Methodology Fusion: The Next Step

Donald J. Reifer

Reifer Consultants, Inc.
P.O. Box 4046
Torrance, CA 90510
Tel: (310) 373-8728
Email: reiferd@source.asset.com
Fax: (310) 375-9845

Abstract

This paper discusses the need to consider solution space objects as we model the problem space and develop our requirements models. Several major efforts are currently underway to develop domain and information engineering methodologies which used different representations to model the problem space.

Most of these methods are being developed independent of one another. As a result, the impact of one on the other is not being addressed. In addition, their is a movement afoot to standardize data and exploit the use of reusable software assets. However, problem space modeling methods specifically avoid being constrained by such solution space objects.

The author proposes a three step process for resolving the problem. First, a high level domain and data model is developed independent of solution space constrains for a product-line. Next, the resulting models are used to evaluate tradeoffs as product or project-related functional, performance, interface and quality requirements are developed at a more detailed level. Finally, solution space objects are considered as design requirements as the requirements unfold and are specified. The overall method fuses or integrates several methods into one single overall solution approach. It makes sure solution space objects are considered as problem space specifications are formulated at the individual project or product level.

To be successful, the process proposed must represent its underlying models using a consistent representation. In addition, the criteria and rules which govern the acceptability of the interim products that result at each step of the process need to be made explicit along with the major tradeoffs that need to be considered. Finally, the methodology must be adapted to be compatible with an object, functional or hybrid model and the translation from one to another.

Keywords: modeling, methodology integration, methodology fusion, standard data, reusable software assets, domain engineering, information engineering

Workshop Goals: Broaden viewpoints, stimulate discussion and generate a few good ideas on how to deal with the challenge.

Working Groups: Domain analysis/engineering; Reuse and OO methods, Method fusion/integration

1 Background

Mr. Reifer currently serves as Chief Architect for the DISA/CIM Software Reuse Program. As part of his tasking, he is leading a joint Air Force, DARPA and DISA effort to provide DoD users with seamless access to a variety of reusable software assets via an interoperable network of heterogeneous reuse libraries. He is also helping to develop strategy and is providing technical leadership for other aspects of the DISA/CIM Software Engineering and Reuse programs.

Mr. Reifer is also serving as the Reuse Advocate to CAE-Link Corporation on the Space Station Simulator Project. This project is building a facility to train Shuttle astronauts and mission controllers in Station operations. His primary focus on this project is on ensuring that reuse opportunities identified are fully exploited.

2 Position - Fusion Method Needed

Reuse is a hot topic in the worldwide software engineering community. Many efforts are under way to improve both the state-of-the-art and state-of-the-practice because of the potential benefits. Many of these efforts are focused on exploiting domain specific architectures where the knowledge base of experience in the applications domain is used to create an architectural framework for building systems [1, 2]. However, the methods being used for domain engineering activities are not being integrated with those being used to perform other critical engineering tasks pertinent to the architecture. An integrated or combined methodology is needed to provide those developing systems with guidance on what activities to do when and how to know when they are done.

A method is defined in this context as consisting of [3]:

- An underlying model - those classes of objects represented, manipulated and analyzed by the method.
- A consistent notation - the means used to represent the products of the method.
- A process or ordered set of steps - the set of activities performed by the user of the methodology and their products.
- Guidance for applying the method - examples of products and the rules or criteria used to evaluate whether or not they are any good.

The purpose of this paper is to discuss a proposed integration or fusion method from these four points of view. This will be accomplished in the next Section. The concepts in this paper are still in the formative stages. Plans to take these concepts and refine them will be discussed in Section III.

A proposed methodology fusion concept which addresses the problem noted above is shown in Figure 1. Problem space modeling techniques are ordered and put together using the process shown in Figure 2. The approach taken is to consider solution space constrains as problem space objects are identified, fused together and evaluated using an iterative refinement process. The ultimate product of this method is a set of requirements for a project or product which is bounded within a domain by a number of constraints.

The object model seems suitable to serve as the underlying model for the proposed method because it permits classes to be specified and analyzed in a manner consistent with most of the popular

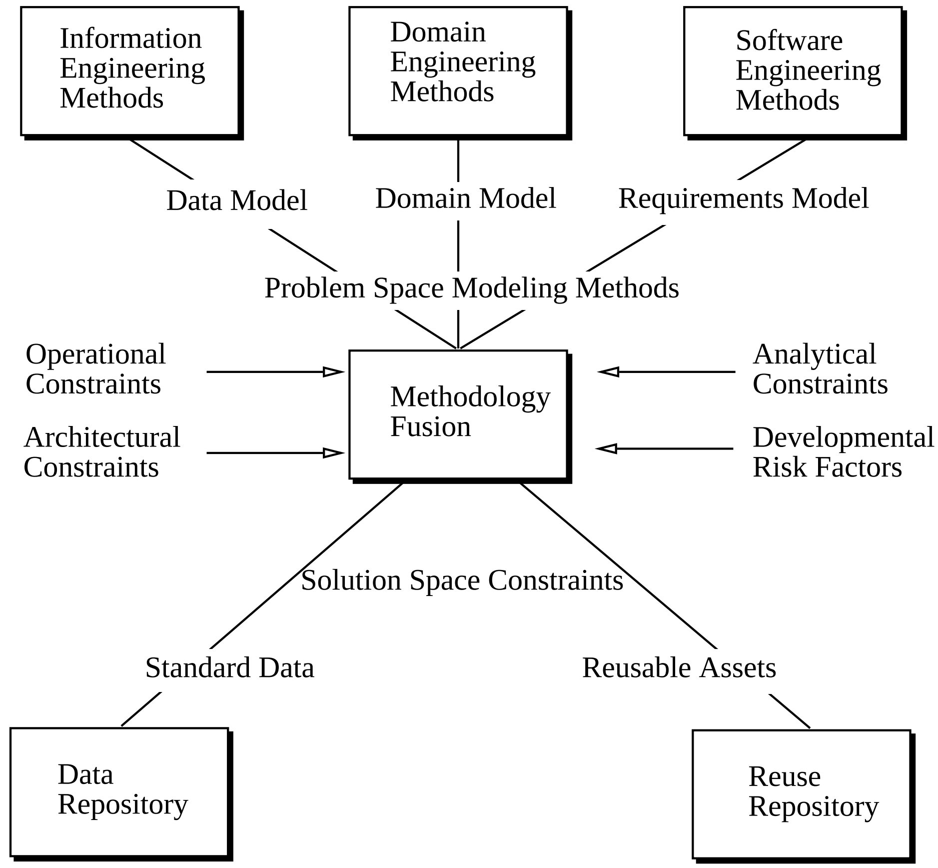

Figure 1: Methodology Fusion Concept

domain and information engineering approaches. An object model is powerful enough to permit the constraints being placed on problem space representations by solution space objects to be traded off and fully addressed.

A hybrid object-oriented notation seems appropriate because none of those analyzed seem to have the power to express the structure, associations and behavior of systems modeled using the fusion of techniques. Most of the weaknesses observed to date have been in the area of performance. To solve this problem, elements of Rumbaugh [4] and Booch [5] could be coupled with specialized notations based upon my experience on the Space Station Simulator project [6].

As illustrated in Figure 2, the process advanced is one of constrained optimization. First, a problem space model is developed free of constrains. Then, constrains are systematically added in step by step so that tradeoffs can be made and the resulting model optimized. The key concept to realize is that problem space models cannot be unconstrained when reusable assets and domain-specific architectures are being considered. Design requirements need to be specified to bound the problem space so that it can be easily mapped to solution space realizations.

Pilot programs will be used once the fusion method is specified to provide examples and develop the rules which will govern what to do when. This "do-a-little, learn-a-lot" philosophy has been used very effectively by many research programs to provide practical guidance to those actually trying to do the work.

Most of what has been proposed is nothing more than a paper tiger. The problem has been

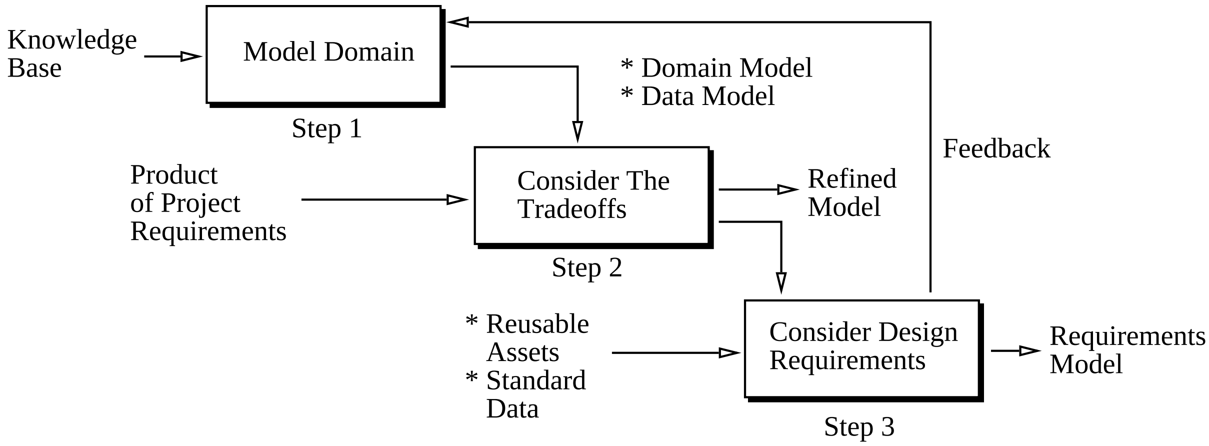

Figure 2: Modified Analysis Process

identified and an approach advanced to attack it. Within the next few months, we will mount a coordinated effort to put the method on paper.

Once specified, we will try it on one or more pilots to refine it based upon operational feedback. Examples will be developed along with rules. Assuming things work out, a fusion method handbook and training courses will be produced and the methodology will be transitioned into use via a defined and disciplined technology transfer process.

My purpose for discussing the proposed fusion method at the workshop is to get feedback. I am particularly interested in related work.

3 Comparison

No similar work other than that referenced is known when dealing with fusion of the techniques shown in Figure 1.

References

[1] C. Braun, "Software Reuse." ACM Professional Development Seminar Notes, November 1992.

[2] L. C. E. Mettala and M. H. Graham, "The Domain Specific Software Architecture Program," in *Proceedings DARPA Software Technology Conference*, pp. 204–209, April 27–30 1992.

[3] K. Kronlof, *Method Integration.* John Wiley and Sons, 1993.

[4] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-Oriented Modeling and Design.* Prentice-Hall, 1991.

[5] G. Booch, *Object Oriented Design With Applications.* The Benjamin/Cummings Publishing Company, 1991.

[6] W. Wessale, D. J. Reifer, and D. Weller, "Large Project Experiences with Object-Oriented Methods and Reuse," in *Proceedings of NASA/SEL Software Engineering Conference*, December 1992.

4 Biography

Mr. Donald J. Reifer is an internationally recognized expert in the fields of software engineering and management, with over twenty-five years of progressive experience in both industry and government. He has successfully managed major projects, served on source selections, wrote winning proposals and led project recovery teams. While affiliated with TRW, Mr. Reifer was the Deputy Program Manager for Global Positioning Satellite (GPS) verification and validation projects. As a Software Director with the Aerospace Corporation, Mr. Reifer managed over $800 million in software contracts for the Space Transportation System (Space Shuttle) Directorate. As a Project Leader at Hughes Aircraft, Mr. Reifer managed several major weapons system developments. Currently as President of RCI, a software consulting firm, Mr. Reifer directs efforts aimed at helping Fortune 500 firms (CAE-Link, Rockwell, Shell, TI, Westinghouse, etc.) and government agencies (DISA, NASA, etc.) to effectively manage large software projects, organizations and technology introduction. His current focus is on introducing software engineering and reuse technology.

Mr. Reifer is author of over 100 papers and several books on software engineering and management topics. He is also the author of the popular ASSET-R size and SoftCost-Ada cost estimation models. Mr. Reifer holds a B.S. in Electrical Engineering from Newark College of Engineering (NCE), an M.S. in Operations Research from the University of Southern California (USC) and the Certificate in Business Management from the University of California at Los Angeles (UCLA). He is an ACM national lecturer and a past member of the Board of Directors of the Ada Software Alliance. His many honors include being listed in Who's Who in the West, the NASA Distinguished Service Medal and the Hughes Aircraft Company Masters Fellowship.