to deal with the multiplicity of system's objectives and functions.

An important consideration in the design of high autonomy systems is the degree of autonomy supported [3]. The definition of *degree of autonomy* is often given in qualitative rather than in quantitative measures. Erickson and Cheeseman [3] define the degree of autonomy as the extent of a system's interaction with its environment through the interface unit. NASA, in its Telerobotics Project [3, 8], defines the degree of autonomy in terms of the level of detail and abstraction that the human operator has to employ when assigning tasks, and how long robots can function on their own without any intervention from their operator. In the model-based architecture, Zeigler [18] defines progressive levels of autonomy achievement as follows:

- *Level 1*: the system should have the ability to achieve its objectives.
- *Level 2*: the system should be able to adapt to major environmental changes.
- *Level 3*: the system should be able to develop its own objectives.

We shall examine how the above three objectives can be met by applying a knowledge-based methodology to design of high autonomy systems.

3 Knowledge-Based Design

"Design is a goal directed activity producing a set of descriptions of an artifact that satisfy a set of given performance requirements and constraints [2]." A number of methodologies and design systems have been developed to aid the engineering design process in different domains [2, 23]

We have been developing a generic design methodology which is specifically well suited for design of hierarchical, modular systems [12, 14, 15]. The framework, called Knowledge-Based Simulation Design Methodology, uses modeling and simulation techniques to build and evaluate models of the system being designed.

The design construction process begins with developing a representation of design components and their variants. To appropriately represent the family of design configurations, we have proposed a representation scheme called the *system entity structure* (SES) [20, 22]. The scheme captures the decomposition, taxonomic and coupling relationships among objects of a design domain.

Procedural knowledge is available in the form of production rules. They can be used to manipulate the elements in the design domain by appropriately selecting and synthesizing the domain's components. This selection and synthesis process is called *pruning* [14, 15]. Pruning results in a recommendation for a *model composition tree*, i.e., a set of hierarchically arranged system objects. Those objects are related to models of design components, which are stored in the design domain-related *model base*.

The final step in the framework is the evaluation of alternative designs. This is accomplished by simulation of models derived from the composition trees. Discrete Event System Specification (DEVS) [20, 21, 22] and other formalisms are used for system specification in the methodology. Performance of design models is evaluated through computer simulation.

The real system component and its design have not received adequate attention in the literature that reviews autonomous systems. We stipulate that the real system design process be an inherent phase in the development of a complete autonomous system. Methods should be available to establish how the environment in which the autonomous architecture is to operate affects the design of this architecture. Conversely, the autonomy requirements will also impinge on the real system design. We now address those issues in more detail.

4 Knowledge Representation and Management

Design domain experts should assist in the development of a comprehensive knowledge representation for both the Real System and Autonomous Architecture components of the Intelligent Autonomous Systems. To accomplish this, we use the system entity structure and its derivative called Frames and Rules Associated System Entity Structure (FRASES). FRASES combines the SES, frames, and rule-based representations. An underlying data structure of FRASES is the system entity tree. Each node of the SES tree has a frame attached to it that encompasses declarative and procedural knowledge in a design problem. Such a frame is called *Entity Information Frame* (EIF). An Entity Information Frame (EIF) integrates design knowledge by providing slots for representing design procedural knowledge [9].

Since most engineering applications are well structured, the development of a comprehensive FRASES structure is relatively easy. A repository of knowledge form previous designs is often available available and new designs are rarely entirely innovative. Refer to Figure 2 which represents a system entity structure of an Intelligent Autonomous System. (For the sake of brevity, we represent only the SES nodes without associated frames.) Assume that the Real World design domain is Flexible Manufacturing. The top level SES of a Flexible Manufacturing Systems is shown in Figure 3. Figure 4 shows possible choices for configuring the Robot component of the FMS.

The Autonomous Architecture has a decomposition that reflects the requirements defined in Section 2. Thus, at the highest level of abstraction, all the major components (i.e., Perceptor, Effector, Executor, etc.) must be identified in the SES. These components can be further decomposed and classified. This results in a knowledge base of components that can be used in domain-specific design. To illustrate this point, let us consider the Perceptor module. This module, as shown in Figure 2, may have a wide range of sensors, e.g., temperature and pressure sensors, tactile sensors, range detectors, television cameras, audio sensors. Similarly, the Effectors may include activating relays, servomotors, valve shutoffs, etc. Requirements and constraints for a specific design problem will determine which sensors and actuators should be used in the system being designed. For example, high precision tactile sensors will not be recommended for applications in which objects operate in extremely high temperatures.

Let us now focus on the representation for the robot structure. The components that interface the autonomous architecture with its real world system (e.g., sensors and effectors) occur in their respective SES representations. The ***uniformity axiom*** [20] precludes the designer from duplicating those entities in the tree.

In addition to the structure representation of the Autonomous Architecture, its operational aspects can be captured by a SES as well. Zeigler, Chi [19], and Luh [7] give several examples that illustrate SES/MODEL-based plan generation for a robot man-

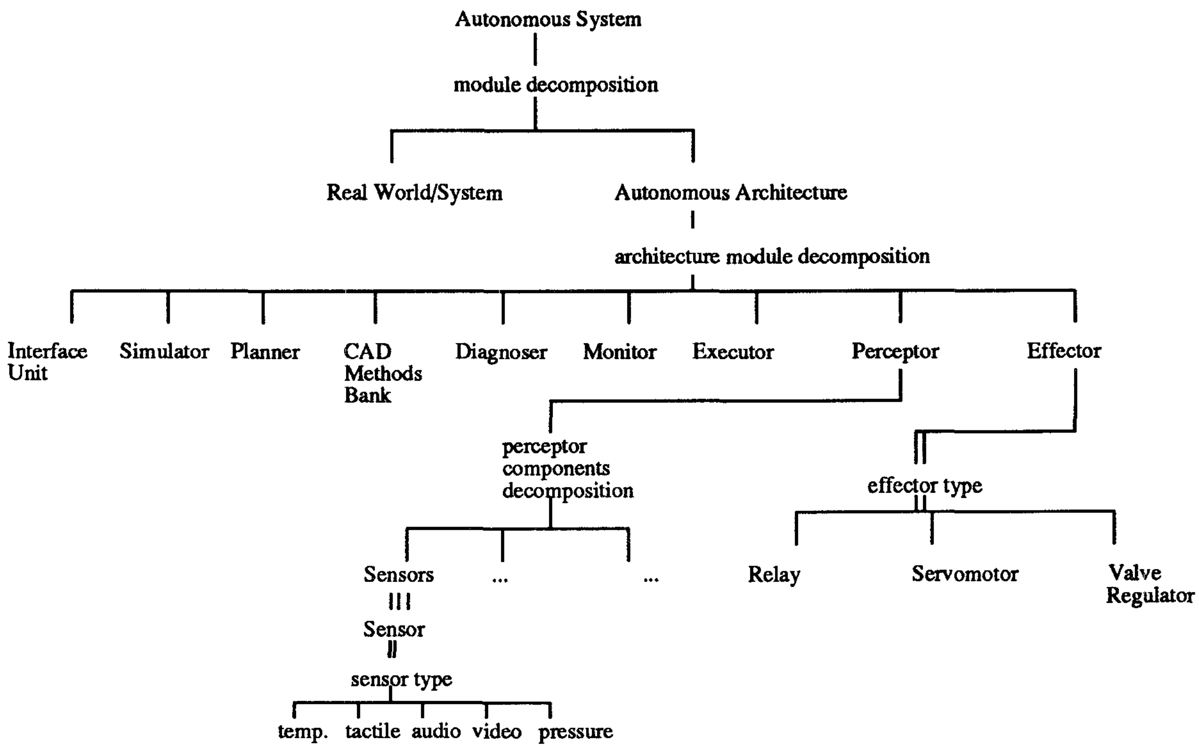

Figure 2: High Level SES of an Autonomous System

aged chemical laboratory. Their approach uses a hierarchical goal structure representation from which a specific task can be formulated. Each subtask has an associated execution model. For example, a robot-based machining operation may be represented as part-retrieval, drilling, and finishing. Each of the three subtasks can be decomposed further in a hierarchical manner. For example, part-retrieval can be represented as pick-up-part, move-part, place-part.

The system entity structure (and its FRASES extension) underlies a family of all possible choices for an intelligent autonomous system design. Selection of specific components and final system configuration depends on the specific design constraints and requirements for achieving autonomy.

5 System Structure Synthesis

In this phase, a design description (in terms of the system's structure and topology) is generated. In our design framework, this is accomplished by rule based-based pruning [14]. Domain experts should translate constraints into pruning rules. Here, the difficulty lies in translating the autonomy requirements into design constraints. The general desiderata such as: the system must plan to realize its goals, the system must monitor its environment, it must have diagnostic capabilities, etc. translate into selection of planners, monitors, diagnoses, etc.

However, the definition of what constitutes high degree of autonomy is still imprecise. Thus, the formulation of design constraints is largely dependent on what designers perceive as desirable characteristics, and is done on a case-by-case basis [3]. For example, many designers consider a mobile robot to be more autonomous then a fixed one. When synthesizing a robot-based automation system, the higher the autonomy required, the more likely they are to select mobile robots.

Recall Figure 4. A rule for selecting a mobile robot may take the following form:

```
R-sel
 if desired autonomy is high or medium  and
    required working area scope is > 25
    square feet  and
    max  arm load <= 1000 lbs and
    available budget is high
```

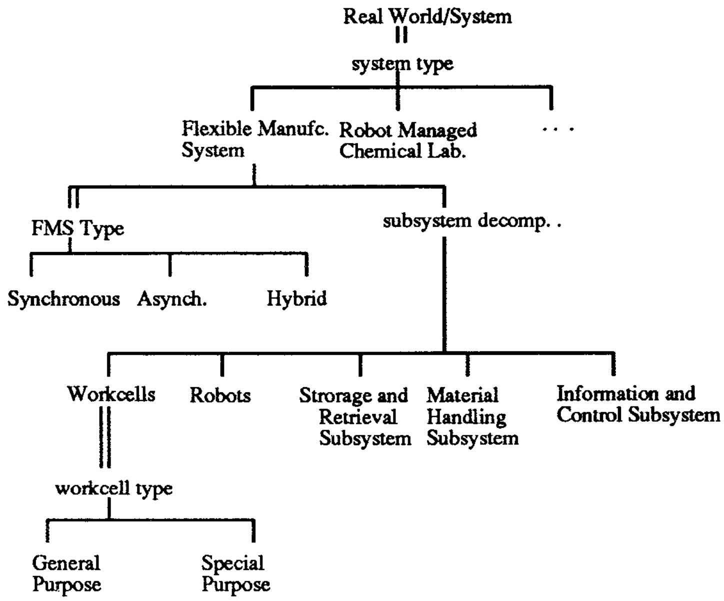

Figure 3: SES of an FMS

```
then
   recommended robot-type is mobile (0.9)
```

Such a selection rule is part of the Robot Entity Information Frame and is used when design consultation process (pruning) is invoked.

The engineering type constraints are easier to handle. For example, if device temperature measurement is required, we must select a temperature sensor or if an assembly system is to operate in hazardous conditions (e.g., high toxicity) only robot-managed workcells can be used. However, the designer must ensure the consistency of the knowledge base so that selections of various system's components are compatible. For instance if optical sensors are to be used in monitoring the production flow in a manufacturing workcell, the monitor module of the autonomous architecture must be capable of processing optical data.

Another important design aspect is the development of pruning mechanisms that will enable us to interface the execution sequence planning with planning of the real system structural design. We have pointed out the importance of this problem in the context of manufacturing systems (layout design) [6, 16]. Here, we briefly explain the concept: Assume that a manufacturing task is defines as assembly of an electric motor. Methods exists that employ an AND/OR graph representation of the motor to generate all feasible assembly sequence trees (plans) [5]. Such a tree may take the form given in Figure 5. A topology of an assembly workcell which can carry out the plan is shown in Figure 6.

6 Modeling

The key supposition of our approach is the use of simulation models to evaluate alternative design solutions. The System Entity Structure/Model Based framework proposed by Zeigler [22] is employed to generate models of the real system, families of planning alternatives, and to build a hierarchical event-based control structure. We refer the reader to [18] for further details.

7 Execution and Performance Evaluation

In principle, an autonomous system could base its operation on a comprehensive model of its environment (and itself). In the model-based architecture partial models of different levels of abstraction are employed. The partial models are oriented towards specific objectives, and thus need to be evaluated in

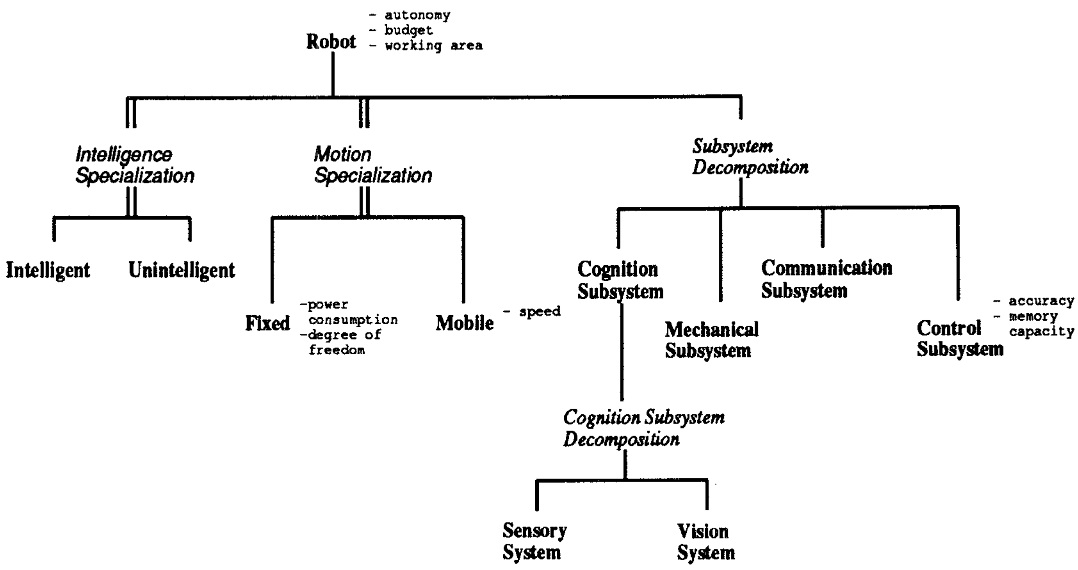

Figure 4: Robot SES

respective experimental frames that reflect those objectives.

The experimental frame specification methodology [11] provides a systematic approach to defining a set of conditions under which an autonomous system is to operate. Consider for the workcell of Figure 6. To measure the utilization profile of the system, we can define a frame as follows:

```
Experimental Frame: UTILIZATION PROFILE

Input Segment:
  None (parts are retrieved from
        feeder buffers)

Output Variables: :
  Machine Status s(m1), s(m2)
      with range {busy, idle}

Summary Variables:
  Utilization Profile:
  u(m1) = time(m1, busy)/total_time
  u(m2) = time(m2, busy)/total_time
```

The distributed frame architecture proposed by Rozenblit [11] supports flexible experimentation with multicomponent systems that may exhibit various degrees of distribution and coordination among their components. The degree of autonomy of individual system components may be observed in local frames or within higher level frames that assess the coordination/cooperation among the components.

8 Conclusions

To conclude, we assess the value of a general methodology to support design of high autonomy systems with respect to the three levels of degree of autonomy presented in Section 2.

Level 1: the system should have the ability to achieve its objectives.

Experimental frames are a means for collecting quantitative data about the degree to which a system is capable of achieving its objectives. Endomorphic modeling facilitates the evaluation of the operational management of the autonomous architecture, i.e., planning, scheduling, diagnosis, and control.

Level 2: the system should be able to adapt to major environmental changes.

The system entity structure and pruning algorithm facilitate rapid knowledge-based selection and configuration of components in the design domain. Structure reconfiguration which may become necessary due to major environment changes or failures of equipment can be enabled by re-pruning the SES underlying the design at hand. Designs are modelled and simulated prior to being deployed. This considerably reduces

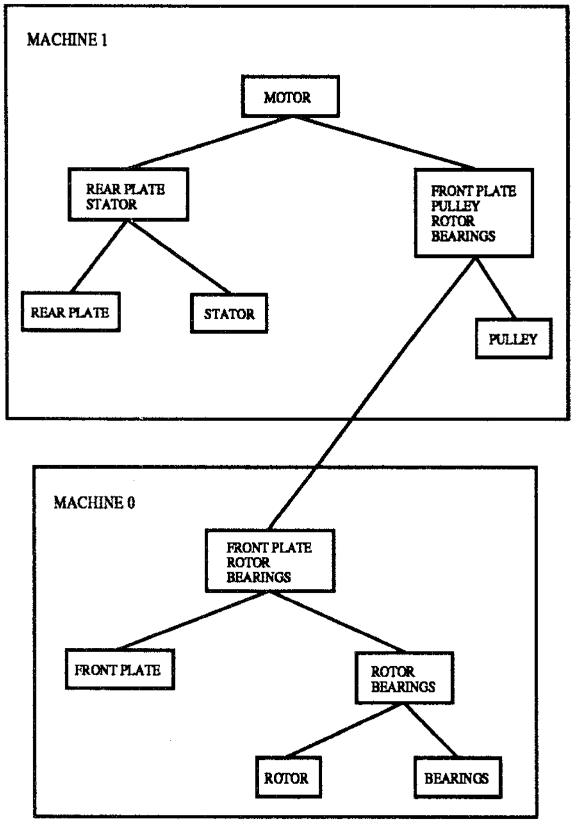

Figure 5: Electrical Motor Assembly Tree

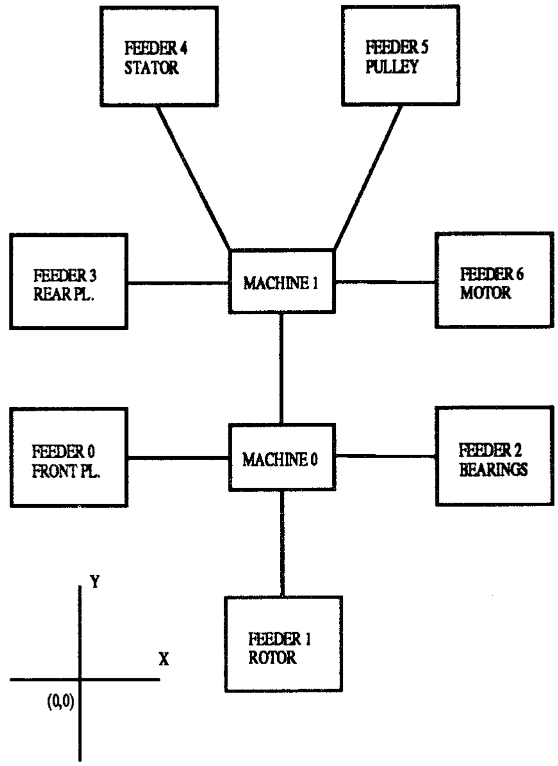

Figure 6: WorkCell Topology for Motor Assembly

the cost of system implementation and its potential re-design.

Level 3: the system should be able to develop its own objectives.

New objectives can be defined as experimental frames. For example, given a frame that collects data about machine utilization in a manufacturing system, a supervisory control unit may set the following goal: *increase utilization of components.* The goal is set based on observations collected from the Utilization Profile Frame. Consequently, several actions can be undertaken. For example, given a constant job arrival rate, the number of machines which carry out the same task can be decreased.

References

[1] P.J. Antsaklis, K.M. Casino and S.J. Wang. Towards Intelligent Autonomous Control Systems: Architecture and Fundamental Issues. *Journal of Intelligent and Robotic Systems*, vol. 1, no. 4, 315-342, 1989.

[2] R. D. Coyne. *Knowledge-Based Design Systems.* Addison-Wesley, 1990.

[3] W. K. Erickson and P.C. Cheeseman. Issues in Design of an Executive Controller Shell for Space Station Automation. *Optical Engineering*, vol. 25, no 11, 1194-1199, 1986.

[4] P.A. Fishwick, J.W. Rozenblit and B.P. Zeigler (Editors). *Proc. of the 2nd Conference on AI, Simulation, and Planning in High Autonomy Systems*, IEEE Press, April 1991.

[5] L.S. Homem De Mello and A. C. Sanderson. AND/OR Graph Representation of Assembly Plans. *IEEE Trans. on Robotics and Automation*, vol. 6, no 2, 188-199, 1990.

[6] W. Jacak and J.W. Rozenblit, Automatic Simulation, Interpretation and Testing of a Task-Oriented Robot Program for a Sequential Technological Process, Robotica, vol.10, 45-56, 1992.

[7] C.J. Luh and B.P. Zeigler. Abstraction Morphisms for Task Planning and Execution. *Proc. of the 2nd Conf. on AI, Simulation, and Planning in High Autonomy Systems*, IEEE Computer Society Press, 50-59, 1991.

[8] NASA, *The Space Station Program.* NASA Publication, 1985.

[9] J.W. Rozenblit and J. Hu. Integrated Knowledge Representation and Management in Simulation Based Design Generation, *Mathematics and Computers in Simulation*, vol. 34, 1992.

[10] J.W. Rozenbit. Design for High Autonomy: An Overview, *Applied Artificial Intelligence*, vol 6. no 1. pp. 1-19, 1992.

[11] J.W. Rozenblit. Experimental Frame Specification Methodology for Hierarchical Simulation Modeling, *International Journal of General Systems.* vol. 18, no. 4, 317-336, 1991.

[12] J.W. Rozenblit and B.P. Zeigler. Design and Modeling Concepts. *International Encyclopedia of Robotics, Applications and Automation*, (ed. Dorf, R.) John Wiley and Sons, New York, 308-322, 1988.

[13] J.W. Rozenblit and B.P. Zeigler (Editors). *Proc. of the 1st Conference on AI, Simulation, and Planning in High Autonomy Systems*, IEEE Press, April 1991.

[14] J.W. Rozenblit and Y. M. Huang. Rule-Based Generation of Model Structures in Multifacetted Modeling and System Design. *ORSA Journal on Computing*, vol. 3, no. 4, 330-344, 1991.

[15] J. W. Rozenblit and B.P. Zeigler, Knowledge-Based Simulation Design Methodology: A Flexible Test Architecture Application, *Transactions of the Society for Computer Simulation*, vol. 7 no. 3, 195-228, 1990.

[16] J. W. Rozenblit and W. Jacak. Simulation Based Planning of Robot Tasks in Flexible Manufacturing, *Proceedings of the 2nd AI, Simulation and Planning in High Autonomy Systems Conference*, 166-173, IEEE Computer Society Press, April 1991.

[17] G.N. Sardis. Intelligent Robotic Controls, *IEEE Trans. on Automatic Control*, AC-28, no. 5, 1983.

[18] B.P. Zeigler. Endomorphic Modeling Concepts for High Autonomy Architectures. *Applied Artificial Intelligence*, vol. 6, no. 1, pp. 19-33, 1992.

[19] B.P. Zeigler and S.D. Chi. Model-Based Architectures for Autonomous Systems. *Proc. of the 1990 IEEE Symp. on Intelligent Control.* Philadelphia, PA, 27-32, 1990.

[20] B.P. Zeigler. *Multifacetted Modelling and Discrete Event Simulation,* Academic Press, 1984.

[21] B.P. Zeigler. System-Theoretic Representation of Simulation Models, *IIE Trans.* March, pp. 19-34, 1984.

[22] B.P. Zeigler, B.P., *Object-Oriented Simulation with Hierarchical Modular Models: Intelligent Agents and Endomorphic Systems*, Academic Press, 1990.

[23] J. Yang and J.W. Rozenblit, Case Studies of Design Methodologies: A Survey, *Proc. of the International Conference on AI, Simulation and Planning in High Autonomy Systems*, IEEE Computer Press, pp. 136-141, 1990.