Accepted for publication in the journal
"International Journal of General Systems"

Hierarchical System Concepts for Simulation of High Autonomy Systems

Tag Gon Kim
Department of Electrical and Computer Engineering
University of Kansas
Lawrence, KS 66045

ABSTRACT

A high autonomy system is an intelligent system with the self-determination power. Such a system must perform a variety of functions such as reasoning, planning, sensing, control, and other activities necessary to achieve predefined system objectives over an extended period of time under uncertainties in its environment. Simulation modelling can be a powerful tool for design of such complex autonomy systems. This paper describes the hierarchical, modular system concepts for simulation of high autonomy systems with an example of an intelligent control system called the AIDECS (AI-based, Distributed Environmental Control System). The paper emphasizes on development of methods, in an object-oriented manner, for isomorphic replications of complex hierarchical structures as a means of constructing hierarchical models. We show how polymorphism and inheritance in the object-oriented programming can be exploited to develop such methods in DEVS-Scheme, a realization of the DEVS formalism in a LISP-based, object-oriented framework.

Key Words: High Autonomy System, Hierarchical System, Object-oriented Programming, DEVS-Scheme, Polymorphism, Models Isomorphism.

1. INTRODUCTION

A high autonomy system is an intelligent system with the self-determination power. Such a system must perform a variety of functions such as reasoning, planning, sensing, control, and other activities necessary to achieve predefined system objectives over an extended period of time under uncertainties in its environment. The architecture of such a system may employ at least three control layers, namely, an execution layer for control and sensing, a management layer for high-level decision-making to achieve overall goals, and a coordination layer for linking the two layers (Saridis, 1983).

At the intersection of artificial intelligence, automatic control, and operations research, design of autonomy systems clearly requires the tools of artificial intelligence and simulation to successfully integrate decision making and physical layers (Zeigler and Rozenblit, 1990). Since an autonomy system operates as an independent unit with layers structure, hierarchical structuring with modular models may be a good approach to simulation modelling of such a system. The main advantage of such a hierarchical, modular approach in simulation modelling is that hierarchial construction provides a convenient means for connecting components in a structured manner while modularity promotes modification and inter-substitution of components at the various levels of the hierarchy (Zeigler, 1990).

As high autonomy systems have multi-components, each having its own subcomponents, simulation models for such systems become complex, hierarchical structures. Often, such complex structures can be conveniently constructed by using isomorphic copies of existing models. Thus, special attention should be paid to check isomorphism between models before using them as component models. To create isomorphic copies

of various class of models in a unified manner, each class must have its own method to create such copies. Thus, object-oriented programming is well suited to developing methods for such a copying process.

Object-oriented programming paradigm is well noted to be compatible to modelling real world systems (O'Keefe, 1986; Zeigler, 1987a). Indeed, DEVS-Scheme realized Zeigler's DEVS (Discrete Event System Specification) formalism in a LISP-based, object-oriented environment. The DEVS-Scheme environment supports specification of discrete-event models in modular, hierarchical fashion (Zeigler, 1987b; Kim and Zeigler, 1990b), a systems oriented approach not possible in conventional simulation languages.

This paper describes the concepts of hierarchical system for simulation of high autonomy systems and development of complex hierarchical models based on the concepts. Specifically, it develops methods for creating isomorphic copies of complex, hierarchical models in DEVS-Scheme as a means of constructing yet complex hierarchical simulation models. Section 2 gives an introductory example of a high autonomy system to motivate simulation based design of high autonomy systems. In section 3, we introduce the concepts of modular, hierarchical system for simulation modelling in such design. Sections 4 and 5 briefly describe the object-oriented programming paradigm, and the simulation concepts and class evolution in the DEVS-Scheme environment, respectively. Section 6 develops methods for isomorphic replications of complex, hierarchical models and the associated methods for checking isomorphism between models in the DEVS-Scheme environment. Conclusions are given in section 7.