

# Evolutionary Design of Complex Systems

## Open Technology for Software Evolution: Hyperware, Architecture, and Process

### Quarterly Report Vol 1, Number 1

Richard N. Taylor, David Redmiles  
Department of Information and Computer Science  
University of California  
Irvine, California 92697-3425

{taylor, redmiles}@ics.uci.edu  
<http://www.ics.uci.edu/pub/edcs/>

Voice: 714-824-6429      FAX: 714-824-1715

For the period: 01 November 1996 through 31 March 1997  
Contract: F30602-97-2-0021

Prepared for:  
DARPA/ITO  
Rome Laboratory, AFMC. USAF  
Office of Naval Research (ONRRO)

*Effort sponsored by the Defense Advanced Research Projects Agency, and Rome Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-97-2-0021. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.*

*The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency, Rome Laboratory or the U.S. Government.*

*Note: The first quarterly report represents a five month period, whereas future reports will cover only three month periods. The first period was adjusted in order to coincide with the University's fiscal calendar. The final quarterly report will cover a four month period.*

*Note: During the first three quarters of this contract we anticipate some overlap in topic, personnel, and publications with the Arcadia contract, under agreement number F30602-97-2-0021.*

# Technical Status Report

We are working on several technologies to aid in the effective evolution of complex software. In particular our research foci are:

- open, broad-based hyperprogram and active rationale support
- component-based architecture evolution technology, and
- multi-stakeholder process support

It is our belief that in order for these technologies to be truly useful they must be integrated together. In addition, they must be open and able to thrive in a heterogeneous, multi-language, and distributed world, populated by many commercial packages, for that is the world of real software evolution.

The key theme to this work is the creation of technology solutions that genuinely meet the needs of development organizations, package those solutions such that they can be effectively transitioned, and work with technology receptors to ensure that the technology is transitioned.

## 1. Ongoing Research and Development

### 1.1. Hyperware

In order to effectively evolve a large, complex system, users must have the power to quickly search, organize, visualize, analyze, relate, and integrate information about and related to that system. Using the lessons learned from our open hypermedia system, Chimera, and our work in the development of World Wide Web (WWW) technologies and standards, we are developing a suite of open hyperprogram services. This technology will support users performing large-scale software understanding and evolution tasks by providing quick access to relevant, distributed information in a version controlled context.

The focus of this work is to:

- leverage the complementary strengths of link server and distributed link systems,
- support configuration management of webs, and
- develop and utilize software agent technology for active rationale support and various hyperprogram support tasks.

#### 1.1.1. Link-Server and Distributed Link Technologies

Our previous work represents two different approaches to providing hyperprogram-like capabilities. Chimera embodies the link server approach while the WWW embodies the distributed link data approach. A link server provides greater control and coherency management of the hypermedia relationships and dependencies; distributed link data provides for better scalability and hyperweb robustness. Neither of these approaches are sufficient, when taken alone, to provide for the control, heterogeneity, adaptability, robustness, and unconstrained distribution necessary to support evolutionary systems. Our approach is to combine the strengths of each, in addition to an underlying support for configuration management and versioning, in order to solve the overall needs of a hyperprogram environment.

In the case of Chimera, our link server hypermedia system, we are exploring the distribution of link bases and the accessibility of link base information to Web agents. The distribution of link bases will allow a Chimera server to access a link base stored at a remote site. The ideal mechanism for gaining access to the remote link base is to name the resource via the WWW's URL mechanism and query the link base using a Web server. We are enhancing Chimera to export and import Chimera link bases and retrieve remote objects referenced by imported link bases on demand, via the Web. Providing Web agent access to information contained in a link base would enable the types of maintenance and analysis performed by these agents to be applied to the link base. The success of the distribution of link bases will be evaluated by demonstrating a hyperprogram web distributed across three sites.

From the Web side, initial areas of improvement focus on the Web's link model, storage of link information external to the linked resources, transfer of link information via HTTP, and correct interpretation of that external link information by Web clients. The Web's link model can be improved by allowing external links to specify both source and destination anchors for linked resources without having to embed the anchors within the resource data; these links can be stored separately on each Web server, manipulated using HTTP methods, and delivered as metadata. In order to demonstrate the benefits of these modifications, a Web client will be enhanced to use the link metadata to layout additional hypermedia anchors on the resource viewer, perform secondary actions called for by special link semantics, and identify third-party sources for additional link information.

This quarter has seen significant progress in the distribution of link base information with the alpha release of Chimera 2.0. This all-Java release was completed in late February. This version of Chimera is deeply integrated with the World Wide Web and provides distributed access to hypermedia information. Chimera 2.0 clients can take the form of both stand-alone applications as well as Java applets. URLs are used to reference Chimera hyperwebs and HTTP is used to establish connections between clients and servers. Chimera's components use HTTP to locate each other across the Internet. For a detailed description of the system, plus demos and software, see the project home page at <http://www.ics.uci.edu/pub/chimera/>.

Research on providing external linking capability for the World Wide Web has been taking place within the WebDAV activity. Central to the WebDAV approach are facilities for recording metadata, information about Web resources, associated with the resources themselves. Instances of metadata include the author, creation date, and character set of a document. Metadata can also be used to record links: a source URL and a set of destination URLs on a resource. Thus the WebDAV metadata capability provides the ability to record and access links on a document, without recording them inside the contents of the document.

The requirements for linking and metadata can be found in the WebDAV requirements document [WebD97]. An initial attempt at developing extensions to HTTP for linking and metadata can be found in [WebD97-2]. Both documents have been released as Internet Drafts by the Internet Engineering Task Force.

### 1.1.2. Configuration Management

A key focus of this work is configuration management support for hyperprograms. The foundation of our approach is the realization that the objects which comprise a software project and the

relationships between those objects may change independently. Hence, we allow changes to artifacts without changes to the hypermedia relationships, and vice-versa. In our approach, the configuration management of artifacts and the configuration management of links are separate.

A distinguishing factor between this approach and other hypermedia versioning systems to date is that the result of a link traversal between versioned objects depends on the user's current task. Three notable outcomes of such a link traversal are: traversal to the version of an artifact in a working configuration (which can support, e.g., software maintenance), traversal to the recent versions of an artifact (which supports dependency tracking), and traversal to a user-selectable version of an artifact (supporting, e.g., software understanding). Our proposal for hyperprogram configuration management supports all three link traversal outcomes. Integral to this capability is providing the user with sufficient visibility into the current versions of artifacts, the current contents of a configuration, and the expected outcome of link traversals, such that the users do not become disoriented. The goal of this research into the appropriate user interface for hyperprogram configuration management is the avoidance of users feeling lost in time (version space) and hyperspace.

We are addressing configuration management of hyperprograms in three phases: (1) configuration management in the link server approach, (2) configuration management in the distributed link data (HTML/WWW) approach, and (3) configuration management of the fusion of phases one and two. All three phases will provide capability for the three types of link traversal discussed above, along with the creation, manipulation, and visualization of configurations of versioned artifacts and their associated webs. Regression to a previous state of a configuration is an important benefit of hyperprogram configuration management, and will be fully supported. Due to the complexity of the data space being manipulated, research on intuitive user interface support for versioned hyperprograms will be an emphasis in all three phases.

Research on provision of versioning capability for the World Wide Web has been taking place within the WebDAV working group. Primary accomplishments to date include the development of a comprehensive set of requirements for performing distributed versioning of Web resources, and two initial attempts at developing extensions to HTTP for versioning capability[WebD97]. The requirements for versioning functionality can be found in the WebDAV requirements document [WebD97-2].

### 1.1.3. Active Agents

Active agents can be characterized as programs which are triggered by specific events in the system and assist the user in some task that must be done in response. Thus, an agent might remind a user to invoke a link generator after checking-in a change to a source code file, and then offer to invoke it automatically. Another agent might prompt for the rationale behind changes to a design specification. This type of interaction is key to supporting hyperprogram evolution. Agents help to keep the web consistent in the face of change. As these agents are expanded to work with teams of people, the collaboration and coordination of individual team members is enhanced. This coordination of the team enables increased understanding of large-scale projects by the team and allows them to perform effective large-scale software evolution. A primary motivation for the development of web agents is to aid the users of a large-scale hyperweb, so that they may better understand and evolve it.

We are exploring integration of active agents with the process environment, architectural support, and hyperprogram services described in this report. Effective use of components provided by the architectural services will enable rapid construction of new agents by reusing parts from existing agents. Finally, process environment services will allow these agents to be inserted into a process guiding a software evolution project, allowing them to be triggered in response to events during the process's execution.

A key enabler of integration of active agents to hyperware systems is found in the Chimera 2.0 beta release completed this quarter. Chimera 2 provides the ability to dynamically specify link semantics (i.e., the behavior of a link traversal). Link semantics are defined as Java classes, which will eventually be the basis of active agents for the evolution of hyperweb systems.

## 1.2. Software Architecture

Our research in the area of software architectures focuses on open, dynamic software architectures with design and active rationale support. We are also working to support hierarchical composition, where an architecture can become a single component in another, larger architecture, built incrementally. We intend to build upon initial results from the C2 software architecture project, which has produced a component and message-based architectural style. We will also integrate the development environment with both hyperprogram and process technologies.

### 1.2.1. A Style Supporting Open, Dynamic Software Architectures

We are leveraging our experience with C2 and the work on domain-specific software architectures (DSSA) to provide techniques and tools to aid architectural design of complex systems. Our system will provide an architectural framework from which a family of applications may be specified. The framework will provide domain independent architectural elements that may be incrementally refined to support domain-specific software architectures. As such, it will enable developers to effectively exploit domain-specific characteristics while reusing the high payoff, domain-independent features of the style. Our approach can be viewed as providing a step in the evolution of DSSA to general architectures.

In our view, a software architectural style must support components of varying granularity, written in different programming languages, and running in a distributed, heterogeneous environment without shared address space assumptions. Furthermore, multiple simultaneous users should be able to interact with the system. Several facilities will be supported to increase component reusability and substitutability across architectures. These facilities include limited architectural visibility by a component, translation of component interfaces, and externalization of component communication mechanisms. The integration of off-the-shelf components and communication mechanisms into the architecture must also be supported, as well as the use of multiple formalisms and software processes.

A novel aspect of our work is support for dynamic architectures. Dynamic architectures provide dynamism in-the-large similarly to the way dynamic languages provide dynamism in-the-small. A dynamic architecture allows modification of a system's architecture after the system has been built (i.e., before or during run-time). This includes addition of new components, upgrading of existing components, removal of unnecessary components, reconfiguration of existing components (e.g., performance tuning), and reconfiguration of the system architecture (e.g., modifying

the mapping of components to networked machines). The components may be specific or independent of the system domain. Supporting these tasks, even after the system has been built, affords system extensibility, customizability, and evolvability.

This quarter we devised a comparison and classification framework for architecture description languages (ADLs) and performed an extensive survey based on that framework. This survey was meant to highlight existing ADL capabilities and shortcomings, which would help in our further investigation of an ADL for C2-style architectures.

We designed and implemented a framework of abstract classes for modeling C2 architectures with concepts, such as components, connectors, communication ports, and so on. Among other things, the framework enables users to manually wrap OTS components. We have demonstrated this with two different implementations of the framework:

- the C++ implementation was used to wrap two different OTS user interface constraint managers: SkyBlue from the University of Washington, and Amulet user interface system's one way formula constraint manager from Carnegie Mellon University.
- the Java implementation was used to wrap an OTS WWW browser, JFox, and a persistent object package, JOP.

All of these components were successfully used in C2 architectures.

We are continuing research on techniques for supporting dynamic modification of software architectures to enable runtime evolution of mission- and safety-critical systems. We used the C2 Java framework as a basis for a prototype tool, ArchShell, which enables users to interactively modify architectures after system generation. ArchShell can currently also perform some modifications of architectures at runtime. An initial version of this tool was released this quarter.

### 1.2.2. Software Architecture Design Environment

One aspect of our approach to supporting the use of architectures in evolutionary software development is centered around Argo, an interactive software architecture design environment. Argo allows for graphical visualization and direct manipulation of architectural models. Architectural components and their relationships are modeled as connected graphs and presented via multiple, coordinated, customizable views. The architect can build, modify and analyze the architecture via Argo. The overall body of information used during software development will be stored in a heterogeneous hyperweb using the technology previously described. Architectural elements in the system will be linked to nodes in the hyperweb and those links will be some of the relationships visualized in Argo.

The architectural model of a complex software system may itself be very complex. Architectures abstract away details to allow for conceptual understanding of the system, but models rich in detail are needed for analysis. When the architect, developers, testers, or other stakeholders undertake changes to the system, they must have appropriate views to facilitate comprehension. The wide range of tasks performed by the various stakeholders demands a wide range of visualizations of the underlying rich architectural model. Appropriate views will only contain details that are relevant to the given task. Therefore, modifications and other types of tasks will be proportional only to the size of the task and not the size of the entire system or its related information space. Interactive customization of the design environment will allow for incremental definition of new views as

recurring comprehension needs are recognized.

Argo will also serve as an interface to analysis capabilities on the architectural model of the system. The architect will use Argo to specify what analyses are desired and to visualize the results by relating the analyses to the architectural model. Argo will contain agents that will analyze the architecture as it is being entered and present knowledge that is relevant and timely to decisions being made by the architect. Many types of analysis can be useful to the architect. We can categorize them as correctness, completeness, consistency, alternative, presentation, performance, and optimization. Correctness agents inform the architect when design rules or style guidelines have been violated. Completeness agents identify parts of the model that have been started but not completed. Consistency agents point out contradictions in the model. Presentation agents advise the architect on effective use of the notation and visualizations. Alternative agents remind the designer of alternatives to a given design decision. Performance agents predict the performance of some system aspects. Optimization agents suggest better values for some design parameters. This is an open-ended taxonomy, as other types of useful analysis will surely be discovered.

Finally, Argo will provide initial capabilities for push-button system generation based on the formal model of an architecture and implementation hints provided by architects. For example, Argo will be able to generate messaging code to enable interoperation of heterogeneous components, main procedures that instantiate and properly connect components, and Makefiles to further automate system generation.

This quarter Argo development concentrated on improving the functionality of the Java-based version of the environment. Support was added for multiple design perspectives and the Argo kernel was extended with domain models for decision process modeling and object-oriented design. Graph editing support was also improved using the Graph Editing Framework (GEF). GEF is a library of Java classes that make it easier to develop new applications that involve diagram editing and connected graph editing. GEF v03 was released this quarter. Both Argo and GEF are freely available at <http://www.ics.uci.edu/pub/arch/argo/> and <http://www.ics.uci.edu/pub/arch/gef/> respectively.

## 1.3. Process

### 1.3.1. Process Support

Effective evolution demands that all stakeholders be able to participate in a system's evolution. This requires involvement in evolutionary processes, visibility into product and process state, use of tools and interfaces that are effective for both technical and non-technical users, and effective mechanisms for facilitating customization and dynamic change. In order to support multiple and varied stakeholders, we are addressing issues pertaining to distributed processes and users, process system integration of 3rd party tools, customization of the process system, and support for dynamism.

We are creating enhanced process support capabilities in an open, distributed process system, focusing on:

- *Multiple Stakeholders*, who play varying roles in the evolution of a software development process and who require a diverse set of interaction mechanisms;
- *Distribution*, to support processes and projects which exist and evolve in different locations;

- *Integration*, to make use of existing complementary 3rd party and off-the-shelf software;
- *Customization*, to allow the process system to evolve to fit changing needs with minimal effort;
- *Dynamism*, to allow modification of a process while it is in progress.

**Multiple Stakeholders:** Because a wide variety of people must be supported during long-term evolution, we focus on the diverse needs of the process system's user population. Both managers as well as developers need the ability to extend and enhance processes, and processes must be understandable by both technically-oriented and non-technical users. As a process evolves, the roles of the various users may change as personnel change or as an individual's or an organization's needs change. In many cases, processes must be migrated from one organization to another, requiring unexpected evolution. Changes to the system can occur at many levels and by many different kinds of users.

One particular focus of our work will be to explore ways to take advantage of WWW interface capabilities for providing multiple views of a process as well as flexible viewers for end-user process support tools such as agenda browsers and task lists. In addition, we will integrate hyperprogram capabilities with the process support system, as described below.

To reduce the cost of end-user involvement of this evolution, we will explore the use of usability agents which provide a means for effective bi-directional communication between developers and users. These agents have the ability to identify mismatches between design data and users' expectations based on task models. When mismatches are found, agents may formulate rationale about the mismatch and determine a course of action, either notifying the appropriate developers or automatically initiating a process to correct the discrepancy if the information is available. For example, a usability agent could collect information about the usage of a distributed system, and may recommend architectural changes to either the customer, installer, or developer to optimize the system.

**Distribution:** Evolutionary product development necessitates a process system that supports distributed activities. Communication and coordination within the development process require the interactions of stakeholders at distributed locations, both before and after initial product deployment. Product users require access to previous development information, processes, and infrastructure which allows them to leverage off the previous development work and rationale. In addition, the differing contexts of multiple distributed developers and users necessitates the ability to coordinate potentially differing off-the-shelf technologies. For example, it might be necessary to take a prepackaged system and enhance it for specific needs. Access to the original development process will help guide the evolutions. Further, evolving a product repeatedly within an application family allows for leveraging one evolutionary process off of those that preceded it, saving time and effort.

We are working to provide support for distributed evolutionary processes by utilizing the hyperprogram technologies we create to facilitate communication and provide easy access to development information, processes, and tools both during initial development and throughout product evolution. Additionally, we are integrating with hypermedia technologies, including uniform resource identifiers (URIs) to associate distributed components, and the distributed execution mechanisms of Python scripts and Java applets. A potential validation of our distributed approach will be to allow remote users of an Endeavors release to perform upgrades to their installation through the use of a distributed process requiring remote access to artifacts (located at our site.)

The evolution of the system in independent remote contexts will be further improved by the enhancement of integration mechanisms allowing differing off-the-shelf tools to support differing user views and roles.

**Integration:** Process support systems must integrate diverse stakeholders and development environments. Flexible integration mechanisms provide the ability to evolve the support system in concert with product evolution. The ideal process support system affords product developers transparent integration with the appropriate external systems. Process support systems need to integrate with external system either through application programmatic interfaces (APIs) or through intermediate forms. We are enhancing the Endeavors integration mechanisms to use both methods without product developer effort or necessitating change to the process definition.

The resulting flexibility will allow product developers to evolve their tool sets for different users and phases of the product's life cycle. Integration with external hyperprogram technologies will demonstrate this capability by utilizing the API style of integration for automatic link creation between different product components (such as code unit and design documents) for different hyperprogram systems.

Further integration work is focused on enhancing the Endeavors process representation to include process rules and multi-site process negotiation by working, to the extent supported by the EDCS program, with technologies from Professor Gail Kaiser's group at Columbia University. Professor Kaiser's research program offers a very flexible and powerful rule-based process server (geared toward maintaining local constraints on process steps and details of sequencing among these steps), flexible transaction support (supporting software consistency maintenance), process evolution capabilities, and the treaty/summit negotiation approach. These capabilities would complement our process work, and in particular the object-oriented approach taken by Endeavors. The transaction and negotiation support would leverage both our hyperprogram work (particularly with distributed configuration management of webs) and process work.

**Customization:** Process support systems are complex software systems which are long-lived and must evolve over time. Capabilities built into the system to make customization efforts easier and less costly are essential for successful system evolution. Process support systems are often intended for use in a particular domain or application area, and end up being used in domains which are different from yet related to the original application area. The system must evolve to meet the changing requirements of the environment in which the system is to be used. Part of providing an open systems approach involves effective mechanisms for facilitating such customization. We will continue to explore and study architectural issues and styles that provide the appropriate levels of abstraction for efficient customization.

Because our approach to process technology involves multiple stakeholders, different abstractions for the model objects need to be maintained. Endeavors supports a multi-tiered object model allowing all stakeholders to customize the abstraction levels for behavior and data appropriate for their site. In this way, developers, customers, managers, and end-users are given the ability to configure the appropriate level of complexity and insight into the software and processes through object customization while still leveraging the evolution processes associated with the product.

**Dynamism:** The ability to dynamically change a process or workflow definition as well as the execution model at the time it is in progress to better fit changing requirements, availability of resources, and the applicability to the current work context is crucial for evolution. This is accom-

plished in Endeavors by reflexively implementing the Endeavors interpreter as a process activity which allows external sites to extend the interpreter in an object-oriented manner to fit their work context, guarantee a minimal level of process execution support at receptor organizations, and relieve the software provider from tracking external site configurations.

Endeavors is currently in the phase of improving its distributed infrastructure. The current model provides distributed persistence via http. A development effort is underway which will serverize the Endeavors foundation layer. Serverization will add full distributed processing between remote Endeavors systems and support for additional Internet compatible protocols (e.g. RMI and IIOP). It will also allow us to support file locking, remote handler invocation, and distributed process invocation.

This design requires a two point break in the system, first at the foundation level where object execution and manipulation take place, and then at the file (persistence) level where all objects and executables are stored. The latter break also introduces opportunities to leverage alternative storage solutions (e.g. alternative file systems and databases), improve performance and security, replication, and enhanced extensibility.

We are also currently working on improved user interface functionality, improved system-level interfaces, and automatic generation of Web-based guidance pages. In this quarter we have also used Endeavors to define our own development and release process and are using this process in our daily development efforts. The current Endeavors version alpha-1.0.6 is available for downloading on the release page <http://www.ics.uci.edu/pub/endeavors/>.

### 1.3.2. Active Agents

Expectation-Driven Event Monitoring EDEM provides developers with a platform for creating software agents to collect usability data and increase user involvement in the evolution of interactive systems. EDEM collects information that is currently lost regarding actual usage of applications to promote a more empirically grounded design process.

EDEM provides the following high-level functionality:

- A Multi-level event model: allowing agents to monitor actual usage of interactive systems at reasonable levels of abstraction.
- Contextualization: allowing contextual information to be captured when appropriate, e.g. system state, event history, and user feedback.
- Two-way communication: expectation mismatches are used as opportunities to establish dialogue and evolve mutual understanding between developers and users.
- Hypothesis-driven data collection: composite event detection is done on client-side and only relevant data is reported.

EDEM provides to following lower-level functionality:

- An interface for developers:
  - A GUI editor for creating agents.
  - Agent definitions may be saved and loaded from files.
  - Agents may be automatically loaded via URL.

- Displays are provided to help developers understand their application interface and design and debug agents.
- An interface for users:
  - A message window is provided to allow users to receive and react to agent messages.
  - An interface for requesting more information and responding to agent messages with feedback to developers is also provided.

This work spans the areas of process, hyperware, and architecture. Under process it is used as a means for gathering usability data under a software development process, under hyperware it collects usability data as one form of rationale, and under architecture it can provide feedback to direct improvements to system design.

This quarter the EDEM high-level design and a prototype implementation was completed.

## 2. Participants

### *Faculty:*

David Redmiles  
Richard N. Taylor

### *Research Programmers:*

Clay Cover  
Arthur Hitomi  
Kari Nies

### *Research Assistants:*

Ken Anderson  
Gregory Bolcer  
Roy Fielding  
David Hilbert  
Peter Kammer  
Michael Kantor  
Tony Kutscher  
Neno Medvidovic  
Peyman Oreizy  
Jason Robbins  
Shilpa Shukla  
Jim Whitehead

## 3. Notable Accomplishments and Technology Transition

### 3.1. Hyperware

In March, 1997, the Internet Engineering Task Force approved the charter of the World Wide Web Distributed Authoring and Versioning (WEBDAV) working group, making WebDAV an official working group of this Internet standards setting body. Jim Whitehead, U.C. Irvine, is Chair of this

working group.

A Chimera-related paper was presented at the 30th Hawaii International Conference on System Sciences [And96].

An alpha version of Chimera 2.0 was released in late February. This version offers a 100% Java-based solution for hypermedia integration of heterogeneous objects (for more information visit the project home page at <http://www.ics.uci.edu/pub/hyperware/>).

On January 13-17 there was a meeting of the WebDAV Design Team, comprised of participants from Microsoft, Netscape, Novell, U.C. Irvine, and U. Colorado, Boulder (Andre van der Hoek). At this meeting, the details of a preliminary protocol specification for performing distributed authoring and versioning capability via the World Wide Web were discussed by the participants. Based on this meeting, a preliminary WebDAV protocol specification [WebD97] was written and submitted to the Internet Engineering Task Force as an Internet-Draft in early February.

On January 27-28 an open meeting on topics of distributed authoring and versioning on the World Wide Web was held at the University of California, Irvine. This meeting provided a forum for discussion of the first draft protocol specification document for WebDAV, and was attended by 26 people. Participants provided significant discussion on all aspects of the draft, and provided compelling rationale for the need to comprehensively revise the draft. Participants also provided significant feedback on requirements issues, and a new draft WebDAV requirements document [WebDAV97-2] was submitted as an Internet-Draft in February.

Jim Whitehead gave a presentation entitled “World Wide Web Distributed Authoring and Versioning (WEBDAV):An Overview” at the IRUS-sponsored, Bay Area Roundtable (BART), March 14, 1997.

Gail Kaiser and Jim Whitehead published a paper entitled “Collaborative work: Distributed authoring and versioning” in IEEE Internet Computing [KW97]. Jim Whitehead also had an article on WebDAV published in ACM Standard View [Whi97].

The specification of the Hypertext Transfer Protocol (HTTP/1.1) was officially published as an Internet Proposed Standard in January [FG+97]. We have assisted in the implementation of HTTP/1.1 within the Apache server (Roy Fielding is a co-founder of the Apache Group and one of its core developers); several beta releases have been made during the quarter, with a final release anticipated before summer.

The specification of Uniform Resource Locators [BFM97] is under revision and will eventually progress as an Internet standard.

### 3.2. Software Architecture

Nenad Medvidovic presented a paper entitled “Reusing off-the-shelf components to develop a family of applications in the C2 architectural style” at the International Workshop on Development and Evolution of Software Architectures for Product Families, Las Navas del Marquis, Avila, Spain [MT96].

First release of ArchShell, a tool enabling users to interactively modify architectures, occurred in November.

On November 21, Jason Robbins presented a talk about Argo entitled “Extending Design Environments” to Collins Avionics (A division of Rockwell International) in Cedar Rapids, Iowa. Rockwell will investigate applying Argo’s approach and design environment infrastructure to specifying requirements for the commercial avionics.

On February 13, Jason Robbins presented Argo to Prof. Stuart Faulk of University of Oregon and Dr. David Weiss of Lucent Technologies. The presentation took place at the Rockwell International Science Center in Thousand Oaks, California. Dr. Joseph Lee of Rockwell was the host.

Version 3 of the UCI Graph Editing Framework (GEF), a Java class library that supports construction of graphical editors, was released. A demo, documentation, and source code are available at <http://www.ics.uci.edu/pub/arch/gef/>. Over 500 downloads of this software have taken place. GEF was winner of the “THE JARS October 1996 TOP 5% LISTING” award.

### 3.3. Process

“Endeavors: A Process System Integration Infrastructure” was published and presented at the International Conference on Software Process in Brighton, UK in December [Bol96].

“Endeavors: An Execution Infrastructure for Maturing Processes” was presented at the Conference on Software Process Improvement in Irvine, CA, January 1997.

Greg Bolcer gave a presentation entitled “Endeavors: Workflow for the WWW” at the IRUS-sponsored, Bay Area Roundtable (BART), January 10, 1997.

Alan Christie, Senior member of the technical staff at the SEI, discussed the possible use of incorporating Endeavors into their design for managing workflow between distributed CERT sites.

ISI - Defense Acquisition Services for High performance Electronic Commerce (DASHER). Endeavors is to provide a integration solution for DASHER, a project led by David Benjamin and Peter Will. There will be transfer of DASHER work to the DARPA/DLA-sponsored Advanced Logistics Program (ALP) by ISI’s DEALMAKER project where a workflow tool will be needed to provide a solution. A late phase of this effort calls for Endeavors to interpret and improve the flexibility of the systems as their workflow solution. Our technology will serve as a sophisticated replacement to their established state machine-based Order Management Service. A preliminary design has been established and is currently undergoing the development stage.

Endeavors is currently providing a project solution for Ray Licon, a Systems Analyst at Pacific Bell. Pacific Bell is currently investigating the use of Endeavors in other parts of the organization to aid in handling distribution and process control with their new merging partner, Southwestern Bell. A detailed Endeavors process has been created and developed and will be undergoing the testing phase shortly. Pacific Bell will benefit from the Endeavors project in three areas:

- The project will be used in an environment where many different people from all parts of the organization will test our technology. People of different organizational levels (end users, managers, and systems developers) will play as stakeholders in the project.
- The processes in this project are typically long (1 week to a month) and with a moderately high request rate. Many process instances will run concurrently with each other. Endeavors is a robust runtime tool that can track these processes and will provide a solution to PacBells process problems.

- Endeavors will help refine and control automation, integrate external process tools, and provide user mechanisms for development, evolution, and maintenance of processes.

William Gilbert, Interoperability Lab Manager, Network Products Division of Oracle installed and evaluated Endeavors for use at his lab for modeling software Quality Assurance processes. He has provided valuable feedback concerning the installation process, user interactions, and process modeling capabilities. Also, he is interested in experiments and integration with Oracle products we may implement.

Bahram G. Sagahri, Sedona Workflow/Tools Division of Oracle is interested in the design concepts and architecture of Endeavors. Endeavors influenced their decision to implement their next generation workflow product completely in Java and to model key aspects of their architecture after Endeavors. The Endeavors team will be meeting with Glenn Seidman, the architect for the Sedona Workflow product, to discuss further interactions.

Sun Microsystems. Sun Microsystems is evaluating using Endeavors as a process execution environment for their internal Java education program.

This quarter the EDEM website was created and EDEM java demos were made available.

In November an EDEM paper was completed and submitted to INTERACT'97.

## **4. Progress on Inter/Intra Cluster Collaborations**

### **4.1. Hyperware**

Engineers at the Northrop-Grumman B-2 Division are using Chimera to create and browse heterogeneous links over collections of engineering information. UCI is developing new client tool bindings to enable inclusions of new document types.

Prior to the IM Cluster Working Group meeting in Manassas, VA, several people from UCI met with Gail Kaiser's group at Columbia University to discuss possible collaboration between OzWeb, Chimera, and Endeavors. Columbia University has since integrated Chimera 1.2 with the OzWeb system. OzWeb can now invoke the Chimera server and clients to retrieve and display hypermedia documents within an OzWeb workflow. Columbia plans to integrate Chimera 2.0 at a much deeper semantic level in the future. We have discussed plans to use some OzWeb technologies within Chimera 2. In particular, the Rivendell tool server will replace the Chimera process invoker.

Gail Kaiser and Jim Whitehead published a paper on collaborative work on distributed authoring and versioning [KW97].

Andre van der Hoek, of CU Boulder, attended a WebDAV Design Team meeting at Microsoft, Redmond, Washington in January, 1997, lending his versioning expertise to this meeting. Andre has also produced a prototype WebDAV compliant web server on top of NUCM.

### **4.2. Software Architecture**

Northrop is examining whether/how to employ the C2 architectural style as the fundamental organizing concept for its prototype B-2 emulation and simulation environment. Based on our experi-

ence supporting OTS component reuse, we are developing technologies to assist in wrapping these tools as C2 components and encapsulating their communication within C2 connectors. Dynamic architectures will enable runtime interchange of functionally compatible emulators to support simulation at different levels of abstraction and granularity. Our architecture design environment will be used to support visualization, customization, manipulation, and analysis of the simulation environment.

Professor Karl Leiberherr (an EDCS PI in the Dynamic Languages cluster) of Northeastern University used GEF to develop the user interface of Demeter/Java.

Professor Michal Young (an EDCS PI in the High Assurance/Real-time cluster) of Purdue University has expressed interest in using GEF to develop the user interface to analysis tools.

Lockheed Martin has expressed interest in Argo and our rationale capture work.

### 4.3. Process

The University of Colorado, Boulder is using Endeavors as a visualization tool for their Balboa project. Alex Wolf and Antonio Carzaniga have implemented a translator that takes a graph generated by Balboa (in the AT&T DOT notation) and creates an Endeavors Network using the Endeavors API's. Balboa is a process discovery tool. It produces a file which lists the components of a directed graph. It is difficult to visualize the structure of the process this file describes. Endeavors provides a natural environment for display of the process and with the addition of Endeavors handlers, can be used to execute the process. UCI will take the lead in developing a dot-based export of Endeavors nets to Balboa.

The Endeavors group has acquired a copy of the Amber rule-based process engine from Gail Kaiser at Columbia University and are determining how to integrate it with the Endeavors infrastructure.

The EDEM project is planning Integration and code exchange with K. Narayanaswamy and M. Feather of Computing Services Support Solutions (CS3) (EDCS Rationale Capture cluster) and Teri Payton and Lyn Uzzle of Lockheed Martin (EDCS Information Management cluster).

## 5. Publications

The following papers submitted/published within this period explicitly reference this contract:

Jason E. Robbins, David M. Hilbert, David F. Redmiles. *Extending design environments to software architecture design*. Submitted to the International Journal of Automated Software Engineering. [RHR96-5]

Nenad Medvidovic, Richard N. Taylor. *Exploiting Architectural Style to Develop a Family of Application*. Submitted to the IEE Proceedings Software Engineering journal. [MT96-2]

## 6. Travel

**Table 1: Project Meetings/Conferences and Attendance**

| <b>Meeting</b>  | <b>Location</b>           | <b>Dates</b> | <b>Attendees *</b>             |
|---|---------------------------|--------------|--------------------------------|
| EDCS IM Cluster Meeting   | Manassas, VA              | Oct 28-30    | RT, DR, KN, GB, DH, JW         |
| WebDAV Requirements Review Meeting  | Xerox PARC, Palo Alto, CA | Nov 14-15    | RF, JW                         |
| Intl. Workshop on Development and Evolution of SW Archs. for Product Families | Avila, Spain              | Nov 18-19    | NM                             |
| Intl. Conf. on Software Process Improvement                                   | Brighton, UK              | Dec 2-6      | RT, GB                         |
| WebDAV Working Group Meeting  | Sunnyvale, CA             | Dec 4-6      | PO, JW                         |
| WebDAV Internet Engineering Task Force(IETF) Birds of a Feather Meeting       | San Jose, CA              | Dec 11       | JW                             |
| WebDAV Design Team Meeting  | Novell Orem, UT           | Dec 18-20    | JW                             |
| 30-th Hawaii Intl. Conference on System Sciences                              | Wailea, HI                | Jan 1-7      | KA                             |
| WebDAV Design Team Meeting  | Microsoft Redmond, WA     | Jan 13-17    | JW                             |
| Conference on Software Process Improvement                                    | Irvine, CA                | Jan 23-24    | RT, GB                         |
| WebDAV Working Group Meeting  | Irvine, CA                | Jan 27-28    | RT, RF, JW                     |
| EDCS Design Rationale Cluster Meeting   | Atlanta, GA               | Mar 5-7      | DR, DH, MK                     |
| 1997 Conference on Human Factors In Computing Systems                         | Atlanta, GA               | Mar 22-27    | SS                             |
| EDCS Info. Management Cluster Meeting   | Melbourne, FL             | Mar 24-26    | RT, DR, KN, KA, DH, PK, NM, JR |

*\*Initials for attendees are based on the list of participants given on page 12.*

## 7. Near Term Plans

### 7.1. Hyperware

Near term development plans for Chimera include integrating Netscape and FrameMaker to Chimera 1.2. We also plan to develop a C API to Chimera 2.0 to allow support for C as well as Java based clients. This should allow a quick port of the existing Shell, Emacs, and GIF viewer clients. We anticipate that we will be supporting Northrop-Grumman in their efforts to build an integrated demo for the July EDCS Program Workshop and Demo Days. We also plan to make additional enhancements and bug fixes to Chimera 2.0 as well as prepare a Chimera 2.0 demo for the July demos.

### 7.2. Process

For the next quarter, we plan to complete and release the serverization of the Endeavors foundation layer. We also hope to complete the automatic generation of Web-based guidance pages. In addition we plan to port Endeavors to JDK 1.1, taking advantage of new AWT features to further enhance our UI functionality.

The Endeavors team anticipates more collaboration with Sun Microsystems and plans on developing a process to drive Sun's web-based instruction delivery system.

Next quarter there are plans to redesign EDEM to allow generic monitoring of any set of objects - not just user interface objects - that conform to Java Beans Specification. We will also work toward a public release of the system. In addition, David Hilbert is working on a survey of existing approaches/technologies/methodologies for improving usability and gathering usability data.

The EDEM project plans to continue integrations with CS3 and Lockheed Martin and a demo will be prepared for the 1997 International Conference on Software Engineering in May.

### 7.3. Software Architecture

In the following quarter we plan to integrate ArchShell, a prototype tools for supporting dynamic architectures, with Argo, our architecture design environment. Users will be able to modify an executing system using Argo and those modification will be reflected in the running system.

We also plan to port the C++ KLAX video game demo to our C2 Java framework to demonstrate at the 1997 International Conference on Software Engineering in May.

A new interface for browsing multiple design perspectives and support for new critics are planned for Argo. There are also plans to port Argo to JDK 1.1

## References

- [And96] Kenneth M. Anderson. *Extending user-interface toolkits with hypermedia functionality*. Proceedings of the Hawaii International Conference on System Sciences. Wailea, Hawaii, January 1997.
- [Bol96] Gregory Alan Bolcer. *Endeavors: A Process System Integration Infrastructure*. Proceedings of the International Conference on Software Process. Brighton, UK, December 1996.
- [BFM97] T. Berners-Lee, R. Fielding, L. Masinter. *Uniform Resource Locators (URL): Generic syntax and semantics*. Internet Draft (work in progress). MIT, UC Irvine, Xerox, March 1997.
- [FG+97] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, T. Berners-Lee. *Hypertext Transfer Protocol - HTTP/1.1. Internet Proposed Standard RFC 2068*. UC Irvine, DEC, MIT, January, 1997.
- [KW97] Gail Kaiser, E. James Whitehead. *Collaborative work: Distributed authoring and versioning*. IEEE Internet Computing, Vol. 1, No. 2, March/April, 1997, pp. 76-77.
- [Med97] Nenad Medvidovic. *A classification and comparison framework for software architecture description languages*. UCI technical report (UCI-ICS-97-02).
- [MT96] Nenad Medvidovic, Richard N. Taylor. *Reusing off-the-shelf components to develop a family of applications in the C2 architectural style*. Presented at the International Workshop on Development and Evolution of Software Architectures for Product Families, Las Navas del Marquis, Avila, Spain.
- [MT96-2] Nenad Medvidovic, Richard N. Taylor. *Exploiting architectural style to develop a family of applications*. Submitted to IEE Proceedings Software Engineering, England, 1996.
- [MT96-3] Nenad Medvidovic, Richard N. Taylor. *A framework for classifying and comparing architecture description languages*. Submitted to the Sixth European Software Engineering Conference, to be held in conjunction with the Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE). September 22-25, 1997.
- [RHR96-5] Jason E. Robbins, David M. Hilbert, David F. Redmiles. *Extending design environments to software architecture design*. Submitted to the International Journal of Automated Software Engineering.
- [WebD97] WebDAV Design Team. *Requirements for distributed authoring and versioning on the World Wide Web*. WebDAV requirements document released by the Internet Engineering Task Force as an Internet-Draft. March 26, 1997.
- [WebD97-2] WebDAV Design Team. *Extensions for distributed authoring and versioning on the World Wide Web*. Preliminary WebDAV protocol specification document released by the Internet Engineering Task Force as an Internet-Draft. March 26, 1997.
- [Whi97] E. James Whitehead. *World Wide Web distributed authoring and versioning (WebDAV): an introduction*. ACM StandardView, March, 1997.