Planning for Dynamic Configuration of Component-based Architectures*

Gail Haddock
University of Texas - Arlington
Department of Computer Science Engineering
416 Yates
Arlington, Texas 76019
haddock@csr.uta.edu

Abstract

The Scenario-based Engineering Process (SEP) is a novel approach to developing complex systems. SEP builds new application systems through a selection process that groups primitive components into application specific components. The selection of primitive components and the construction of interfaces among components in an application system is currently a tedious manual undertaking. The automation of this process will require a configuration system that can support the complex interactions of the components, the dynamic requirements of users, and the capabilities of providing multiple viewpoints and managing extensive domains. This paper will discuss our work on a configuration system based upon UM-PRS, the University of Michigan Procedural Reasoning System. To support the SEP methodology, we are developing a system, SEPRS, that will use planning techniques for component selection, integrated with semantic understanding techniques for dynamic interface construction, to create a flexible and powerful system engineering environment.

Introduction

We are building a system engineering environment that will improve the productivity and efficiency of complex system development. This environment requires the development of sophisticated tools for knowledge acquisition, system analysis, architecture design, configuration, implementation, and simulation. This paper focuses on the configuration tool under construction. It will be built upon the architecture of UM-PRS, a coordinated planning and control system that is being developed for the autonomous vehicle domain. Many parallels exist between the selection of appropriate plans for immediate circumstances and the selection of appropriate components for dynamic user requirements. This paper will discuss those parallels.

Section 2 introduces the Scenario-based Engineering Process (SEP) for constructing complex systems. The University of Michigan Procedural Reasoning System, UM-PRS, is introduced in Section 3 as a plan selection, execution, and monitoring system. Section 4 discusses the parallels between the two systems, and introduces SEPRS, a variation of UM-PRS that will perform necessary configuration functions for SEP.

* This research has been partially supported by the National Science Foundation, the National Center for Manufacturing Sciences, and the Advanced Research Projects Agency.

SEP

The Scenario-based Engineering Process is an open systems architecture-based approach to system engineering (Haddock 1993). Reference level specifications are used to create new application systems that will easily integrate with other systems developed through the same specifications, as shown in figure 1. *Reference requirements* describe a system's capabilities in general terms without specific values or determinations. A *reference architecture* contains primitive components that map to the references requirements. One primitive component consists of one basic responsibility necessary for part of the requirements to be met, the resources needed by the responsibility, and the products resulting from the responsibility. An *architectural description language* is used as rules for guiding the selection of primitive components. *Domain models* are used for understanding the context of primitive components (Haddock & Harbison 1994).

The reference requirements, reference architecture, and domain models provide an abstracted view of the system. The application requirements are used to create an application from the reference representations by guiding the selection and grouping of primitive components from the reference architecture into components in the application architecture. The process of creating an application architecture iterates through a) application requirements definition, b) selection of primitive components from the reference architecture, c) selection or deletion of other primitive components in the application architecture according to the architecture description language rules, d) leading back to the reference architecture for additional primitive components.

Selected components are integrated into an application architecture by specifying the interfaces between one component's resources and another component's products. Messages are built according to the responsibilities and contexts of the components. Automatic interface construction is currently being developed as part of the Adaptive Semantic Language (Hannon 1994). Given a set of

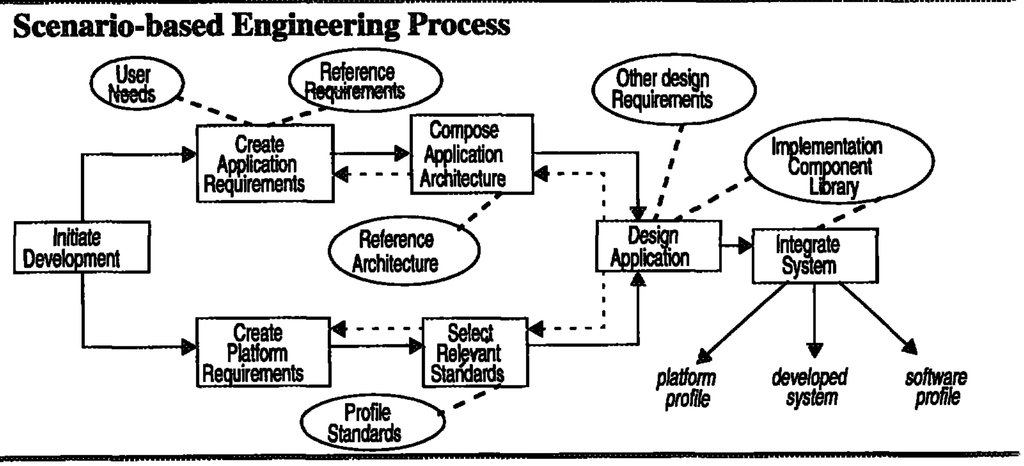

Figure 1

components and a new component to be added to the set, the Adaptive Semantic Language will create the interfaces and messages necessary for the new component to communicate with the existing set. This is done using modified human language understanding techniques, retaining the rich semantics of human language while limiting the ambiguity.

Once the application architecture is determined, the application system is developed by separate organizations taking responsibility for building various components. Design requirements, standards, profiles, and computational models are used to complete the system using the application architecture as a guide. Thus, an organization can focus on areas in which they excel, confident that their product will interface with products from other organizarions and together create a successful system.

SEP is has been used to develop specifications for the Next Generation Controller project funded by the National Center for Manufacturing Sciences. It is also being used in the Trauma Care Information Management System project funded by the Advanced Research Projects Agency. Both projects are lead by Dr. Karan Harbison at the University of Texas - Arlington.

UM-PRS

The University of Michigan Procedural Reasoning System, UM-PRS (Lee et al. 1993), a C++ implementation of PRS (Georgeff & Lansky, 1990), is a reactive system and not a typical planning system, see figure 2. Rather than determine a path from a goal back to a present state by an ordered application of operators, UM-PRS assumes these types of plans already exist. Its strength is in its determination of when to use which plan in a rapidly changing execution environment. A belief area is used to hold a representation of the environment. Knowledge areas (KA) contain previously built plans along with their purpose and context. A set of knowledge areas is selected as being pertinent to the current world model and is maintained by the intention structure. Goals are used to guide the system's actions. An interpreter controls the system's activities.

UM-PRS is currently being used in the autonomous vehicle domain. Its ability to continually consider the real-time dynamic environment and access plans accordingly fits well into military applications, where plans have already been generated in the form of standing operating procedures and reactions to the quickly changing environment are paramount. Our SEP domain does not require the hard real-time speed of a reactionary system. How-

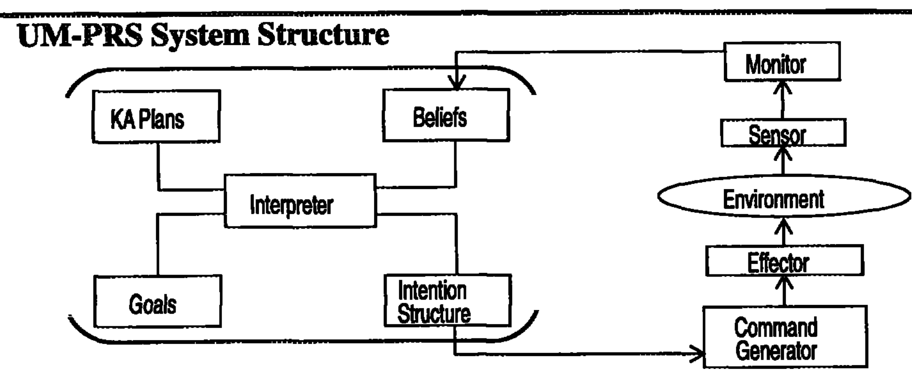

Figure 2

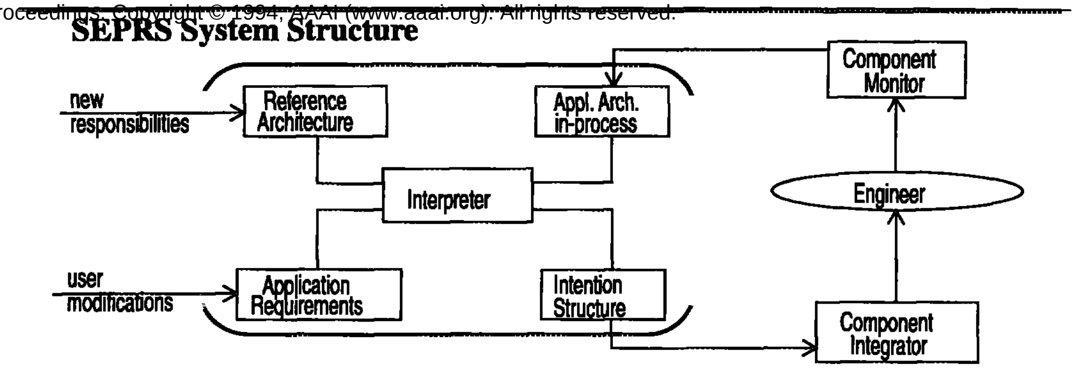

Figure 3

ever, much of the UM-PRS architecture maps readily to the configuration problem in the SEP domain.

SEPRS

The Scenario-based Engineering Procedural Reasoning System, SEPRS, will use the architecture of UM-PRS to implement a configuration system for SEP, shown in figure 3. Primitive components of the reference architecture will take the place of knowledge area plans, as each primitive component is a responsibility that has its own sub-responsibilities, resource requirements, and product interfaces. The architecture description language will also reside in the reference architecture area, as it contains rules associated with the primitive components. The goals will be replaced by reference/application requirements.

The interpreter will use the application requirements as goals to satisfy through access to the reference architecture. Components previously formed into an application architecture in-progress will be in the beliefs area. They are accessed by the interpreter to determine which goals are not yet satisfied. The interpreter will activate relevant primitive components that are maintained by the intention structure. The intention structure will release chosen primitive components to the component integrator. The component integrator will employ Adaptive Semantic Language techniques to build the interfaces and messages necessary for adding the primitive component to the application architecture.

The grouping of primitive components into components remains a manual task, as this grouping can be done from a variety of viewpoints. For example, some groupings may be done for marketing purposes. Other groupings may indicate the assignment of construction of the component to various organizations. The environment area of UM-PRS then becomes our system engineer, directing the grouping process. The system engineer's modifications are added back to SEPRS through a component monitor, who sends the component determinations into the in-progress area, thus completing the cycle.

Building the SEP configuration system on the SEPRS architecture supports more than the component selection process. Modifications to the application requirements are also supported. Since requirements are maintained in the goals area and are continually accessed by the interpreter, user modifications can be interjected at any point in the architecture creation cycle. These modifications may immediately cause primitive components to be deselected and their interfaces disconnected, which may then require an extensive reconfiguration of the architecture. The SEPRS architecture also supports the expansion of the reference architecture. As new technologies are invented that result in new responsibilities, those responsibilities can be added through the reference architecture area.

Conclusion

We are building a configuration system, SEPRS, for component-based architecture methodologies by adapting the UM-PRS planning system. We expect it to fit well in our system engineering environment that includes scenario modeling, object-oriented analysis and design, and simulation systems. These systems will be interconnected through the Yellow Rose, a LOOM-based semantic network kernel that allows systems to share information on an application under development. Work on SEPRS and the Yellow Rose is continuing.

References

Georgeff, M. and Lansky, A. 1990. Reactive Reasoning and Planning. Allen, J., Hendler, J., and Tate, A., eds., *Readings in Planning*, San Mateo, Ca: Morgan Kaufmann.

Haddock, G. 1993. The Scenario-based Engineering Process: A Lifecycle Incorporating Formalisms, Methodologies, and Tools into Domain Specific Software Architectures. Master's thesis, Dept. of Computer Science Engineering, University of Texas at Arlington.

Haddock, G., and Harbison, K. 1994. From Scenarios to Domain Models; Processes and Representations. In Proceeding of Knowledge-based Artificial Intelligence Systems in Aerospace and Industry. Bellingham, Wa: International Society for Optical Engineering.

Hannon, C. 1994. An n-Towers Model for the Knowledge Representation of an Adaptive Semantic Language. Master's thesis, Dept. of Computer Science Engineering, University of Texas at Arlington.

Lee, J., Huber, M., Durfee, E., Kenny, P. 1993. UM-PRS: An Implementation of the Procedural Reasoning System. Technical Report, University of Michigan