MORALE Project Annual Report
For the Period July 1, 1998 to June 30, 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 July 1, 1998 until June 30, 1999. Its structure consists of an overview followed by descriptions of the component research efforts. Because of the overall limitation in length, the individual subsections are not included here and can be found at our web site (URL http;//www.cc.gatech.edu/morale/local/annual99.ps. In particular, the cumulative list of publications comprises thirty-six entries and would cause this document to exceed its three page limit.

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: Mary Lefferson, Lilia Moshkina, William Murdock, Heather Richter, 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 suggesting reuse candidates.

In essence, MORALE is a method for evolving software. As such, its ultimate validation is by customers who find that it effectively supports their efforts. Near-term validation takes the form of case studies where the method is applied to characteristic problems. MORALE also includes tool support, comprising the Esprit de Corps Suite of software evolution tools. Validation for the tools consists of evaluations by other EDCS participants and by industrial partners.

III. Specific accomplishments for the year

- The Esprit de Corps Suite of nine software evolution tools was demonstrated in Arlington at

the EDCS demo days. It comprised the following pieces.

Table 1: Esprit de Corps Suite Status

Name	Status	Description
SIRRINE	Released	Adaptive design critic
MORPH	Operational	User interface migration
Dowser	Operational	Domain-based source code and documentation browser
REMORA	Prototype	Reconcile multiple architectural representations
ISVis	Public release	Dynamic program analyzer for software architectures
SAAMPad	Prototype	Architecture rationale capture and access
VisEd	Released	Architecture visualization and editing
ACME Server	Released	Multiplatform architecture representation services
ScenIC View	Prototype	Entry and management of requirements scenarios

- 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.

IV. 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.

V. Plans for the remainder of the project

- Complete release of all tools. Further integration of the tools using ACME and XML.
- Complete Nortel case study.
- Final revisions to the MORALE guidebooks.

VI. List of written publications in technical journals

Papers on MORALE constituent processes are listed in their subsections.

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

I. Personnel

- Ashok Goel
- Graduate students: J. William Murdock

II. Description

The goal of the model-based redesign component of MORALE is the use of 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 been developing a methodology, MESA (Model-based Evolution in Software Agents) and a tool, SIRRINE (Self-Improving Reflective Reasoner Integrating Noteworthy Experience). The current version of our tool, SIRRINE2, provides a proof-of-concept instantiation of the MESA process.

III. Specific Accomplishments for the year

- We issued a Beta 0.5 release of SIRRINE2, and have made revisions to the tool based on the early beta testing results. 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. Preliminary beta testing 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 Beta 0.5 release was a module added this year 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 further refined the MESA methodology.
- We have begun work on a new system called REM (Reflective Evolutionary Mind). We intend for REM to build upon the theoretical and practical successes of the SIRRINE tool but to support substantially more powerful and interesting evolutionary redesign mechanisms. Of particular interest in REM is *proactive* system evolution: evolving systems in response anticipated new demands rather than waiting for a system failure to force evolution.

IV. Interactions and Collaborations

- We continued our efforts in the collaboration with 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. Plans for the Upcoming Period

1. We intend to issue a full Version 1.0 release of SIRRINE2 in the near future. This release will allow us to further validate our claim that SIRRINE2 provides a useful, usable instantiation of the MESA methodology.
2. We intend to continue to study integration of SIRRINE2 with other MORALE tools. The existing ACME support in SIRRINE2 should provide a technical foundation for exploring potential user benefits of combining SIRRINE2 with other tools in the Esprit De Corps Suite.
3. We will continue development of our new system, REM.

VI. List of Written Publications

1. J. William Murdock and Ashok K. Goel. A Functional Modeling Architecture for Reflective Agents. In AAAI98 Workshop on Functional Modeling and Teleological Reasoning, Madison, WI, July, 1998.
2. Todd W. Griffith and J. William Murdock. The Role of Reflection in Scientific Exploration, In Proc. of the Twentieth Annual Conference of the Cognitive Science Society (Cogsci'98), Madison, WI, August 1-4, 1998.
3. J. William Murdock. Prolegomena to a Task-Method-Knowledge Theory of Cognition. In Proc. Twentieth Annual Conference of the Cognitive Science Society (Cogsci'98), Madison, WI, August 1-4, 1998.
4. 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.
5. Ashok Goel and J. William Murdock. Meta-Cases: Explaining Case-Based Reasoning. In 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.
6. Ashok Goel, Eleni Stroulia, Zhong Chen and Paul Rowland. In 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.
7. Eleni Stroulia and Ashok Goel. A Model-Based Approach to Blame Assignment: Revising the Reasoning Steps of Problem Solvers. In Proc. National Conference on Artificial Intelligence - AAAI96, Portland, Oregon, August 1996.
8. Eleni Stroulia and Ashok Goel. Problem-Solving Methods and Applications in Robotics. In Notes of the IJCAI-97 Workshop on Problem Solving Methods, R. Benjamins and D. Fensel (editors), Nagayo, Japan, July 1997.
9. Eleni Stroulia and Ashok Goel. Redesigning the Problem Solving Steps: A Model-Based Approach. In Proc. International Joint Conference on Artificial Intelligence - IJCAI97, Nagoya, Japan, August 1997.

MORALE Project Annual Report
Work on *Model-Oriented Reengineering Process for Human-Computer Interface (MORPH)*

I. Personnel

- Melody Moore: Level of Participation: 40%
- Mary Lefferson
- Lilia Moshkina

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 chooses 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 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 over one hundred times faster at detecting and transforming user interface components than a human analyst.

The MORPH technique and toolset supports automated migration of legacy system user interfaces to graphical environments. The MORPH process that incorporates control and data flow analysis with language-neutral rules and heuristics to match user interface patterns in legacy code. The detected user interface components are stored in an abstract form in a knowledge base, and inferencing is used to support transformation of the components to replacements in a graphical widget set. 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.

III. Specific Accomplishments for the year

- Completed the development of the prototype MORPH toolset, including the detection engine

with a specialized data flow analyzer targeted for user interface components, a rule base and pattern recognizer, and a transformation engine to identify appropriate graphical replacements for the user interface components identified.

- Performed extensive validation of the MORPH technique and toolset using real legacy systems. The technique was evaluated for effectiveness of detection, effectiveness of transformation, performance, and scalability. 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.
- Performed a manual experiment in code generation and restructuring, addressing the issue of restructuring a sequential program to be event-driven. Successfully demonstrated heuristics for generation and by migrating a real legacy system to html forms and cgi using MORPH-suggested replacement widgets.
- Experiments with automating code generation demonstrated the feasibility of adding generation to the MORPH process as well as illuminating some issues for future work, such as architectural restructuring to support shifting dialogue initiative.
- We have begun initial experiments using MORPH to migrate applications to the web. This includes the development of a supporting knowledge base for storage of the abstract models and transformation to graphical components. Implemented transformation support for html forms, tcl/tk, and Java's AWT widget set.

IV. Interactions and Collaborations

- Met with Spectra Research to discuss incorporating MORPH ideas into their reengineering tools.
- Collaborated with Nortel to apply the MORPH technique to a current NORTEL product (RT1000).

V. Plans for remainder of the project

We plan to continue MORPH prototype development and validation, including the following activities:

- Continue work on automating migration to the web.
- Continue the work on generation and restructuring, experimenting with automating the manual process that has been demonstrated.
- Investigate the application of MORPH techniques to other problem areas, such as automated integration of communication interfaces.
- Complete work with NORTEL.
- Write final report.

VI. List of written publications in technical journals

1. Moore, Melody. "Representation Issues for Reengineering Interactive Systems", ACM Computing Surveys, Vol. 28 No. 4es, Number 4, December 1996.

2. Melody Moore and Spencer Rugaber. “Domain Analysis for Transformational Reuse”, *Proceedings of the Fourth Working Conference on Reverse Engineering*, Oct 6-8, Amsterdam, The Netherlands, IEEE Computer Society, 1997.
3. 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.

MORALE Project Annual Report
Work on *Reverse Engineering*

I. Personnel

- Linda Wills and Spencer Rugaber
- Graduate students: Rich Clayton, 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 is exploring 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 for the Year

We completed and released for public use infrastructure tool support for data interchange and visualization using the ACME architectural description language. This support includes:

- 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.

We have constructed and released for public use an update of the visualization tool, called ISVis (Interaction Scenario Visualization), for viewing and analyzing event logs of programs executing specific scenarios. This 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.

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. A preliminary version of this tool was demonstrated at the EDCS Demo Days in June.

The Dowser, domain-based reverse engineering tool was also demonstrated in Arlington. It allows users to browse source code and program documentation using domain terms in addition to traditional source code browsing tools.

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.

V. Plans for the remainder of the project

- ISVis Version 1.2, incorporating Java language analysis and event capture. We are also looking at a generalized static information capture algorithm.
- Continue development of REMORA and architectural information synthesis.
- Continue the case study at Nortel.
- Public release of the Dowser.
- Study interleaving at the level of design patterns to provide a rigorous characterization of how components satisfying multiple requirements are composed in software. (This is being funded as part of the NSF-supported, basic research component of EDCS, DARPA contract number E501.)

VI. List of written publications in technical journals

1. Jerding, D. and Rugaber, S., “Using Visualization for Architectural Localization and Extraction,” in Proceedings of the 4th Working Conference on Reverse Engineering, Amsterdam, the Netherlands, October, 1997. An expanded version of this paper has been invited for submission to 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,” in Proceedings of the 5th International Workshop on Program Comprehension, pp. 102-110, Dearborn, MI, May 1997. An expanded version has been invited for submission to a special issue on program comprehension of the Journal of Systems and Software.
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,” in 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. An updated version of this paper was also published in the Proceedings of the 5th International Workshop on Program Comprehension, Dearborn, MI, May 1997, where Rugaber participated on a panel on this topic.
5. R. Waters and S. Rugaber and G. Abowd. "Architectural Synthesis: Integrating Multiple Architectural Perspectives". Submitted to ICSE Conference 1999.
6. Richard Clayton and Spencer Rugaber and Linda Wills. "Dowsing: A Tools Framework for Domain-Oriented Browsing Software Artifacts". May 1998, To appear in the Proceedings of the Automated Software Engineering Conference, 1998.
7. Spencer Rugaber and Linda Wills. "Position Paper on Research Infrastructure for Reengineering". May 28-30, 1997, *5th International Workshop on Program Comprehension*.
8. Richard Clayton and Spencer Rugaber and Linda Wills. "On the Knowledge Required to Understand a Program". October 1998, To appear in the *Proceedings of the Fifth IEEE Working Conference on Reverse Engineering'98*.
9. Spencer Rugaber and Jim White. "Restoring a Legacy: Lessons Learned", 28-33. July-August 1998, *IEEE Software*, 4, 15.
10. Robert Waters and Gregory Abowd. "Architectural Synthesis: Integrating Multiple Architectural Perspectives". To appear in the proceedings of the Working Conference on Reverse Engineering (WCRE) 1999.
11. Robert Waters, Spencer Rugaber, and Gregory Abowd. "Architectural Element Matching Using Concept Analysis". Paper submitted to the Conference on Automated Software Engineering (ASE).

MORALE Project Annual 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.

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 people 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 the architectural impact. The approach being taken in MORALE is to augment 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 will assist in the visualization and understanding of the software architecture. From the requirements side of MORALE we will 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 will continue 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. Specific Accomplishments for the Year

In this past year, our efforts focussed on a second iteration on the SAAMPad tool in support of the live capture of architectural rationale. A live case study was conducted 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 access environment involved the creation of a novel multiple scale timeline control for visualizing

large scale video archives (see Richter et al. 99b).

Additional activities include:

- Gregory Abowd served on the ICSE'99 program committee.
- In June, Gregory Abowd was an invited panelist at the Motorola Software Engineering Symposium discussing the topics of university/industry software engineering research and software architecture.

IV. Interactions and Collaborations

- 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.
- Initiated interactions with Atlanta branch of Nortel to do a SAAM evaluation of their RT-1000 call handling system. This involved three meetings in October and December that included tutorial presentations on SAAM and moderating and recording the SAAM evaluation.

V. Plans for the remainder of the project

In the final six months of this project, we will complete a study of the effectiveness of our prototype architectural rationale access interface in collaboration with Nortel.

VI. List of written publications in technical journals

1. Rick Kazman, Gregory Abowd, Len Bass, and Paul Clements, "Scenario-Based Analysis of Software Architecture," IEEE Software, November 1996, pp. 47-55.
2. 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, Jason and Gregory Abowd (significant contributions by Brotherton and Abowd, principal authorship by Brotherton) Rooms Take Note: Room Takes Notes! Proceedings of the AAAI Spring Symposium on Intelligent Environments. AAAI, March 1998.
4. Brotherton, Jason A., Janak R. Bhalodia, and Gregory D. Abowd. (significant contribution by all, principal authorship by Brotherton and Abowd) Automated Capture, Integration, and Visualization of Multiple Media Streams. In the *Proceedings of the IEEE Multimedia and Computing Systems '98 Conference*, July1998.
5. 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, H., Pascal Schuchhard and Gregory Abowd (principal contributions by Abowd and Schuchhard, principal authorship by Abowd and Richter). Position paper to be presented at the First Working IFIP Conference on Software Architecture, February, 1999. Position paper to be published at http://www.bell-labs.com/user/dep/prof/wicsa1/.
7. Abowd, Gregory D. Software engineering issues for ubiquitous computing. Submitted to ICSE'99. August 1999.
8. [Abowd 99] Gregory D. Abowd. Software Engineering Issues for Ubiquitous Computing. In

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

9. [Richter et al. 99a] Heather Richter, Pascal Schuchhard and Gregory D. Abowd. Automated capture and retrieval of architectural rationale. In the Proceedings of the First Working IFIP Conference on Software Architecture, WICSA1. Position paper, San Antonio, TX, February 1999.
10. [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.
11. Two papers currently submitted for review to the Working Conference on Reverse Engineering (WCRE'99) and Automated Software Engineering (ASE'99) on the topic of architectural synthesis, co-authored with Bob Waters and Spencer Rugaber.

MORALE Project Annual Report
Work on *Mission Oriented Requirements*

I. Personnel

- Technical lead: Colin Potts
- Graduate students funded under EDCS program:
- Others: Reginald L. Hobbs (Army Research Laboratory), Idris Hsi.

II. Description

ScenIC (Scenario-Based Inquiry Cycle) is a strategy for determining and validating the requirements for evolving systems, the evolution of which is triggered by changes to the missions of the systems. It is an inquiry-driven approach, in which an incremental process of requirements representation, question-asking, decision-making and refinement leads to the systematic specification of new functionality and its allocation to architectural components. ScenIC forms the primary forward-engineering component of the MORALE framework, but utilizes architectural models obtained from reverse engineering and SAAM analysis.

ScenIC has three facets, corresponding to project "memories": semantic, episodic and working. The semantic facet consists of a set of standard representations for requirements-related information. It includes semi-formally expressed networks of objectives, tasks and obstacles, actors, including coarse-grained elements of system architecture, and allocations of tasks to actors. The episodic facet consists of illustrative episodes of actual or envisaged system use. Scenarios are built out of goal-directed episodes, during which instances of actors perform instances of tasks. Episodes may also illustrate the occurrence or mitigation of obstacles. The working memory facet includes semi-formal issues and reminders and drives the refinement of the representations. It thereby also serves to capture the rationale for decisions and assumptions. The principal style of raising and resolving working memory elements, though not the only one, is by walking through scenarios. ScenIC refinements consist of a number of requirements transformations, such as introduction of new elements, expansion of detail, realization of goals as actions, allocation of responsibilities to architectural or environmental components, and elaboration of simplified goals by introducing obstacles (real- world complications, faults or exceptions) and auxiliary requirements to handle them.

The ScenIC approach is being validated as part of the broader MORALE technology transfer and evaluation effort. In the short-term, we are constructing example specification fragments of evolving systems using ScenIC. So far, these have included web browsing, meeting scheduling, and air defense.

Because the ScenIC approach requires the use of lightweight tool support to manage the profusion of semi-formal requirements information that it produces, part of its validation involves the construction of prototype support technology. ScenIC View is an editing and analysis environment for ScenIC representations. It supports the creation, editing and annotation of semantic and episodic representations of the system's mission and requirements. Structured annotations form the ScenIC

working memory. ScenIC View is implemented in Java 2 and therefore runs on all platforms that support the Java 2 runtime environment. ScenIC Drive, a component of ScenIC View, is a prototype process guidance tool. For a selection of ScenIC method heuristics, and under user control, it automatically posts annotations to the working memory that remind the analyst of issues not yet raised.

III. Specific Accomplishments for the year

- An overview of the ScenIC method has been written to supplement the guidebook and was published as a conference paper.
- The half-day tutorial has been revised and presented at CEIRE'98 and ICSE'99. It will be abbreviated and incorporated into a full-day MORALE tutorial at ICSM'99.
- ScenIC View 1.0 implementation is complete. Release is pending on completion of user documentation.
- The prototype version of ScenIC Drive (0.1) has been integrated with the ScenIC View 1.0 distribution. A sample of six user-activated heuristics are supported in this version.

IV. Interactions and Collaborations

- Papers presented and papers published:

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

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

  Hobbs, R. L. and C. Potts. "Towards a Framework for Hypermedia Scenarios" Georgia Tech, College of Computing Research Report GIT-CC-98-06, 1998.

  Anton, A. I and C. Potts. "A Representational Framework for Scenarios of System Use." Requirements Engineering 1998 Vol. 3(3-4): pp. 219-241.

  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.

  Potts, C. "Feature Evolution" presented at IFIP Working Group 2.9 (Requirements Engineering) meeting, Hawk's Key, Florida, January 1999.

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

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

Potts, Colin. Tutorial: “Inquiry-Driven Requirements Analysis,” International Conference on Software Engineering, Los Angeles, CA. May 1999.

Potts, Colin. “ScenIC: A Strategy for Inquiry-Driven Requirements Determination,” Proceedings 4th International Symposium on Requirements Engineering. Limerick, Ireland. June 7-11, 199. IEEE Computer Society Press, pp. 58-65.

Potts, Colin. Minitutorial: “From “Requirements Engineering” to “Design for Usefulness””. Proceedings 4th International Symposium on Requirements Engineering. Limerick, Ireland. June 7-11, 199. IEEE Computer Society Press, p 189.

- Consultative and advisory functions:
  Colin Potts consulted with the Technical University of Tampere and served as opponent for the PhD dissertation of T. Mikkonen on the topic of development of evolvable object-oriented system designs.

- EDCS collaborations:
  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. Our approach to representing rationale defers to WinWin’s model of stakeholder “win conditions” and conflict resolution. WinWin, in return, obtains win conditions from ScenIC goals, and links to scenario and other analyses in the recording of stakeholder positions.
- Non-EDCS collaborations:
  Collaboration is ongoing with the EC CREWS project (Univ. Paris, Sorbonne, Univ. Aachen, Germany, and City Univ., London) and with Prof. Van Lamsweerde’s group (Univ. Louvain) and with Prof. A. Anton’s group (Univ. North Carolina) on the analysis of requirements for robust systems using goals, scenarios and obstacles, with Prof. Finkelstein’s software and systems engineering group at University College, London on the management of requirements rationale, and with the FEAST initiative of Prof. M.M. Lehman on evolutionary dynamics of complex software.

V. Plans for the remainder of the project

- During the remainder of the project, we will revise the ScenIC guidebook and issue the final version.
- The tutorial on requirements analysis for evolving systems that Potts presented at CAiSE’97, ICSE’98, CEIRE’98 and ICSE’99 will be further revised for future presentation at professional meetings and industry gatherings. Together with material on architecture recovery and analysis it will be presented as part of a full-day tutorial at ICSM’99 in Oxford, UK.
- ScenIC View 1.1 (lightly revised version of 1.0 with more complete installation and user documentation) will be released.
- ScenIC View 2.0 (greater usability and XML interchange format) will be released.
- ScenIC Ride 1.0 (more heuristics and user control) will be integrated with ScenIC View and released as part of ScenIC View 2.0
- A detailed mapping from ScenIC to REM/TMK will be documented and illustrated with an example.

VI. List of written publications in technical journals

1. C. Potts, Supporting Software Design: Integrating Design Methods and Design Rationale, in T. P. Moran & J. M. Carroll (Eds.), Design rationale: Concepts, techniques, and use. Hillsdale, NJ: Lawrence Erlbaum Associates, 1996.
2. C. Potts and L. Catledge. Collaborative Conceptual Design: A Large Software Project Case Study. Computer-Supported Cooperative Work: The Journal of Collaborative Computing, 5:414-445, 1996.
3. C. Potts and I. Hsi., Abstraction and Context in Requirements Engineering: Toward a Synthesis, Annals of Software Engineering 9: 1-39, 1997.
4. Anton, A. I and C. Potts. "A Representational Framework for Scenarios of System Use." Requirements Engineering 1998 Vol. 3(3-4): pp. 219-241.
5. K. Takahashi, C. Potts and K. Ota. Tuiqiao: A Hypertext Tool for Requirements Analysis (submitted to Journal of Automated Software Engineering).
6. C. Potts and A. Anton. A Model of Embodied Purpose: Goals, Obstacles and Action in the Description and Refinement of Desired Behavior (in preparation. Likely journals ACM TOSEM, Information Systems, IEEE TSE).
7. I. Hsi and C. Potts. The Evolution of Software Features and the Reverse Engineering of Goals. (in preparation).
8. C. Potts and I. Hsi. Feature Ecologies. (in preparation. Likely journals: CACM, HCI).
9. C. Potts. Software as Ascribed Ontology and Teleology. (in preparation for special issue of Information System Frontiers)
10. C. Potts. Three Reasons for Proactive Design Rationale. (In preparation. Likely journals: ACM TOSEM, Information Systems, IEEE TSE).
11. C. Potts and W. Newstetter. The Purpose and Context of Information Systems: Conflicts between Requirements Engineering and Organizational Practice. (In preparation. Likely journals: ACM Trans. Information Systems; IEEE Trans. Systems, Man and Cybernetics)