MORALE Project Final Report
For the Period July 1, 1996 to December 31, 1999

The MORALE project at the Georgia Institute of Technology is sponsored by the Defense Advanced Research Projects Agency as part of its Evolutionary Design of Complex Software (EDCS) program. The program manager is John Salasin. This document provides a report on project activities from its inception on July 1, 1996 until its completion on December 31, 1999. Its structure consists of an overview followed by descriptions of the component research efforts. The URL for the project web page is **http;//www.cc.gatech.edu/morale**.

I. Personnel

- Technical point of contact: Spencer Rugaber.
- Other MORALE faculty participants: Gregory Abowd, Ashok Goel, Mike McCracken, Melody Moore, Colin Potts, and Linda Wills. Professor Wills is a member of the School of Electrical and Computer Engineering; Professor Moore is a member of the Faculty of Computer Information Systems Department at Georgia State University to which we subcontract; all the others are in the College of Computing at Georgia Tech.
- Graduate students during the course of the project: Christian Ertmann-Christiansen, Dean Jerding, Mary Lefferson, Lilia Moshkina, William Murdock, Heather Richter, Pascal Schuchhard, Ian Smith, Kurt Stirewalt, and Robert Waters.

II. Description

The goal of the MORALE project is to facilitate the evolution of legacy software systems. Facilitation takes the form of improved quality by requirements validation, reduced risk via architectural evaluation and assessment, and increased productivity from maintenance and access to design rationale and from high-level reuse of architectural components.

MORALE exploits both high level (mission-oriented) requirements and existing software assets in order to predict the extent to which new system requirements perturb an existing system. Requirements are derived from mission-oriented goals and are explored through scenarios, and the existing system is analyzed in terms of its response to high-level events traces (scenario instantiations). These inputs are expressed in an architectural description language. The common representation enables predictions of architectural distortions as well as suggesting reuse candidates.

In essence, MORALE is a method for evolving software as well as tools to support the method. Project validation takes the form of case studies where the method and relevant tools are applied to characteristic problems.

III. Overview of accomplishments

The two main accomplishments of the project were the developments of the constituent MORALE methods and their supporting tools. The following table lists the methods and the tools that have been built to support them.

Table 1: Esprit de Corps Suite Status

Method Name	Tool Name	Description
MESA	SIRRINE	Adaptive design critic
MORPH	MORPH	User interface migration
Synchronized REfinement	Dowser	Domain-based source code and documentation browser
Infrastructure	REMORA	Reconcile multiple architectural representations
Synchronized Refinement	ISVis	Dynamic program analyzer for software architectures
SAAM	SAAMPad	Architecture rationale capture and access
Infrastructure	VisEd	Architecture visualization and editing
Infrastructure	ACME Server	Multiplatform architecture representation services
ScenIC	ScenIC View	Entry and management of requirements scenarios

A. Specific Accomplishments

- One of the PI's (Moore) obtained her PhD based on work accomplished on the MORALE project. Three other students supported by the project were awarded PhD degrees (Jerding, Smith, and Stirewalt). Furthermore, four students were awarded MS degrees (Lefferson, Moshkina, Schuchhard, and Waters).
- A case study of the MORALE software evolution method was performed at the NORTEL laboratory in Atlanta.
- Performed extensive validation of the MORPH technique and toolset using real legacy systems. The MORPH technique was shown to be as effective as a human analyst manually reengineering the code, and 100 times faster. MORPH also scales to large programs, successfully analyzing programs of 80,000 lines.
- A full-day tutorial on MORALE has been accepted for presentation at ICSM'99.

II. Interactions and collaborations

- Collaborated with Nortel to apply the MORPH technique to a current NORTEL product (RT1000).
- Spectra Research of Dayton, Ohio is commercializing the Dowser tool for the simulation domain (SimDowser).
- We have an informal collaboration with the Media-Doc group at USC.
- We are also actively collaborating with the ACME research group at CMU and with the ADL Toolkit group.

- In December, Heather Richter, Bob Waters and Gregory Abowd spent a day at the Software Engineering Institute discussing collaboration possibilities on the topic of live architectural rationale capture and architectural synthesis with researchers in the Software Architecture and Product Line Architecture groups, headed by Linda Northrop.
- ScenIC is closely aligned with the process of requirements and architectural negotiation, and thus we are collaborating with Profs. Boehm's and Horowitz's groups at Univ. of Southern California.

III. List of written publications in technical journals

Papers on MORALE constituent processes are listed in their subsections.

MORALE Project Final Report
Work on *Model-Based Redesign*

I. Personnel

- Ashok Goel
- Graduate students: J. William Murdock

II. Description

The model-based redesign component of MORALE is uses a function-oriented architectural description language to enable architectural reconfiguration of software systems as their mission-oriented specifications evolve. To support model-based redesign we have developed a methodology, MESA (Model-based Evolution in Software Agents) and a tool, SIRRINE (Self-Improving Reflective Reasoner Integrating Noteworthy Experience). SIRRINE provides a concrete instantiation of the MESA process.

III. Specific Accomplishments

- SIRRINE2 Version 1.0 was released. SIRRINE2 extends the theoretical foundations of MESA developed in earlier projects such as Autognostic and SIRRINE1. Unlike these systems, however, SIRRINE2 provides a principled, usable language for modeling software systems. In Autognostic and SIRRINE1, architectural models were implemented simply as data structures represented internally within the reasoning shells. While there was a specific account of the content of these models, there was not a coherent set of syntactic structures for presenting this content. Consequently, there was not a sharp distinction between writing the tool and using it. SIRRINE2 provides a concretely-defined language for specifying architectural descriptions as well as a set of access functions for manipulating these descriptions. Testing (both locally and by external users) has suggested that these features greatly facilitate use of SIRRINE2 by people who were not involved in its development and have only minimal understanding of its internal workings.
- Included in the SIRRINE2 Version 1.0 release was a module which supports ACME export of SIRRINE2 software models. By converting the SIRRINE2 models into the ACME architectural interchange format, we support integration of SIRRINE2 with other MORALE tools as well as with other software that supports ACME.
- Using the SIRRINE2 tool as a foundation, we have incrementally refined the MESA methodology.

IV. Interactions and Collaborations

- During the course of this project, we collaborated substantially with the EDCS-supported FAMILIAR project at Knowledge Evolution and Synquiry Technologies. Of particular value to our efforts on SIRRINE2 was insight from the work on FAMILIAR regarding the relationships between functional models and traditional software architectures with respect to ACME integration.

V. New Discoveries

1. The revised Task-Method-Knowledge language of SIRRINE2 is a new formalism for representing functional knowledge of software systems. Our experience has shown this formalism to be substantially more accessible than its predecessors.
2. The credit assignment mechanism in SIRRINE2 represents a substantial theoretical development. It is described in detail in the ASE'99 paper, referenced below.
3. We have found the model revision strategies in SIRRINE2 to be capable of small but significant and useful changes to existing software systems.

VI. List of Written Publications

1. J. William Murdock. "Semi-Formal Functional Software Modeling with TMK." Georgia Institute of Technology College of Computing Technical Report GIT-CC-00-05, 2000.
2. J. William Murdock and Ashok K. Goel. "An Adaptive Meeting Scheduling Agent." *Proc. First Asia-Pacific Conference on Intelligent Agent Technology (IAT'99),* Hong Kong, December 15-17, 1999.
3. J. William Murdock and Ashok K. Goel. "Towards Adaptive Web Agents." *Proc. Fourteenth IEEE International Conference on Automated Software Engineering (ASE'99),* Cocoa Beach, FL, October 12-15, 1999.
4. J. William Murdock and Ashok K. Goel. "A Functional Modeling Architecture for Reflective Agents." *AAAI98 Workshop on Functional Modeling and Teleological Reasoning,* Madison, WI, July, 1998.
5. Todd W. Griffith and J. William Murdock. "The Role of Reflection in Scientific Exploration." *Proc. Twentieth Annual Conference of the Cognitive Science Society (Cogsci'98),* Madison, WI, August 1-4, 1998.
6. J. William Murdock. "Prolegomena to a Task-Method-Knowledge Theory of Cognition." *Proc. Twentieth Annual Conference of the Cognitive Science Society (Cogsci'98),* Madison, WI, August 1-4, 1998.
7. J. William Murdock. "Modeling Computation: A Comparative Synthesis of TMK and ZD." Georgia Institute of Technology College of Computing Technical Report GIT-CC-98-13, 1998.
8. Ashok Goel and J. William Murdock. "Meta-Cases: Explaining Case-Based Reasoning." *Proc. Third European Workshop on Case-Based Reasoning,* Ouchy, Switzerland, November 1996; available in book form as *Lecture Notes in Artificial Intelligence 1168: Advances in Case-Based Reasoning,* Ian Smith and Boi Faltings (editors), Berlin: Springer-Verlag, 1996.
9. Ashok Goel, Eleni Stroulia, Zhong Chen and Paul Rowland. "Model-Based Reconfiguration of Schema-Based Reactive Control Architectures." *Notes of the AAAI Fall Symposium on Model-Based Autonomy, AAAI Technical Report,* B. Williams and Pandu Nayak (editors), October 1997; this paper was also presented at the IJCAI-97 poster session.
10. Eleni Stroulia and Ashok Goel. "A Model-Based Approach to Blame Assignment: Revising the Reasoning Steps of Problem Solvers." *Proc. National Conference on Artificial Intelligence - AAAI96,* Portland, Oregon, August 1996.
11. Eleni Stroulia and Ashok Goel. "PSM Modeling and Self-Adaptation in Robotics Applications." *Notes of the IJCAI-97 Workshop on Problem Solving Methods,* R. Benjamins and D. Fensel (editors), Nagayo, Japan, July 1997.
12. Eleni Stroulia and Ashok Goel. "Redesigning a Problem-Solver's Operators to Improve Solu-

tion Quality." *Proc. International Joint Conference on Artificial Intelligence - IJCAI97,* Nagoya, Japan, August 1997.

MORALE Project Final Report
User Interface Reengineering - MORPH

I. Personnel

- Melody Moore, Ph.D.
- Spencer Rugaber, Ph.D.
- Graduate students: Mary Lefferson and Lilia Moshkina (Georgia State University)
- Undergraduate students: Phil Seaver, John Brewster, Jon Preston (Georgia Tech)

II. Description

A significant problem in evolving large software systems is adapting the user interface to a new environment. Often, drastic changes in the user interface are inevitable, as in migrating a character-oriented system to a workstation with Graphical User Interface (GUI) capabilities. Since it has been estimated that half or more of the code in an interactive system is devoted to implementing the user interface, adding automated assistance to the task of reengineering user interfaces can substantially reduce the cost and effort for software migration and evolution.

The Model Oriented Reengineering Process for Human-computer interface (MORPH) technique and toolset automate user interface migration. Program understanding techniques are used to detect user interface components from code, which are then modeled using a knowledge representation. A transformation step uses inferencing to choose appropriate objects from a graphical toolkit to implement the user interface functionality. Design guideline information in the knowledge base addresses quality issues for the new interface. The current implementation of the MORPH toolset supports the automatic migration of character-oriented user interfaces to the Web by transforming them into HTML forms using CGI.

The MORPH technique and toolset have been validated empirically with a suite of information systems applications obtained from several military and industry sources. Measures include comprehensiveness of the user interface domain model, efficacy of the detection technique, and effectiveness of the representation in describing the application models. The MORPH toolset has been shown to be as effective as a human analyst manually reengineering the code, and over one hundred times faster at detecting and transforming user interface components.

III. Specific Accomplishments

- The prototype of the MORPH toolset was completed and demonstrated, including the following components:

  - *a flow analyzer,* consisting of a control flow analyzer, and a specialized data flow analyzer,

  - *a rule-based recognizer* containing a set of rules and heuristics to identify user interface dialogue component patterns from code,

  - *a transformation engine*, consisting of a translator and a transformer that loads application information into the MORPH knowledge base and composes queries to locate the appropriate GUI toolkit widget replacement, and

- *a generation technique* (currently manual) to insert the GUI widget replacements into the original legacy code.

- The MORPH user interface domain model was completed, formally specified, and implemented as part of the MORPH knowledge base. The domain model was validated for breadth and delineation via experiments with representing application user interfaces and GUI toolkits in terms of the domain model.
- Three GUI toolkits, HTML forms, Java AWT and Tcl/Tk, were implemented as part of the MORPH knowledge base to serve as the targets of transformation. The MORPH transformation process was validated by experiments transforming legacy code to the Tcl/Tk widget set. Findings showed that with a correct definition of a GUI toolkit MORPH is 100% accurate in finding replacement GUI widgets in legacy code.
- The MORPH toolset underwent extensive validation for performance and scalability, comparing it against manual analysis on both small and large programs. The MORPH toolset performed significantly better than manually reengineering the code (two orders of magnitude faster and with comparable accuracy). Some limitations were discovered and documented in preparation for future work.
- We performed a manual experiment to evaluate the feasibility of automating code generation for MORPH. We addressed issues such as architectural restructuring to support shifting dialogue initiative. The manual experiment supported the hypothesis that generation can be automated, because we were able to devise a set of heuristics for restructuring a sequential program to be event-driven. We successfully demonstrated these heuristics by migrating a real character-oriented legacy system to html forms and cgi using MORPH-suggested replacement widgets.

IV. Interactions and Collaborations

- *Spectra Research* - Consulted with Spectra several times to discuss incorporating MORPH ideas into their reengineering tools.
- *Nortel* - Collaborated with Nortel to apply the MORPH technique to reengineer user interface components of a current NORTEL product (RT1000), a sales force automation and tracking system.

V. Other pertinent information

GSU Subcontract

During the course of this contract, Melody Moore transitioned from her Senior Research Scientist position at Georgia Tech to accept a position as an Assistant Professor in the Computer Information Systems department at Georgia State University. The MORPH portion of the MORALE contract was placed into a subcontract between Georgia Tech and GSU. Dr. Moore’s students, Mary Lefferson and Lilia Moshkina, were been funded from this subcontract, as well as summer funding and equipment for Dr. Moore.

VI. Future Work

The MORPH technique lays the groundwork for many possibilities in reengineering and migration technologies:

- Generalization of MORPH techniques to detect architectural information in code.
- Adapt the MORPH technique for generalized API detection and transformation. This could be applied to other areas such as communications interface migration and potentially automated integration of stovepipe systems.
- Continue to investigate automated code restructuring, in particular to shift dialogue initiative (required to convert a structured program into an event-driven program).
- Use MORPH techniques to extract information for domain modeling.
- Incorporate dynamic analysis into the current MORPH static analysis technique in order to remove the limitations of static analysis.

VII. Publications

- Moore, Melody. "Rule-Based Detection for Reengineering User Interfaces." *Proceedings of the Third Working Conference on Reverse Engineering (WCRE),* IEEE Computer Society Press, Monterey, California, November 8-10, 1996.
- Moore, Melody. "Representation Issues for Reengineering Interactive Systems." *ACM Computing Surveys*, Vol. 28 No. 4es, Number 4, December 1996.
- Moore, Melody and Rugaber, Spencer. "Using Knowledge Representation to Understand Interactive Systems." *Proceedings of the Fifth International Workshop on Program Comprehension (IWPC'97),* IEEE Press, May 28-30, Dearborn, Michigan, 1997.
- Melody Moore and Spencer Rugaber. "Domain Analysis for Transformational Reuse." *Proceedings of the Fourth Working Conference on Reverse Engineering*, October 6-8, Amsterdam, The Netherlands, IEEE Computer Society, October 1997.
- Moore, Melody. *User Interface Reengineering*. PhD dissertation, GVU report number GIT-GVU-98-014, College of Computing, Georgia Institute of Technology, Atlanta GA 30332-0280, October 1998.

VIII. Papers Written but not yet Published

- Moore, M. and Moskina, L. "Migrating Legacy Systems to the Internet., submitted to the *International Conference on Software Engineering,* Limerick, Ireland June 00.
- Purao, S., Storey, V., and Moore, M. "Reconciling and Cleansing: An Approach to Inducing Domain Models., to be submitted to a journal in 2000.

Advisory functions

- Moore - Chair of the Reengineering working group at the DARPA PI Meeting in Dallas, October 1996.
- Moore - Invited participant in the NSF/DARPA Workshop on Software Engineering and Programming Languages, June 1996.

MORALE Project Final Report
Work on *Reverse Engineering*

I. Personnel

- Linda Wills and Spencer Rugaber
- Graduate students: Rich Clayton, Dean Jerding, Ian Smith, Kurt Stirewalt, and Robert Waters

II. Description

The goal of the reverse engineering component of MORALE is to extract architectural information from an existing system. The recovery of architectural descriptions forms the basis for assessing the impact of changes to the existing architecture. It also aids in identifying candidates for reuse in the evolving versions of the software and helps validate how well new requirements describe the existing software.

Beyond the traditional structural analyses provided by existing commercial and research tools, the reverse engineering component extracts behavior information from dynamic analyses of the message flow among architectural modules. In addition, the reverse engineering component of MORALE explored techniques for interleaving detection (determination of situations in which code implements more than one design goal or requirement) and domain-oriented analysis (taking advantage of application domain knowledge in understanding an existing software system). This approach has been validated by applying it to a case study of the evolution of the Mosaic web browser and by releasing the tools for dynamic program analysis and domain-oriented analysis to external users for evaluation.

After recovering various perspectives of the architecture during the recovery process, these perspectives are analyzed and reconciled to form a complete and consistent representation of architecture which is useful for further forward-engineering analysis activities in MORALE. This is a challenging task since information may be derived from high-level abstractions such as system-level actors in ScenIC or from low-level information such program event traces.

III. Specific Accomplishments

The reverse engineering portion of the MORALE project developed several tools in support of extracting and analyzing the architectures of evolving software systems. All of these tools have been publicly released.

- **ISVis**. We have constructed and released for public use an updated version of the visualization tool called ISVis (Interaction Scenario Visualization), for viewing and analyzing event logs of programs executing specific scenarios. The latest upgrade adds functionality to connect to an ACMEServer and export an ACME description of the recovered architecture. The new version of ISVis also exports formatted event data for use in the Balboa analysis tool. This enhances the use of the ISVis event traces for architectural localization activities.
- **ACMEServer**. ACMEServer is a tool which wraps the CMU ACME library with a communications infrastructure. This infrastructure allows multiple tools to manipulate the same ACME description. It is available for UNIX and Windows platforms.

- **VisEd**. The visualizer editor provides an extensible graph display and layout environment for ACME descriptions and other architectural tools. It executes on both UNIX and Windows platforms.
- **REMORA**. We developed a research prototype of REMORA (Resolution of MORALE Architectures) an automated techniques for the synthesis of architectural information derived from any potential source.
- **Dowser**. The Dowser, domain-based reverse engineering tool allows users to browse source code and program documentation using domain terms in addition to traditional source code browsing tools.

In addition to the tools described above, we have also accomplished the following:

- Our architectural recovery process has been refined in the context of a case study on the Mosaic web browser. This case study focused on the addition of user-configurable viewers to Mosaic, version 2.4.
- More broadly, to promote sharing of results and resources in reengineering, we are leading efforts to create and populate an international reverse engineering research resource repository. The repository is designed to contain tools, experimental results and case studies, bibliographies, and educational materials.

IV. Interactions and Collaborations

- Spectra Research of Dayton, Ohio is commercializing the Dowser tool for the simulation domain (SimDowser).
- We have an informal collaboration with the Media-Doc group at USC.
- We are also actively collaborating with the ACME research group at CMU and with the ADL Toolkit group.
- We have been collaborating with the TARDEC group of the Army Tank and Automotive Command (TACOM) in Warren Michigan. The collaboration was arranged by David Dampier and Glenn Racine of the Army Research Lab and Mike Saboe of TACOM. TARDEC has a charter to develop a generic Software Loader/Verifier for Army vehicles. We applied parts of the MORALE methodology to the domain analysis they are undertaking and have given them feedback on the requirements they are generating.
- EDCS collaborations: we signed a non-disclosure agreement with Xinotech and tried without success to install their tools for purposes of evaluation.

V. List of written publications in technical journals

1. Jerding, D. and Rugaber, S.. "Using Visualization for Architectural Localization and Extraction." *Proceedings of the 4th Working Conference on Reverse Engineering,* Amsterdam, the Netherlands, October, 1997. An expanded version of this paper has been accepted for publication in a special issue on reverse engineering of the *Science of Computer Programming*.
2. Clayton, R., Rugaber, S., Taylor, L., and Wills, L.. "A Case Study of Domain-based Program Understanding." *Proceedings of the 5th International Workshop on Program Comprehension,* pp. 102-110, Dearborn, MI, May 1997.
3. Abowd, G., Goel, A., McCracken, M., Moore, M., Potts, C., Rugaber, S., and Wills, L.. "Mission-Oriented Legacy System Evolution Through Architectural Recovery and Evaluation," *Proceedings of the ICSE-97 Workshop on Migration Strategies for Legacy Systems,* A. Cim-

itile, H. Muller, and R. R. Klosch (eds.), pp. 14-18, Boston, MA, May, 1997. Also available as Technical University of Vienna Technical Report TUV-1841-97-06.

4. Rugaber, S. and Wills, L. M.. “Creating a Research Infrastructure for Reengineering.” *Proceedings of the Third Working Conference on Reverse Engineering,* Monterey, CA, pp. 98-102, November 1996.
5. Richard Clayton and Spencer Rugaber and Linda Wills. “Dowsing: A Tools Framework for Domain-Oriented Browsing Software Artifacts.” May 1998, *Proceedings of the Automated Software Engineering Conference,* 1998.
6. Spencer Rugaber and Linda Wills. “Position Paper on Research Infrastructure for Reengineering”. May 28-30, 1997, *5th International Workshop on Program Comprehension*.
7. Richard Clayton and Spencer Rugaber and Linda Wills. “On the Knowledge Required to Understand a Program.” October 1998, *Proceedings of the Fifth IEEE Working Conference on Reverse Engineering’98*.
8. Spencer Rugaber and Jim White. “Restoring a Legacy: Lessons Learned”, 28-33. July-August 1998, *IEEE Software*, 4, 15.
9. Robert Waters and Gregory Abowd. “Architectural Synthesis: Integrating Multiple Architectural Perspectives”. *Proceedings of the Working Conference on Reverse Engineering (WCRE) 1999.*
10. Robert Waters, Spencer Rugaber, and Gregory Abowd. “Architectural Element Matching Using Concept Analysis”. *Conference on Automated Software Engineering (ASE’99)*.
11. Spencer Rugaber. “A Tool Suite for Evolving Legacy Software.” August 30 - September 3, 1999, *International Conference on Software Maintenance’99.*
12. Spencer Rugaber and Mark Guzdial. “Ectropic Software.” *Workshop on Software Change and Evolution (SCE’99),* Los Angeles, California, May 17, 1999.

MORALE Project Final Report
Work on *Software Architecture Analysis*

I. Personnel

- Gregory D. Abowd
- Others: Collaboration with P. Clements, L. Bass and R. Kazman on SAAM development; Pascal Schuchhard (U. of Delft) on implementation of first SAAMPad prototype; Heather Richter (Georgia Tech) on development of second SAAMPad prototype and SAAMPad case studies.
- Degrees granted: Pascal Schuchhard, Masters Degree, U. of Delft.

II. Description

This subarea is extending work on the Software Architecture Analysis Method (SAAM) developed by Abowd and SEI colleagues. The particular focus of SAAM is to provide a scenario-based method and ultimately some tools to support an impact analysis procedure for determining how proposed changes to an existing system impact the evolution of a systems software architecture. In relation to the MORALE project, this software architecture impact analysis provides a bridge between the scenario-generation efforts of the goal elaboration effort in requirements analysis and the architectural discovery work in reverse engineering.

The initial SAAM process was a labor-intensive process. There was the requirement for much brainstorming and negotiation to establish an understood architectural representation for a system and an agreed-upon set of new requirements, in the form of natural language scenarios, that would form the basis for discussing architectural impact. The approach being taken in MORALE is to provide computer-augmentation for some of the tasks of SAAM that were exclusively reliant on people. One of those tasks is the discovery of the architecture for the existing system. Reverse engineering work done in MORALE assists in the visualization and understanding of the software architecture. From the requirements side of MORALE we gain better methods for representing and discovering the set of scenarios that drive the mission-oriented evolution of the system under consideration.

The SAAM method has evolved through case studies done by Dr. Abowd and colleagues at the SEI. We continued this validation process by applying SAAM to systems that involve a number of other MORALE and EDCS researchers. The biggest contribution of the MORALE project with respect to SAAM is the development of prototype ubiquitous computing tools to support the natural capture of architectural rationale and provide better interfaces to search that rationale later. This work leverages off of existing research by Dr. Abowd in ubiquitous computing applications for automated capture and access of live experiences, funded in part by the National Science Foundation through the Classroom 2000 project.

III. Summary of Accomplishments

- Completed draft of SAAM process guidebook to aid in training for impact analysis.
- Completed MORALE case study applying SAAM to the evolution of the Mosaic Web browser.

- Completed first prototype of an environment for automating the capture and visualization of architectural rationale that arises in SAAM evaluations. This prototype, SAAMPad, was demonstrated at the DARPA EDCS meeting in July 1998 and is summarized in (Richter et al 99a).
- Conducted a live case study with Nortel in Atlanta in which a SAAM session was captured and used to develop an environment in which a SAAM document is associated with live video discussion of relevant parts of the architecture. The prototype was demonstrated at the DARPA EDCS meeting in June 1999.
- The prototype access environment involved the creation of a novel multiple-scale timeline control for visualizing large scale video archives (see Richter et al 99b)   A study of the effectiveness of this prototype architectural rationale access interface was completed and the results were submitted to ICSE (see Richter et al 99c)

IV. Interactions and Collaborations

- Dr. Abowd collaborated with SEI colleagues Clements, Bass and Kazman on issues of architectural evaluation.
- Initiated interactions with Atlanta branch of Nortel to do a SAAM evaluation of their RT-1000 call handling system. This involved three meetings that included tutorial presentations on SAAM and moderating and recording the SAAM evaluation.
- Gregory Abowd served on the ICSE'99 program committee.
- In June 1999, Gregory Abowd was an invited panelist at the Motorola Software Engineering Symposium discussing the topics of university/industry software engineering research and software architecture.

V. List of written publications in technical journals

1. [Kazman 96] Rick Kazman, Gregory Abowd, Len Bass, and Paul Clements. "Scenario-Based Analysis of Software Architecture." *IEEE Software,* November 1996, pp. 47-55.
2. Kazman 97] R. Kazman, P. Clements, L. Bass, G. Abowd, "Classifying Architectural Elements as a Foundation for Mechanism Matching." *Proceedings of COMPSAC 1997,* (Washington, D.C.), August, 1997, pp. 14-17.
3. [Brotherton et al. 98a] Brotherton, Jason and Gregory Abowd. "Rooms Take Note: Room Takes Notes!" *Proceedings of the AAAI Spring Symposium on Intelligent Environments. AAAI,* March 1998.
4. [Brotherton et al. 98b] Brotherton, Jason A., Janak R. Bhalodia, and Gregory D. Abowd. "Automated Capture, Integration, and Visualization of Multiple Media Streams." *Proceedings of the IEEE Multimedia and Computing Systems '98 Conference*, July1998.
5. [Abowd et al 98] Abowd, Gregory, Christopher Atkeson, Jason Brotherton, Tommy Enqvist, Paul Gulley and Johan LeMon. "Evaluating the Impact of Capture, Integration and Access on Education." *Proceedings of CHI'98.* April 1998, pp. 440-447.
6. [Richter et al. 99a] Heather Richter, Pascal Schuchhard and Gregory D. Abowd. "Automated capture and retrieval of architectural rationale." *Proceedings of the First Working IFIP Conference on Software Architecture, WICSA1*, San Antonio, Texas, February 1999.
7. [Schuchhard 99] Pascal Schuchhard. *Supporting the Software Architecture Analysis Method Through Automated Capture.* Masters Thesis Report, Technical Informatics, Faculty of Information Technology and Systems, Delft University, February 1999.
8. [Abowd 99] Gregory D. Abowd. "Software Engineering Issues for Ubiquitous Computing."

Proceedings of International Conference on Software Engineering, ICSE'99, Los Angeles, California, pp. 75-84, May 1999.

9. [Richter et al. 99b] Heather Richter, Jason Brotherton, Gregory D. Abowd and Khai Truong. "A multi-scale timeline slider for stream visualization and control." Georgia Tech GVU Center Technical Report, June 1999.
10. [Richter et al. 99c] Heather Richter, Gregory D. Abowd. "Automating the capture of design knowledge: a preliminary study." Georgia Tech GVU Center Technical Report GVU-99-45, December 1999. Submitted to ICSE'00.

MORALE Project Final Report
Work on *Mission Oriented Requirements*

I. List of personnel and degrees received

- Technical lead: Colin Potts
- Graduate students funded under EDCS program: Christian Ertmann-Christiansen.
- Others: Ana A. Anton, Lara D. Catledge, Idris Hsi, Reginald L. Hobbs (Army Research Laboratory).
- Advanced degrees awarded: A. A. Anton, Ph.D. (June, 1997) "Goal-Based Requirement Methodology"

II. Description and accomplishments

MORALE supports the specification of system requirements for new and revised features of evolving systems through the following three contributions of the *ScenIC* method and its tool *ScenIC View*:

- *Requirements Elaboration Strategy and Tools:* ScenIC supports a semantic model of system objectives and task goals and an elaboration procedure for refining objectives and tasks into automation requirements. ScenIC View supports the construction of ScenIC models and their incorporation into XML-based requirements and design documents. A major component of ScenIC, supported also by ScenIC View is the systematic analysis of the requirements in terms of obstacles that may block the accomplishment of system tasks. Using ScenIC View, it is possible to record obstacles and secondary tasks that defend against or mitigate the risks of obstacles.
- *Scenario Analysis Strategy and Tools*: ScenIC involves a strategy for deriving an episodic model of envisaged system behavior. Scenarios are structured in terms of the accomplishment of objectives or task goals and are constructed according to precise recommendations for requirements validation. ScenIC View integrates the episodic and semantic models so that consistency of terminology and structure is assured.
- *Requirements Decision Making and Rationale Model:* ScenIC is built on a rationale model or "working memory", supported by ScenIC View as a collection of annotations attached to semantic model or episodic model objects. Ongoing decision making is supported by tracking open issues. Recurring methodological issues can be optionally raised automatically by the ScenIC View 'apply heuristics' feature (formerly ScenIC Ride) which results in reminder annotations being attached to the objects in question (e.g. reminders to consider obstacles to tasks that have not yet been recorded). Six major methodological heuristics have been implemented in the release of ScenIC View.

ScenIC is documented in a guidebook and in a half-day tutorial, which has been given at several major software engineering and technology transfer conferences (CaiSE'97, CEIRE'98, ICSE'98 and ICSE'99). ScenIC and its immediate precursor, GBRAM, have been validated on several case studies, including a major intranet-based information system for a virtual corporation, the evolution of web browsing features, customer service, and cooperative time management and planning. In a controlled experiment, we have demonstrated the superiority of GBRAM/ScenIC to OMT for refining detailed requirements. ScenIC has formed the central method behind the European Community's CREWS project (Univ. Aachen, City Univ., London, and Univ. Paris, Sorbonne). A detailed mapping from ScenIC to REM/TMK has been documented and illustrated with the web browsing example.

ScenIC View is implemented as a Java application and incorporates the SAX XML parsing library. It requires the Java 2 virtual machine to run. Requirements models may be stored as persistent object collections or in ScenIC's XML-based interchange format SCML. Scenify, a prototype tool transforms stylized natural-language input into candidate semantic and episodic models.

III. Comparison with goals as expressed in the proposal

The accomplishments enumerated above correspond directly with the proposed goals. The name 'ScenIC' conveyed the primacy of scenarios (the ScenIC episodic model) in specifying features, but the resulting method involves tighter integration between the elaboration of goals and obstacles and the exploration of usage scenarios. There was less opportunity to evaluate empirically the claims made for ScenIC than was proposed initially.

IV. Consultative and advisory functions

Potts has consulted with Colorado Technical University on the requirements engineering content in their Masters in Software Engineering program. Potts was "opponent" (a form of external committee member) for Joachim Karlsson's Ph.D. dissertation defense at the University of Linkoping, Sweden. The dissertation concerned the assignment of priorities to requirements. Potts consulted with the Technical University of Tampere and served as opponent for the Ph.D. dissertation of T. Mikkonen on the topic of development of evolvable object-oriented system designs.

Prof. Potts has consulted with Software Productivity Solutions Inc., which was working on a parallel Rome-funded effort in scenario generation for early requirements identification. He has also consulted with GTE Laboratories in an internal project on the use of goals and scenarios in evaluating proposals for business process reengineering.

V. List of publications

1. C. Potts. Requirements Models in Context. Presented at the 3rd International Symposium on Requirements Engineering (RE'97), Annapolis, MD, January 7-10, 1997.
2. C. Potts and W. Newstetter. Naturalistic Inquiry and Requirements Engineering: Reconciling their Theoretical Foundations. *Proc. 3rd International Symposium on Requirements Engineering (RE'97),* Annapolis, MD, January 7-10, 1997. IEEE Computer Society Press, pp. 118-127.
3. C. Potts and I. Hsi., Abstraction and Context in Requirements Engineering: Toward a Synthesis, *Annals of Software Engineering 9:* 1-39, 1997.
4. G. Abowd, C. Ertmann-Christiansen, A. Goel, D. Jerding, M. McCracken, M. Moore, J.W. Murdock, C. Potts, S. Rugaber and L. Wills. MORALE: Mission-Oriented Architectural Legacy Evolution. *Proc. Int. Conf. Software Maintenance,* Bari, Italy, October 1997. IEEE CS Press.
5. Anton, A. I and C. Potts. "A Representational Framework for Scenarios of System Use."Requirements Engineering 1998 Vol. 3(3-4): pp. 219-241.
6. A. Anton & C. Potts. The Use of Goals to Surface Requirements for Evolving Systems. *International Conference on Software Engineering (ICSE'98),* Kyoto, April 1998.
7. Eastman, C., Hsi, I. and C. Potts. "Coordination in Multi-Organization Creative Design Projects" Georgia Tech, Graphics, Visualization and Usability Center Research Report 98-36. 1998.
8. Hobbs, R. L. and C. Potts. *Towards a Framework for Hypermedia Scenarios* Georgia Tech, College of Computing Research Report GIT-CC-98-06, 1998.
9. Potts, Colin. "ScenIC: A Strategy for Inquiry-Driven Requirements Determination*," Proceedings 4th International Symposium on Requirements Engineering.* Limerick, Ireland. June 7-11,

1999. IEEE Computer Society Press, pp. 58-65.

VI. Interactions at conferences, seminars, etc.

Conference presentations

3rd International Symposium on Requirements Engineering (RE'97), Annapolis, MD, January 7-10, 1997:
Keynote address. C. Potts. Requirements Models in Context.

Refereed paper. C. Potts and W. Newstetter. Naturalistic Inquiry and Requirements Engineering: Reconciling Their Theoretical Foundations

CHI'97 Workshop on Patterns of Interaction, Atlanta, GA. March,1997:
Invited paper. C. Potts, Patterns of Intent and Interaction.

REFSQ'97: Third Int. Work-shop on Requirements Engineering: Foundation for Software Quality. Barcelona, Spain: June 16-17, 1997:
Invited paper: C. Potts, Fitness for Use: The System Quality that Matters Most.

Workshop on Scenario Management, Schloss Dagstuhl, Germany. February 8-13, 1998.
Invited participation: C. Potts.

Proc. International Conference on Software Engineering (ICSE'98), pp. 157-166. Kyoto, Japan, April 19-28, 1998:
Refereed paper: A. Anton and C. Potts. The Use of Goals to Surface Requirements for Evolving Systems.

IFIP Working Group 2.9 (Requirements Engineering) meeting, Hawk's Key, Florida, January 1999
Invited presentation: Potts, C. Feature Evolution.

International Symposium on Requirements Engineering. Limerick, Ireland. June 7-11, 1999.
Refereed paper: Potts, Colin. ScenIC: A Strategy for Inquiry-Driven Requirements Determination

Refereed Minitutorial: "From "Requirements Engineering" to "Design for Usefulness"".

Conference tutorials:

Conf. Advanced Information Systems Engineering. Barcelona, Spain: June 18-20, 1997 C. Potts, Determining the Requirements for Evolving Systems.

International Conference on Software Engineering (ICSE'98), Kyoto, Japan, April 19-28, 1998. C. Potts. Requirements Analysis for Evolving Systems.

British Computer Society Conference on Industrial Requirements Engineering, London, UK. October 1998. Potts, C. Requirements Determination for Evolving Systems.

International Conference on Software Engineering (ICSE'99), Los Angeles, May, 1999. Potts, C. Tutorial: Inquiry-Driven Requirements Determination, IEEE CS Press.

Seminars and colloquia:

Software Engineering Distinguished Speaker Colloquium, University of Colorado, Colorado Springs, CO. April 13, 1998. C. Potts. Requirements Engineering in the Real World.

Georgia Tech, Graphics, Visualization and Usability Center Brown Bag seminar, October 1998. Eastman, C., Hsi, I. and C. Potts. “Coordination in Multi-Organization Creative Design Projects”

Invited colloquium, Department of Computer Science, Tampere Technical University, Finland. March 1999. Potts, C. “From “Requirements Engineering” to “Design for Usefulness””

Invited colloquium, Department of Information Systems, Univ. Jyvaskyla, Finland. March 1999. Potts, C. “Coordination in Multi-Organization Creative Design Projects”