N90-27296

KNOWLEDGE-BASED SYSTEMS AND NASA'S SOFTWARE SUPPORT ENVIRONMENT

Tim Dugan, PRC, Houston
Cora Carmody, PRC, Houston
Kent Lennington, LMSC, Houston
Bob Nelson, NASA, Reston

ABSTRACT

This paper describes a proposed role for knowledge-based systems within NASA's Software Support Environment (SSE). The SSE is chartered to support all software development for the Space Station Freedom Program (SSFP). This includes support for development of knowledge-based systems and the integration of these systems with conventional software systems. In addition to the support of development of knowledge-based systems, various software development functions provided by the SSE will utilize knowledge-based systems technology.

KEY WORDS

Ada, Development Tools, Knowledge-Based Systems, Software Engineering, Space Station

INTRODUCTION

NASA'S SOFTWARE SUPPORT ENVIRONMENT

In order to provide a complete and consistent support environment for software development for the Space Station Freedom Program (SSFP), NASA initiated the Software Support Environment (SSE) project. The SSE is intended to support the life-cycle management of operational software (both ground and flight software) as well as the life-cycle management of the software for the SSE itself. The SSE software consists of both Commercial Off-The-Shelf (COTS) applications and custom software augmented by methods, procedures, standards, documentation, and training materials to support users of the environment. The SSE will evolve over the life cycle of the SSFP in order to be responsive to its user community's needs.

THE RELATIONSHIP OF KNOWLEDGE-BASED SYSTEMS TO THE SSE

The uses of knowledge-based systems technology within the SSE may be classified as either:

- support for the development of knowledge-based systems products and the integration and deployment of these products with conventional software systems, or

- support for SSE operations.

Figure 1 illustrates the relationship of knowledge-based systems technology to the SSE and SSFP.

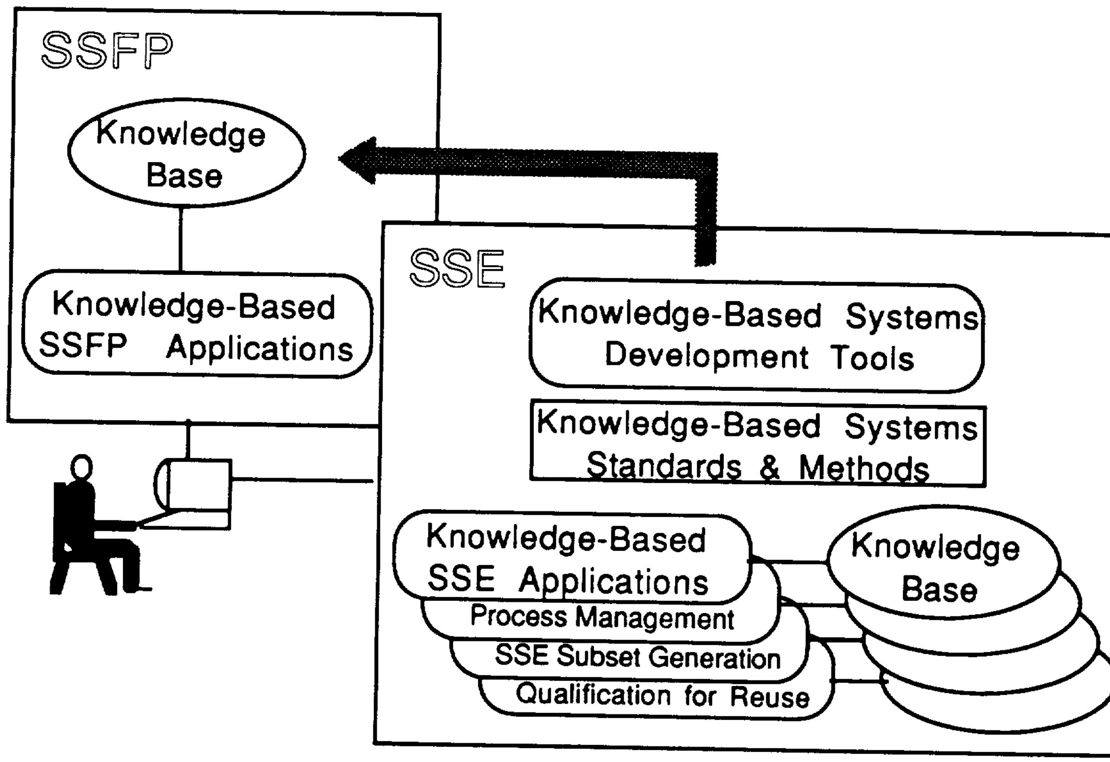

Figure 1. Two Uses of Knowledge-Based Systems Technology in the SSE

The remainder of this paper describes SSE support for and use of knowledge-based systems in more detail.

KNOWLEDGE-BASED SYSTEMS DEVELOPMENT

Increased feasibility of knowledge-based systems technology within an operational environment has produced a significant demand for automated development support. A large variety of commercial software systems are available that support the development and deployment of knowledge-based systems. In order to provide criteria for selecting existing tool sets or developing custom tools, an initial set of functional requirements for these tools has been developed. Based upon these requirements, a preliminary design of the knowledge-based support tools was also produced. In order to supplement the knowledge-based systems development environment, the SSE will provide support for an expert systems development methodology. A description of the status of these efforts follows.

REQUIREMENTS ON THE SSE FOR A KNOWLEDGE-BASED SYSTEMS DEVELOPMENT ENVIRONMENT

The SSE contractor has defined a set of functional requirements for the SSE's knowledge-based systems development environment. These requirements are documented in detail in [LMSC-1] and summarized in [LMSC-4].

According to these requirements, the SSE will provide the necessary support to develop and deploy knowledge-based applications. This support includes capabilities for developers of applications to choose from a variety of knowledge representation schemes and reasoning strategies. The five broad categories of requirements include knowledge representation, reasoning strategies, external software integration, development, and delivery.

Knowledge Representation Requirements. The knowledge-based systems support tools will provide several different integrated knowledge representation schemes. One of these is production rules which are constructed from conditional patterns, priorities, and resulting actions. External procedures may be invoked from either the pattern or action part of a rule. A second form of knowledge representation is performed through object manipulations. Objects will support strong typing, multiple inheritance, uncertainty, truth maintenance, and external procedure calls.

Reasoning Strategies Requirements. The knowledge-based systems support tools will provide for reasoning using forward- and backward-chaining of rules. Reasoning strategies will include reasoning about uncertain data, hypothetical reasoning, constraint checking, and access to external procedures through rules and objects.

External Software Integration Requirements. The knowledge-based systems support tools will provide interfaces to external software systems. This includes the capability to execute functions attached to objects and rules which are implemented in other languages, particularly Ada. The knowledge-based systems support tools will include an interface to the knowledge base which may be accessed asynchronously from other languages.

Development Requirements. The knowledge-based systems support tools will provide an environment for developing, testing, debugging and validating knowledge-based applications in an integrated environment. This environment includes an integrated knowledge base editor with incremental compilation capabilities. Debugging support includes the capability to trace the execution of rules within a knowledge-based system through the use of break points. Users will have the ability to interactively query a knowledge base and change its content. Additionally, users will have access to on-line help and explanation facilities. And, finally, the environment will provide the capability to save its current state in a file to be recreated at a later time.

Delivery Requirements. A user of the knowledge-based systems support tools will have the capability to deploy knowledge-based applications onto SSFP flight hardware and ground elements of the Space Station Information System. Knowledge-based systems may execute either as an interactive adviser or as an embedded subsystem of another application. Deployed versions of a knowledge-based application need not contain code for functions of the knowledge-based system support tools which are not required by a particular application. There will be no built-in restrictions on the size of a knowledge base and no restrictions on the number of users of the static portion of a knowledge base.

Note that deployment of knowledge-based applications onto SSFP flight hardware implies certain performance constraints. The exact nature of these requirements has not been investigated at this time.

THE PRELIMINARY DESIGN FOR KNOWLEDGE-BASED SYSTEMS SUPPORT TOOLS

A high-level preliminary design of the SSE software is presented in [LMSC-3]. This document provides a context for the design of the knowledge-based systems support environment which is described in more detail in [LMSC-1] This document furthers the perspective that, in terms of an overall software system, a knowledge-based application is just another software component. However, the development of a knowledge-based system distinguishes it from traditional software systems in that the use of interactive development environments provide the capability to build successively more sophisticated prototypes of the desired system. Once a prototype of a knowledge-based system has reached a sufficiently mature form, it may be deployed in the form of an Ada package which may be embedded in a larger system.

Figure 2 depicts a Buhr-style diagram portraying high-level view of the knowledge-based systems support tool. The major components include a development interface which allows the user to interact with the environment to develop and deploy an application and an inference engine which supports object management, pattern matching, and execution management.

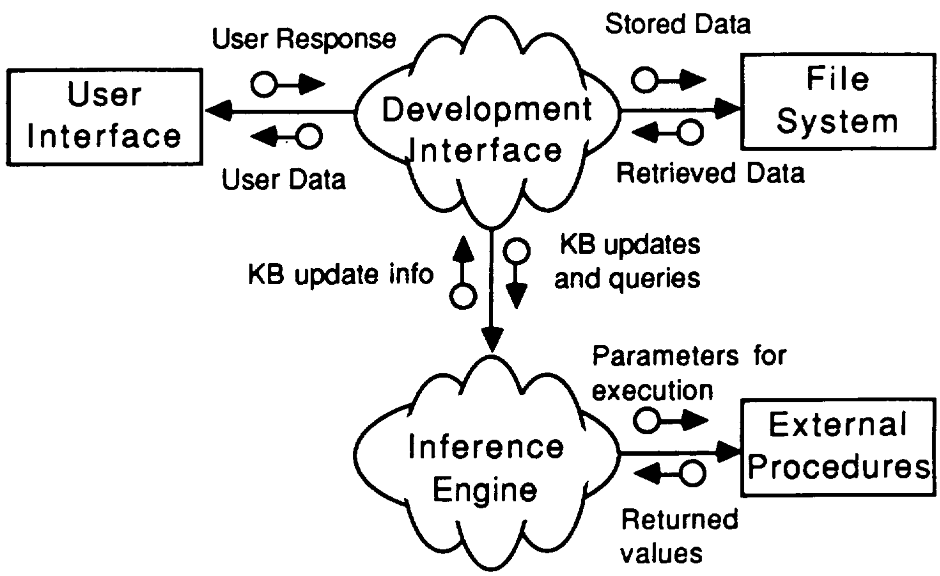

Figure 2. Knowledge-Based Tool Preliminary Design

METHODS SUPPORT FOR EXPERT SYSTEMS DEVELOPMENT

The SSE contractor is in the process of defining methodologies to be supported by the SSE as described in [LMSC-2]. Due to important differences in the life cycle of expert systems, the contractor has also prepared a supplementary document [LMSC-1] that describes the differences between the methodological support for traditional software systems and that needed for expert system development. The supplement emphasizes iterative knowledge acquisition, the use of prototypes, validation techniques, and user interfaces for expert systems.

IMPLEMENTATION OPTIONS

Some of the options for implementing SSE knowledge-based systems development tools that are under consideration include the purchase of an existing COTS product or the enhancement of a public domain product known as CLIPS. The final decision will depend in part on a comparison of the short-term benefits for developing or modifying a given product to meet the requirements compared to the long-term benefits of using that product. Adherence by the tool to any existing or emerging standard should play a role in the selection.

USE OF KNOWLEDGE-BASED SYSTEMS BY THE SSE

The SSE is itself a large, complex software system. As such, it provides many opportunities for utilizing knowledge-based systems technology. Although not all areas where knowledge-based systems may be used have been identified, some of the candidate areas are described below.

SUBSET GENERATION

One important function of the final SSE system is the capability to automatically configure new Software Production Facilities (SPFs) and individual projects at SPFs. This capability is referred to as subset generation because each SPF contains some subset of the complete set of SSE capabilities. The subset generation capability is somewhat similar to DEC's XCON as described in [Waterman] and [Barker]. XCON provides for automated configuration of VAX computer systems. XCON performs these configurations in a fraction of the time that a technician would take to perform the same task. XCON is implemented in the language OPS5.

Much like XCON, the SSE subset generator will produce candidate configurations for SPFs and individual projects at SPFs based upon the needs at that site. Human users will then analyze the resulting SPFs to determine the validity of the configuration. Once it has been determined reasonable, the subset generator function will provide at least partial automated support in configuring the SPF.

SOFTWARE LIFE-CYCLE PROCESSES

A second area that is a candidate for use of knowledge-based technologies is the automation and control of software life-cycle processes. This goal evolved from a desire to coordinate unconnected tools into an integrated development environment such as that described in [Bisiani et al]. These goals were combined with the SSE goal of providing consistent life-cycle management for all SSFP software producing a desire to manage development processes through process programming. Some research indicates that a mix of procedural and rule-based approaches are best for describing such processes [Taylor]. To supplement this view, there is an emerging body of theory relating to process programming which is described in [Tully].

The management of life-cycle processes as they are applied to development products will require an interface between the knowledge-based application and the SSE project object base. One recommended approach outlined in [McKay] is to provide a standardized interface to the project object base through the use of an Information Resource Dictionary System (IRDS). The IRDS standard (ANSI X3H4) allows the development of a semantic model of a project object base in an entity-relationship form.

Reusable Library

A critical need for the SSE is to provide a useful library of reusable components. Unlike typical reusable libraries, the SSE reusable library will include not just reusable code but reusable objects from all phases of the software life cycle. The two primary uses of knowledge-based systems technology as it applies to reusable libraries are the qualification of components for inclusion in the reusable library and the search for reusable components based upon functional requirements.

CONCLUSION

Summary

This SSE support for development of knowledge-based systems has been described and references for further information has been provided. Additionally, some of the planned uses of knowledge-based systems within the SSE have been described.

Future Directions

Future directions for support of knowledge-based systems are currently dependent on the recognition of new requirements based upon the needs of SSFP contractors. Future directions for the utilization of knowledge base technology within the SSE are also dependent on the determination of need as well as the identification of emerging uses of knowledge-based systems that support software engineering. Some examples include:

- Project management support as described in [Bimson]
- Performance tuning of the SSE System such as is described for UNIX® in [Samadi].

Work Pack Contractors may influence the directions of the SSE by contacting an SSE support representative or submitting an SSE Change Request.

®UNIX is a registered trademark of AT&T Bell Laboratories.

REFERENCES

[Barker] V. Barker, D. O'Conner, "Expert Systems for Configuration at Digital: XCON and Beyond," *Communications of the ACM,* Vol. 32, No. 3, March 1989

[Bisiani] R. Bisiani, F. Lecoual, V. Ambriola, "A Tools to Coordinate Tools," *IEEE Software,* Vol. 5, No. 6, November 1988.

[Bimson] K. Bimson, L. Burris, "Assisting Managers in Project Definition: Foundations for Intelligent Software Management," *IEEE Expert,* Vol. 4, No. 2.

[LMSC-1] LMSC Internal Document DC-301, Expert Systems Preliminary Design Documents, August 4, 1989.

[LMSC-2] LMSC F255442, *SSE System Methods Manual,* Lockheed Missiles and Space Corporation, September 2, 1988

[LMSC-3] LMSC F255458, *SSE Preliminary Design Document* Lockheed Missiles and Space Corporation, July 17, 1989

[LMSC-4] LMSC F255472, *SSE Detailed Requirements Specification,* Lockheed Missiles and Space Corporation, July 17, 1989

[McKay] C. McKay, "Adding Knowledge base technology to Object Management Systems: Addendum 5.4.7," Software Engineering Research Center/High Technologies Lab, University of Houston at Clear Lake, no date.

[Samadi] B. Samadi, "TUNEX: A Knowledge-Based System for Performance Tuning of the UNIX Operating System," in *IEEE Transactions on Software Engineering,* Vol. 15, No. 7, July 1989.

[Taylor] F. Taylor *et al.,* "Foundations for the Arcadia Environment Architecture," ACM, 1988

[Tully] C. Tully, Editor, *"Representing and Enacting the Software Process," Proceedings of the 4th International Software Process Workshop,* ACM SIGSOFT Software Engineering Notes, Vol. 14, No 4, June 1989.

[Waterman] D. Waterman, *A Guide to Expert Systems,* Addison-Wesley, 1986