An Organon: Intelligent Reuse of Software Assets And Domain Knowledge

James Solderitsch

Valley Forge Laboratories
Unisys Defense Systems Inc.
Paoli, PA 19301
E-mail: jjs@prc.unisys.com

Abstract

Methods and techniques to produce large systems can be viewed on a continuum ranging from pure manual construction with no tool-based support to a highly evolved environment that greatly aids in the engineering of these systems. An organon represents one view of such an environment. The paper describes work underway at Unisys, partially supported by the DARPA/SISTO STARS Program, to advance the state-of-practice on the path to such an organon.

Keywords: reuse, software engineering, knowledge-based systems, construction, generation, domain model

Introduction

Unisys and its affiliates participating in the STARS[1] (Software Technology for Adaptable, Reliable Systems) program are keenly interested in understanding, developing (or acquiring) and applying technology to support the development of complex software systems based on a reuse perspective. This support is currently reflected in joint activities being pursued by all of the STARS Prime contractors (Boeing, IBM and Unisys) to produce a STARS Reuse Concept of Operations (CONOPS) and an Asset Library Open Architecture Framework which together will guide future STARS efforts in the area of software reuse.

The content of this paper reflects an approach to reuse that is in tune with the emerging STARS CONOPS. This approach emphasizes domain life-cycle activities which utilize a variety of techniques and methods. Too often, the word reuse is interpreted strictly to mean the reuse of code components, whether informally through ad hoc reuse by a programmer remembering and

[1] Unisys is supported in STARS under contract number: F19628-88-D-0031.

reusing a previously composed code fragment, or more formally by finding and retrieving code from a code library. Reuse should be understood in a larger context to mean the reuse of knowledge about the complex application areas (or domains) to be served by software systems for these areas. Engineering models that capture knowledge about these application domains can provide a vital basis for technology to support reuse. Such domain models themselves require their own enabling technology so that they can effectively infuse the application development process.

Vision

The mode in which domain models drive the development of systems can be viewed on a continuum ranging from the production of a system via a custom design and handwritten code to the automatic generation of a system from a high-level specification in a domain-specific language. In the past, knowledge gained from system development and deployment has been left in the heads of expert software developers, and has not been extracted and collected for corporate reuse. Thus there has been little movement on the maturity continuum. The development and application of domain models can accelerate this movement and thereby boost system quality and system development productivity. Unisys believes that a knowledge-based approach to the creation of domain models can provide a successful strategy for moving the engineering of software systems further along this important maturity scale.

An *Organon*[2] represents a vision of a knowledge-based environment able to support large-scale reuse based on the development and use of domain models. An organon[Sim88] will:

- be an interactive and evolving public storehouse of expertise and componentry serving particular application domains;
- effectively support wide-spectrum reuse including requirements, design and test cases;
- support a number of different points-of-view on system engineering including those of construction and generation; and,
- be a central repository of domain expertise that effectively combines people, plus emerging and maturing methods, plus supporting technology.

As domain models mature, the component-based construction of systems by human hands can evolve to a computer assisted, and eventually computer controlled, production of these systems from available components based on a domain-specific software architecture. When domain knowledge is sufficiently complete, work at the component level will give way to design and specification at the subsystem level with the corresponding system components produced automatically from templates and other library components. In the end, an organon can support complete generation of software subsystems from high-level, domain-derived specification languages. Such generation is possible for domains that are mature enough to admit the expression of parametric information in sufficient detail to allow the automatic production of code based on this parametric information.

[2] The dictionary defines organon as "...an instrument for acquiring knowledge; specifically, a body of methodological doctrine comprising principles for scientific and philosophical procedure and investigation". The ancient library at Alexandria was known as an organon.

This vision represents an advanced instantiation of the STARS Reuse CONOPS that is currently under development. The CONOPS proposes a Reuse Process Framework that incorporates families of related reuse supporting activities:

- reuse planning;
- asset creation;
- asset management; and,
- asset utilization.

An organon will enable and facilitate the production, management and application of software assets which will run the gamut from source code modules and test cases to domain-specific requirements to software generation subsystems.

Reality

Unisys is currently working to lay a foundation to move the state-of-practice further along the model-based maturity scale described earlier. Unisys is acquiring and producing a technology base to provide a machine readable and processable representation of domain models in a form which domain-specific tools may directly utilize and manipulate. The Reusability Library Framework[SWT89] (RLF), a step along the path leading to an organon, is a set of Ada knowledge-based tools (semantic network and rule-based systems that can be used in concert) to support the definition and manipulation of domain models.

The RLF is currently targeted to the production of domain-specific, knowledge-based[MC89] software library systems where the engineer is supported in his or her production of software systems using software assets contained in the library. Domain models are captured in sophisticated semantic networks which capture the crucial objects, operations on objects and relationships between objects within the domain. Library assets are catalogued and stored according to this domain model. In addition, the human engineer is aided by a rule-based inference system which provide guidance in exploring and understanding the domain model and the assets within the library. Domains addressed to date include Anti-Submarine Warfare (ASW), User Interfaces (in particular the use of Ada/Xt) and Ada benchmarks.

The RLF has also been used to develop a model-based tool utilization assistant (TUA) for the domain of document preparation. An early version of the RLF was used to produce an Ada Unit Test Assistant (Gadfly) [WSS+88] which contained a model of test heuristics and generated test plans based on parsing of Ada units and interaction with a human test engineer.

Broad objectives of the RLF project include:

- develop knowledge-based interfaces to repository (object) management systems;
- investigate the mapping between application domain and reuse technology (part selection, part composition, part generation);

- go beyond supporting retrieval of static parts to include program generation, system/software configuration, system/software testing and even system/software design and requirements analysis;
- support the basic integration of reuse technologies (knowledge-based and generation techniques); and,
- perform some applied research in domain analysis.

Recently, an X Windows-based graphical browser interface was developed for librarian applications built on top of the RLF.

Conclusion

Currently, the RLF attempts to support the middle of the continuum with the human engineer being guided by an RLF-based environment. In order to move farther along, additional support for generative approaches must be provided. The Fourth Annual Workshop on Reuse presents a stimulating opportunity to share the Organon vision with other researchers and practioners of reuse. Through attending and presenting at the workshop, it is hoped that the vision can be compared to current reality and be influenced by the recent work of others. In turn, the lessons learned and experiences of the RLF project and the STARS Reuse joint activities can help fellow attendees in their current work in support of reuse.

References

[MC89] R. McDowell and K. Cassell. The RLF Librarian: A Reusability Librarian Based on Cooperating Knowledge-Based Systems. In *Proceedings of RADC 4th Annual Knowledge-Based Software Assistant Conference*, September 1989.

[Sim88] M. Simos. The Growing of an Organon: A Hybrid Knowledge-based Technology and Metholodogy for Software Reuse. In *Proceedings of the National Conference on Software Reusability*, April 1988.

[SWT89] J. Solderitsch, K. Wallnau, and J. Thalhamer. Constructing Domain-Specific Ada Reuse Libraries. In *Proceedings, 7th Annual National Conference on Ada Technology*, March 1989. pages 419-433.

[WSS+88] K. Wallnau, J. Solderitsch, M. Simos, R. McDowell, K. Cassell, and D. Campbell. Construction of Knowledge-Based Components and Applications in Ada. In *Proceedings of AIDA-88, Fourth Annual Conference on Artificial Intelligence & Ada.* George Mason Univ., November 1988.

1 About the Author

Jim Solderitsch is technical lead for the portion of the Unisys STARS Reuse task being performed at Valley Forge Laboratories, Unisys Defense Systems. He previously served as chief programmer for the Reusability Library Framework (RLF) project. Before coming to Unisys in 1986, he was an assistant professor in the Mathematical Sciences department of Villanova University. He received his Ph. D. in Mathematics in 1977. His current research interests include knowledge-based approaches to software reuse and program generation techniques. He is a member of the ACM.