product or process-oriented assets from any portion of the software life cycle (requirements, test plans, procedural knowledge). System architecture methods can be invoked here as supporting methods. For example, a current ARPA task is implementing an ODM supporting method for building GenVoca-style asset base architectures, utilizing a suite of tools developed as part of the STARS Army Demo project [7].

Tool Support. Organon Motives, Inc., in partnership with WPL Laboratories, Inc., and with partial support from an ARPA STARS grant and contribution from Loral Defense Systems-East, is currently re-engineering RLF to be more comprehensive and accessible. Originally implemented in Ada83, has been ported to Ada95 (utilizing the widely available GNAT Ada95 compiler), and re-architected to utilize World Wide Web- and Object Request Broker (ORB)-based technology to support a Web-accessible browser/editor interface. The new OpenRLF system will provide an extensible framework for building domain models and applications that navigate these models, providing key support technology for ODM-based domain modeling.

References


Mark Simos is president of Organon Motives, Inc., a software consulting and engineering research firm specializing in domain engineering methods (ODM), tools (RLF), and reuse strategic planning. Mr. Simos first learned the "craft" of building reusable components and application-specific languages as an applications specialist in the computer typesetting field. He has been an active participant in the reuse field since moving into research in the mid-80's, and has authored or co-authored many technical and working group reports and published papers on reuse and domain analysis. He was initial chief programmer on the STARS RLF project, is the principal developer of the Organization Domain Modeling (ODM) method, and co-authored several other STARS reuse documents including [CFRP93]. He holds an MSE from University of Pennsylvania. He is also a well-known songwriter and fiddler in traditional folk music circles.

Domain Analysis, Domain Modeling, and Domain-Specific Software Architectures: Lessons Learned

Will Tracz
Loral Federal Systems Owego, NY
tracz@lfs.loral.com

In 1992, the ARPA-funded Domain-Specific Software Architecture (DSSA) program Avionics Domain Application Generation Environment (ADAGE) project developed a domain analysis process for creating domain-specific software architectures.

This process was based on the STARS domain analysis process, developed by Ruben Prieto Diaz, and FODA (Feature-Oriented Domain Analysis), inspired by Shalom Cohen at the Software Engineering Institute. The
spiral/iterative ADAGE Domain Analysis process consisted of 5 phases:

1. Define the Scope of the Domain - Define what can be accomplished — emphasis is on the user's needs

2. Define/Refine Domain-Specific Elements (Concepts and Requirements) - Similar to Requirements Analysis — emphasis is on the problem space

3. Define/Refine Domain-Specific Design and Implementation Constraints - Similar to Requirements Analysis — emphasis is on the solution space

4. Develop Domain Models/Architectures - Similar to High-Level Design — emphasis is on defining module/model interfaces and semantics.

5. Produce/Gather Reusable Workproducts - Implementation/collection of reusable artifacts (e.g., code, documentation, etc.). What was new and different about this process was the explicit emphasis on separating functional requirements from implementation and design constraints.

In 1994, this process was selected as "best of breed" by Guillermo Arango in a comparison with 14 other Domain Analysis processes.

As a result of applying this process to the avionics domain, two key insights were gained: the need for integrated hypermedia tool support to represent domain knowledge and the need to support the capture and analysis of "operational requirements" or scenarios. In 1993, ADAGE developed a Tcl/Tk-based scenario-based domain analysis tool called DOMAIN (Domain Models All INtegrated). In 1994, a java-based version of this same tool was completed. Both tools had built-in support for object-oriented patterns that would form the micro-architectures of subsystems in the overall software architecture.

While DOMAIN remains a prototype, it is the authors conclusion that scenario-based domain analysis plays a pivotal role in requirements elicitation and refinement. Furthermore, the separation of problem space from solution space as called out by the separate steps in the process, is crucial in managing the complexity of most real-world domains and, in particular, conducive to creating layered architectures and OO patterns or frameworks.

Will Tracz is a senior programmer for the Loral Federal Systems Group Owego Division (formerly the IBM Federal Systems Company Owego Laboratory) where he is currently an ARPA PI (Principle Investigator) on the DSSA-ADAGE (Domain-Specific Software Architecture - Avionics Domain Application Generation Environment) Project. While with IBM, he was a member of IBM Corporate Reuse Council, IBM FSC Reuse Steering Committee, and editor of the IBM Corporate Programming Reuse Newsletter. Currently he is a member of the IEEE Computer Society Subcommittee on Software Reuse advisory board, editor of the ACM SIGSOFT Software Engineering Notes, and column editor for IEEE Computer. He also served as newsletter editor and chairman of ACM SIGMicro. Tracz has over 60 publications on software engineering, microprogramming, and software reuse. His current book, "Confessions of a Used Program Sales: Institutionalizing Software Reuse," published (1995) by Addison-Wesley, provides a roadmap for introducing reuse into an organization by first using anecdotes and analogies to illustrate the non-technical inhibitors to software reuse, then providing a detailed process for identifying and creating reusable software assets.

Making The Method Fit: An Industrial Experience in Adopting Feature-Oriented Domain Analysis (FODA)

Nathan S. Zalman
BNR Inc, Research Triangle Park, NC
nathanz@bnr.ca

BNR, Inc., has recently concluded a pilot study in the application of Feature-Oriented Domain Analysis (FODA) to the tasks of requirements analysis and reuse, and the re-engineering of legacy systems. The study was conducted at its Research Triangle Park, North Carolina, research and development facility. The project began as a collaboration involving BNR's Global Services Planning and Directory and Operator Services organizations, and the Application of Software Modeling group of the Software Engineering Institute at Carnegie-Mellon University.

As a result of this study and the core domain modeling competency gained from it, BNR, Inc., launched an organization to provide consulting services in Domain Modeling & Analysis to the Corporation, and to external customers. Current FODA projects encompass both software domains, such as telephony systems, and non-software domains, such as quality management systems.

FODA provides a method for performing domain analysis that highlights the identification, classification, and description of the user-visible features of families of software systems in a domain. Kang [Kang90] and Peterson [Peterson91] describe the initial content and development of the FODA methodology.