MORALE Project Annual Report
For the Period July 1, 1997 to June 30, 1998

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, 1997 until June 30, 1998. Its structure consists of an overview followed by descriptions of the component research efforts.

I. Personnel

- Technical point of contact: Spencer Rugaber.
- Other MORALE 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; all the others are in the College of Computing at Georgia Tech.
- Graduate students: Dean Jerding, Mary Lefferson, William Murdock, Heather Richter, Pascal Schuchhard, 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 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. Validation for the tools consists of evaluations by other EDCS participants and by industrial partners.

III. Specific Accomplishments for the year

Our plan for the second year of the project primarily concerned tool development. We have designed a suite of tools to support the MORALE process. The most mature of these is ISVis (Interaction Scenario Visualizer). In addition to ISVis itself, we have written extensive documentation and constructed a tutorial movie using Macromedia Director. ISVis has been released to our

EDCS colleagues and to the general public. We are currently working on a new release that provides ACME import and export. We have also been designing and prototyping other tools. In particular, we have built tools to support ScenIC (ScenICView), MESA (SIRRINE), MORPH, and SAAM (SAAMPad). With the exception of ScenICView, these tools were demonstrated at the Baltimore EDCS demos. In addition, we have developed and demonstrated two infrastructure tools—VisED, a software architecture visualizer and editor, and ACME Server, a multi-platform, multi-architecture, multi-tool manager (bus) for ACME architecture descriptions.

IV. Interactions and Collaborations

- We have participated in informal collaborations with the Win-Win group at USC, the Media-Doc group, also at USC, and with Knowledge Evolution.

V. Plans for the Upcoming Year

- Tools development: ISVis Version 1.1 release, SAAMPad and Remora Beta release, Release 1.0 of SIRRINE, MORPH, VisEd, ACME Server and ScenicView.
- Tool interoperation via ACME.
- Phase 2 case study. We have begun a case study with Nortel.

VI. List of Written Publications in Technical Journals

Papers on MORALE constituent processes will be listed in their sections.

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, nearly completed version of our tool, SIRRINE2, provides a proof-of-concept instantiation of the MESA process.

III. Specific Accomplishments for the year

- We have constructed a new version of the SIRRINE tool, SIRRINE2. 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. We expect these features to enable the use of SIRRINE2 by people who were not involved in its development and have only minimal understanding of its internal workings.
- Existing case-studies on the Mosaic WWW browser and a simple meeting scheduling system have been re-examined from the perspective of SIRRINE2. We have found that SIRRINE2 does provide model-based redesign capabilities, comparable and in some respects superior to those in SIRRINE1 when operating on the same agents. Thus we have developed preliminary evidence that SIRRINE2 combines its novel usability and conceptual coherence features with existing, proven autonomous reasoning capabilities.
- The above work has resulted not only in the development and preliminary evaluation of SIRRINE2, but also in the further refinement of the methodology called MESA (for Model-Based Software Evolution).

IV. Interactions and Collaborations

- We continued our efforts in the collaboration with EDCS supported FAMILIAR project at Knowledge Evolution and Synquiry Technologies. We have made considerable progress in studying the theoretical synergies between our TMK modeling framework and the ZD modeling framework used in FAMILIAR. A technical report detailing our work in this area has been released (see Section VI below).

V. Plans for the Upcoming Year

1. We intend to release a publicly accessible version of the SIRRINE2 tool in the near future. We hope that this release will allow us to validate our claim that SIRRINE2 provides a useful, usable instantiation of the MESA methodology.
2. We will begin development of a new system, 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. In particular, we expect REM to focus on two substantial theoretical issues which we feel SIRRINE and earlier work do not adequately address:
   - *Proactive Evolution*: Existing work, such as SIRRINE2, largely focuses on evolution in response to a system failure. We intend to expand this paradigm into situations in which evolution is driven by a new functional requirement rather than a failure to meet existing functional requirements.
   - *Interactive Systems*: We intend to provide more elaborate support for systems which interact extensively both with a user and with other computer systems (e.g., databases, WWW). While SIRRINE2 has been used on systems of this sort, the current version of the TMK modeling language does not allow explicit representation of system interactions. We have found that this significantly limits possibilities for reasoning about highly interactive systems.

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 User Interface Reengineering

July 1, 1997 - Jun 30, 98

I. Personnel

- Melody Moore
- Spencer Rugaber
- Mary Lefferson

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.

III. Specific Accomplishments for the year

- 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 recognizer* containing a set of rules to identify user interface dialogue components from code, and

  - *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.

- The MORPH 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.
- Two GUI toolkits, 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 was validated for performance and scalability, comparing it against manual analysis on a large program. The MORPH toolset performed significantly better (two orders of magnitude faster and with comparable accuracy). Some limitations were discovered and documented in preparation for future work.
- 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.

IV. Interactions and Collaborations

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.

V. Other pertinent information

GSU Subcontract

Melody Moore is now an assistant professor in the Computer Information Systems department at Georgia State University. The MORPH portion of the MORALE contract has been placed into a subcontract between Georgia Tech and GSU. Melody's student, Mary Lefferson, has been funded from this subcontract, as well as summer funding and equipment for Melody. We plan to continue the collaboration between the two universities as the MORALE project progresses.

Plans for next year

Next year we plan to continue MORPH prototype development and validation, including the following activities:

- Participate in a case study with Nortel to employ MORPH techniques to migrate character-oriented legacy applications for the PROTEL language.
- Investigate the generalization of MORPH techniques to detect architectural information in code
- Investigate the issues inherent in restructuring code to shift dialogue initiative (required to convert a structured program into an event-driven program).
- Continue work on automating migration to the web.

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. Moore, Melody and Rugaber, Spencer. “Domain Analysis for Transformational Reuse”, to appear in the journal *Science of Computer Programming,* special issue on reverse engineering, (pending)

MORALE Project Annual Report
Work on *Reverse Engineering*

I. Personnel

- Linda Wills and Spencer Rugaber
- Graduate students: Dean Jerding, 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 the 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.

III. Specific Accomplishments for the Year

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

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

We have an informal collaboration with the Media-Doc group at USC.

V. Plans for the Upcoming Year

- ISVis Version 1.1, incorporating ACME import and export, public release.
- Explore the use of the Balboa tool to enhance ISVis' ability to detect patterns in event traces.
- Perform a case study at Nortel.
- 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. Cimitile, 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.

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 SAAMPad prototype; Heather Richter (Georgia Tech) on SAAM development 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

- 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 annual DARPA EDCS meeting in July.
- Initiated talks with Nortel to do a large industrial case study of SAAM that will eventually lead to real-world experience with the automated tools such as SAAMPad.

- Presented two major conference papers (at IEEE Multimedia and CHI) on the general automated capture problem.
- Submitted paper to IFIP WICSA1 workshop that will be presented as position paper. Also submitted paper to ICSE on software engineering issues for ubiquitous computing that arise from experience in this project and other related ones at Georgia Tech.

IV. Interactions and Collaborations

- Initiated interactions with Atlanta branch of Nortel to do SAAM evaluation of their RT-1000 call handling system. This involved three meetings that included presentations on SAAM, SAAMPad and a tutorial on architectural evaluation.

V. Other pertinent information

- We are in need of further information on other software architecture manipulation tools. Rather than build our own architecture drawing tools, we would like to use the results of others in this area. To increase our ability to collaborate, we have been working to make our tools ACME-compliant.

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.

MORALE Project Annual Report
Work on *Mission Oriented Requirements*

I. Personnel

- Technical lead: Colin Potts
- 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, air defense, and, most recently, telephone agent performance monitoring. 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. We are currently implementing ScenIC View, an editing and analysis environment for ScenIC representations.

III. Specific Accomplishments for the year

- ScenIC has been codified and documented in a guidebook.

- A half-day tutorial has been prepared on using scenarios in requirements determination for evolving systems. This tutorial has been given at CAiSE'97 and ICSE'98.
- A pre-release prototype version of to ScenIC View, has been implemented.
- The ScenIC conceptual schema has been finalized, and a scenario interchange language, SCML-0 has been specified as an XML DTD.
- Scenify, an experimental tool to translate stylized natural-language input into SCML-0 scripts has been implemented.
- A case study has started with a telephone company to apply ScenIC to the monitoring of customer-service agents. The ScenIC guidebook has also been used in a graduate class on requirements engineering.

IV. Interactions and Collaborations

- Papers presented and papers published:

  C. Potts and I. Hsi., Abstraction and Context in Requirements Engineering: Toward a Synthesis, Annals of Software Engineering 9: 1-39, 1997.

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

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

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

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

- 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 PhD dissertation defense at the University of Linkoping, Sweden. The dissertation concerned the assignment of priorities to requirements.
- 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 and with Prof. Van Lamsweerde's group (Univ. Louvain) on the formalization of goals and the categorization of obstacles, with

Prof. Bubenko's group (Royal Inst. Technology, Stockholm) on the use of scenarios in the anticipation of mission changes, and Prof. Loucopoulos's group (Univ. Manchester Inst. Science & Technology) on the management of requirements rationale. Earlier work on GBRAM and the Inquiry Cycle is continuing in collaboration with Dr. Takahashi (NTT Multimedia Lab) and Prof. Anton (North Carolina State Univ.), and the results of this work will influence ScenIC. Further parallel projects are studying the role of fieldwork-based methods in predicting the requirements for evolving systems. These involve collaboration with Motorola Cellular Infrastructure Group and with Dr. Newstetter and Prof. Eastman (Georgia Tech.) and the supervision of two further graduate students.

V. Other pertinent information

- In the coming year, we will revise the ScenIC guidebook.
- The tutorial on requirements analysis for evolving systems that Potts presented at CAiSE and ICSE'98 will be presented at CEIRE'98 (London) and has been submitted for ICSE'99.
- Specific plans for work on ScenIC and ScenIC View are as follows.
  The heuristics in the ScenIC criticism component still require substantial organization and clarification, and external collaborations are working on the classification of obstacle types as a means to drive the exploration of scenarios and exception-handling behaviors.
  ScenIC View will be released. Methodological patterns and actions will be codified and implemented in ScenIC Ride (an active ScenIC guide and automated reminding agent). The current case study will be completed and another conducted.
- One possible problem is the unavailability of detailed, unclassified information about mission-oriented evolution in previous and ongoing projects. Our current MORALE contacts are promising, but continue to have a stronger technology-driven than mission-oriented rationale for evolution.
- A secondary problem is a continuance of the difficulty in recruiting graduate student assistance.

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. K. Takahashi, C. Potts and K. Ota. Tuiqiao: A Hypertext Tool for Requirements Analysis (submitted to Journal of Automated Software Engineering).
5. C. Potts & R. Hobbs. A Layered Conceptual Model and Hypertext Representation for Goal-Based Scenarios (In preparation. Likely journals: Automated Software Engineering).
6. C. Potts and A. Anton. Scenarios in Software Engineering: Matching Types and Representations to Purpose. (in preparation for special issue of Requirements Engineering on scenario management).
7. A. Anton, C. Potts and K. Takahashi. Refining Organizational Goals into System Require-

ments: An Intranet Case Study. (In preparation. Likely journals: IEEE Software; Comm. ACM).
8. 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)