

AFRL-IF-RS-TR-2002-205
Final Technical Report
August 2002



A MEDIUM FOR DISTRIBUTED COLLABORATION IN INFORMATION INTENSIVE DOMAINS

Maya Design, Inc.

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. F300

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

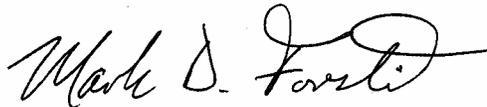
The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-205 has been reviewed and is approved for publication

APPROVED:



MARK D. FORESTI
Project Engineer

FOR THE DIRECTOR



MICHAEL L. TALBERT, Major, USAF
Technical Advisor
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE Aug 02	3. REPORT TYPE AND DATES COVERED Final Jun 97 – May 02	
4. TITLE AND SUBTITLE A MEDIUM FOR DISTRIBUTED COLLABORATION IN INFORMATION INTENSIVE DOMAINS			5. FUNDING NUMBERS C - F30602-97-C-0262 PE - 62301E PR - F300 TA - 01 WU - 00	
6. AUTHOR(S) David Bishop				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Maya Design, Inc. 2100 Wharton St. Pittsburgh, PA 15203			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency 3701 North Fairfax Drive Arlington, VA 22203-1714			10. SPONSORING / MONITORING AGENCY REPORT NUMBER AFRL/IFTD 525 Brooks Rd Rome, NY 13441-4505 AFRL-IF-RS-TR-2002-205	
11. SUPPLEMENTARY NOTES AFRL Project Engineer: Mark D. Forest, IFTD, 315-330-2233, forestim@rl.af.mil				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 Words) Visage Link extends Visage's capabilities to include collaboration among geographically distributed users. Visage Link facilitates collaboration among users by providing an architecturally sound framework for shared and private workspaces. In Visage Link, the set of data concepts in a Visage frame (a component that shows different views and arrangements of data elements) provides a shared domain of discourse. Like many collaboration and CSCW systems, Visage allows collaborating users to exchange and view data in shared frames. Visage Link also allows different users to view different representations of the same data, each in a manner that enhances their ability to contribute to a distributed problem.				
14. SUBJECT TERMS Collaboration, CSCW, Visualization, Polymorphism, Information-Centric, Direct-Manipulation, Visage			15. NUMBER OF PAGES 35	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

TABLE OF CONTENTS

Overview	2
The Research Agenda	3
The Research Approach.....	5
Technical Work Accomplished	8
U-form Serialization	8
Handheld Devices: Palm Pilot.....	9
Repository: Servers.....	10
Repository Components	12
Brief Overview of the VIA Repository	12
Library	13
Repository Server	13
Scalability	14
Repository: Visage	14
Repository: Clients.....	15
Integration: Rough-N-Ready.....	16
Integration: CoMotion.....	18
Attachments.....	20
Repository Configurations / Shepherds.....	23
Out-of the box Collaboration.....	25
Goals	25
Demonstrations	26
Repository Monitor	27
The Great Irony	28
Shared Marking	29
Conclusion	31

Introduction

This document, dedicated to the Visage-Link project (subtitled "A Medium for Distributed Collaboration in Information Intensive Domains") provides a forum for sharing project information including analyses, investigations, and demonstrations. It includes observations, positive and negative results, and design criteria established, procedures followed, and lessons learned.

Visage-Link is one of the components of DARPA's Collaboration, Visualization, and Information Management (CVIM) program. It is also one part of the Visage project, which is an advanced direct-manipulation, information-visualization environment currently under development at MAYA Design Group. Visage-Link extends Visage's capabilities to include collaboration among geographically distributed users.

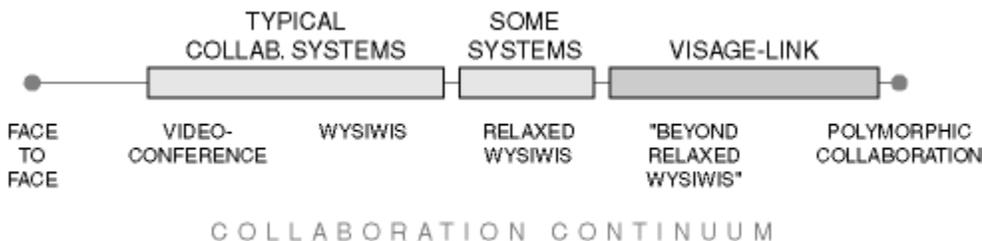
Visage-Link facilitates collaboration among users by providing an architecturally sound framework for shared and private workspaces. In Visage-Link, the set of data concepts in a Visage frame (a component that shows different views and arrangements of data elements) provides a shared domain of discourse. Like many collaboration and CSCW systems, Visage allows collaborating users to exchange and view data in shared frames. Visage-Link also allows different users to view different representations of the same data, each in a manner that enhances their ability to contribute to a distributed problem.

For example, two users may be visualizing a group of radio stations, but one may be using a map frame while the other is viewing a chart showing the output power of each station. Visage-Link gives the users the ability to manipulate the shared concepts by adding and removing them from the shared set, by using drill-down and roll-up navigation, and by dragging them to new visualizations.

Overview

Collaborative systems can be considered to occupy positions along a continuum from face-to-face meetings to abstract communication. Consider data visualization systems that lie along this continuum. At the one end, participants can all see the same visualizations at the same time; in fact, they're practically breathing the same air. At the other end, collaboration is markedly different: participants are free to construct their own representations, and their interactions must be deeper; more about the interpretation of the information than its presentation. This paper discusses the aspects of such a collaborative system, its practicality and usefulness, and some of the pitfalls discovered during the pursuit of it. The research system built to explore this polymorphic (multiple independent representations) collaboration is Visage-Link, an extension of the Visage direct manipulation information-centric environment developed by MAYA Design.

Tightly-coupled collaborative presentation is referred to as WYSIWIS (What You See Is What I See). Relaxed-WYSIWIS systems have been developed that loosen the coupling. Visage-Link resides in the space beyond relaxed WYSIWIS.



The Research Agenda

The Visage-Link project was designed to leverage the information-centric environment provided by Visage¹ in order to facilitate collaboration among geographically distributed users. The notion of frames was to be extended to include *shared frames* that allow the set of data concepts in a Visage frame to provide a shared domain of discourse. Not only can collaborating users see the same presentation of concepts, but they will be able to see different visualizations of the same concepts (*polymorphism*). This is significantly more powerful than simply sharing a single picture, since polymorphic displays can be tailored to users' roles and goals.

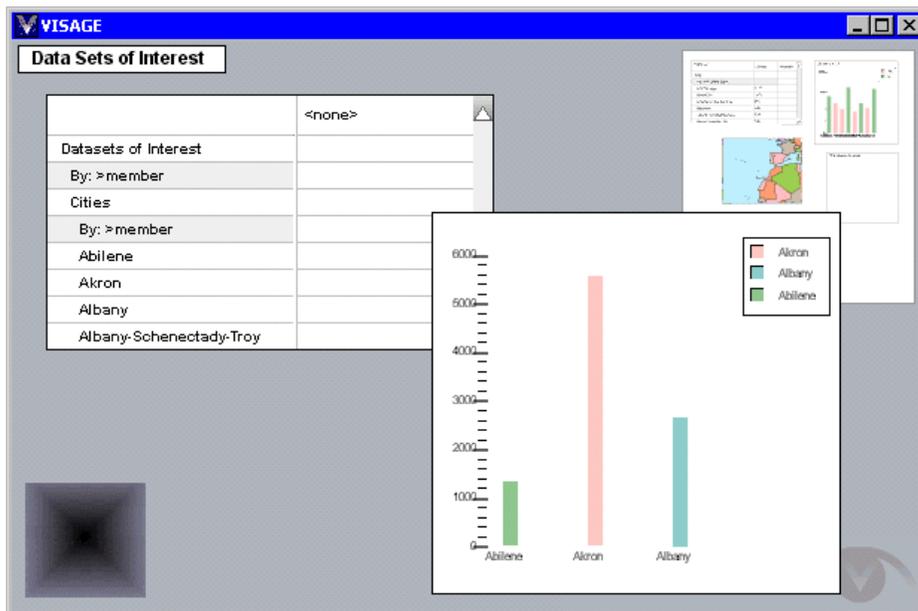


Figure 1 - A Visage Workspace, showing visualization frames in an information-centric environment. The Visage-Link project extended this environment to include collaborative capabilities.

¹ Roth, S. F.; Lucas, P.; Senn, J. A.; Gomberg, C. C.; Burks, M. B.; Stroffolino, P. J.; Kolojejchick, J. A.; and Dunmire, C. 1996. Visage: A user interface environment for exploring information. In Proceedings of Information Visualization, 3--12. San Francisco: IEEE.

Another way to view Visage-Link is as the next step in the paradigm shift defined by our previous work on the Visage project in order to facilitate collaboration among geographically distributed users. Enabling a smooth continuum from synchronized shared views (supported by classical collaboration systems) to coordinated views containing different presentations and visualizations of the shared data space was a primary goal. *Polymorphic Collaboration* is the Holy Grail of the Visage-Link project, and is its most distinguishing characteristic when compared to other collaborative systems. The project was designed to build on this approach to enable multi-platform collaboration, including palmtop and handheld displays. Developing user-configurable sequences of information flows using asynchronous scripting techniques, and using Visage's visualization capabilities to provide top-sight of the global workflow were additional components of the project.



Figure 2 - A high-level Visage-Link goal is to allow two users to collaborate regarding the same information but use views tailored to their individual needs (Polymorphism)

The Research Approach

The long-term Visage-Link roadmap required that several things coalesce into a coherent system. First, every Visage element must have a corresponding concept in a repository, including those elements that make up the Visage system itself (i.e., frames, scripts, and visualizations). Achieving Visage-Link objectives hinges on building a robust, scalable, distributable repository to contain data concepts (called U-Forms in the Visage vernacular). Visage's unique data model requires that data concepts be nothing more than uniquely identified bundles of attribute-value pairs. (They are called U-Forms because they are related to Michael Dertouzos' E-Forms, but have universally

unique IDs.) Our approach is to build a shared repository which communicates using a well-documented message format, is robust, and is massively scalable. Because most available database solutions (including object databases) do not allow real-time schema changes, the U-Form repository must be built from the ground up (it is essentially schema-free).

The Repository is the shared data store, built with the Visage data model in mind, that makes collaboration in Visage-Link possible. Second, Visage must be made to adhere to our Repository-Executor-Transducer (R-E-T) model; the crucial salient point being that *transducers* present data concepts to the user as perceivable elements (usually rendering them to a display). Third, repositories must be distributed, to allow multi-user access across varying levels of connectivity, differing topologies, and varying levels of service.

Repositories are important because they allow visualizations to be built from the ground up using only data concepts. This is crucial to Visage-Link because it allows collaborators to share not only data, but the information necessary to recreate visualizations as well (and recreating them is far more powerful than simply sharing screens like more typical teleconferencing-like solutions). Adhering to the R-E-T model allows the visualizations to be recreated – it is exactly the existence of the executor and transducers that allow shared data concepts to be presented to collaborators. Distributed multi-user repositories are obviously necessary to support sharing; we intend to support several models, including replication, shared repositories, distributed repositories, proxy access, and replication by proxy.

Rebuilding Visage to rely on the shared repository enables the kind of collaboration we covet. A U-Form repository allows the radical separation of data from presentation that is necessary to support polymorphic collaboration. The repository and its API are also necessary to allow development of proper clients for

palmtop and handheld devices—these devices are expected to perform as well-integrated parts of a larger Visage environment, not merely tangentially associated or just as adjuncts to the workstation client. MAYA's general approach is to focus on building an *architecturally correct infrastructure* that enables our objectives.

Much of our effort centers on the goal of polymorphic shared frames, because we believe this to be the crux of Visage-Link's innovation. Visage is uniquely positioned to provide this due to its information-centric architectural framework and the conceptual goals behind MAYA's R-E-T model. Polymorphism will allow users to share a *domain of discourse* while visualizing the data in a manner that enhances their individual ability to contribute to distributed analysis. With this power comes the potential for complexity, and we intended to investigate User Interface approaches to mitigate difficulties with this new collaboration paradigm.

Technical Work Accomplished

U-form Serialization

We referred to our first prototype for a collaboration function within Visage as the "Black Hole." The "Black Hole" was a frame that was connected to one other user's display. Dropping Visage elements into this frame would cause them to appear on the other user's Visage desktop. This may seem like something that is of limited use -- a miniature step forward in the collaboration space. In fact, it was an important step for several reasons. First, in the space of collaboration options, this fits best with the goals of Visage -- since Visage's strength hinges on the separation of the data from presentation, it is important that the collaboration be about the data. In fact, the black hole was important as a first step as much for what it wasn't -- we did not share users' screens with each other; we did not add videoconferencing-like features. Instead, we made it possible for users to share data with each other, and just the data.

Immediately, we could see the collaborative possibilities. A user could sift through data, and once something interesting is found, drag the interesting data to the black hole to allow another user to continue the analysis. If the second user finds some related data, they can drop it in the black hole to send it back to the first user. This notion of sharing the data (versus sharing the visualization or the presentation of the data) is very powerful; Sarin and Greif² identified this in 1995.

The Information Architecture of Visage is built upon the notion of concepts, which are individual data objects, organized into webs by relations. Visage frames (the individual visualizations) are stored in this same way; they are constructed from webs of

² Sarin, S. & Greif, I. (1985). Computer-Based Real-Time Conferencing Systems. *Computer*, 33-45.

concepts. Since the black hole allowed users to transmit concepts to each other, extending it to handle webs of concepts allowed us to transmit visualization frames from user to user. Here again, the architecture of Visage allowed new forms of collaboration. If one user sends a map containing some interesting data to another, the second user can change the visualization to any other frame in his stable, allowing him to perform analyses that may not have been able to be performed by the first user, identifying new patterns in the data.

Later iterations of Visage-Link, of course, took this concept and extended it much further.

Handheld Devices: Palm Pilot

MAYA built a robust version of a Visage-style repository for 3Com's Palm Pilot. A shared library was constructed which implements the repository and the repository API, necessary for building Visage-compatible visualizations on a handheld device. The repository is also a solid foundation for data connectivity between handheld devices and server-based repositories. We implemented conduits to allow Visage concepts to be replicated between desktop clients or repository servers and hand-held clients. User interface design work included the creation of visualization frames for the Palm Pilot based on the information-centric paradigms present in the desktop/workstation versions of Visage. For the Palm Pilot, we created the following frames and functionality:

- An Outliner, to display concepts in a spreadsheet-like display, and to allow navigation among webs of concepts by following relationships.
- A Map, to display concepts based on their geographic location. Like the workstation versions of Visage, the map frame provides for any number of maps to be displayed (i.e. World, PACOM, United States, Pittsburgh). The Palm

Pilot-based map frame, however, does not connect to a live map server to allow infinite zooming and panning.

- A concept editor, to allow users in the field to edit attribute values. Editing concepts on a Palm Pilot provides a mechanism for users to take Visage into the field, record observations directly, and to synchronize the edits with other Visage users when the Palm Pilot is returned to its cradle (see *conduit*).

A Palm *Conduit* that handles the synchronization between the Palm Pilot based repository and workstation-based repositories.

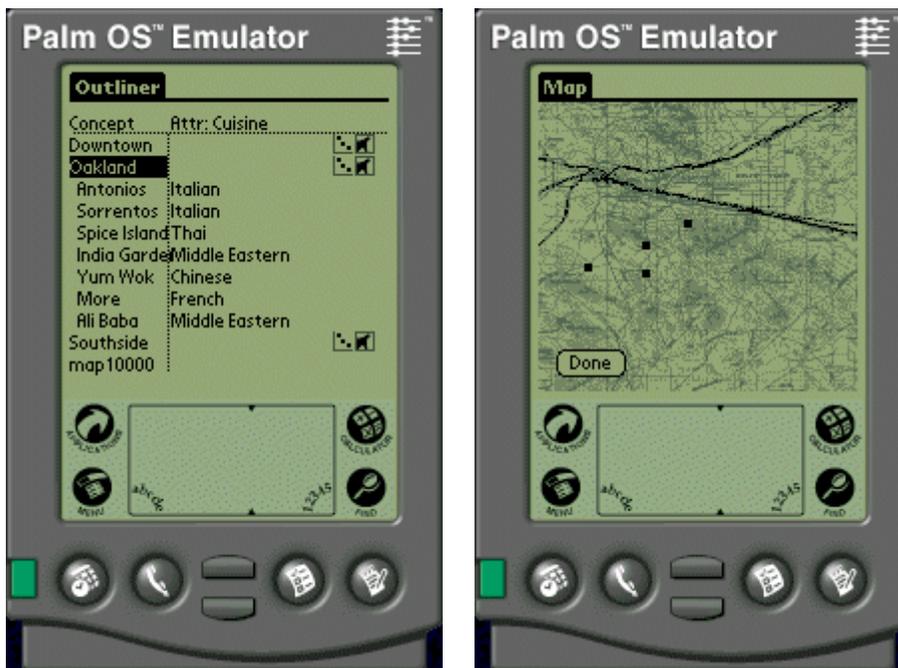


Figure 3 - Palm Pilot screens showing Visage frames as implemented for a handheld platform

Repository: Servers

MAYA brought several repositories on-line which implement robust storage of U-forms so critical to many of our Visage goals, including many of the collaborative goals of Visage-Link.

Although some of the repository work was in the domain of the Visage Hardening project, much of the development has been

funded by Visage-Link because of the excellent opportunities for research in the collaboration space that a robust, shared Visage repository affords.

Several repositories, prototypes, and test beds have actually been built. We have built a Java-based repository for experimentation and demonstration, as well as Python-based repository emulators, file-system based local repositories, and experimented with building repository interfaces on top of relational databases.

Our most robust, production-level repository, known as the VIA³ Repository, is used every day by Visage users, and it can be run on Unix and Windows platforms. The repository continues to be at the center of Visage research and deployment. MAYA Viz's CoMotion⁴ collaborative visualization product is built on top of the Visage repository and uses much of the technology developed by the Visage-Link project. DARPA's Genoa and Archer projects both use the Visage repository, and MAYA's ongoing Visage research, including RenEx, Expeditions, and Civium all rely on the repository.

³ Visage Information Architecture

⁴ <http://www.mayaviz.com/>

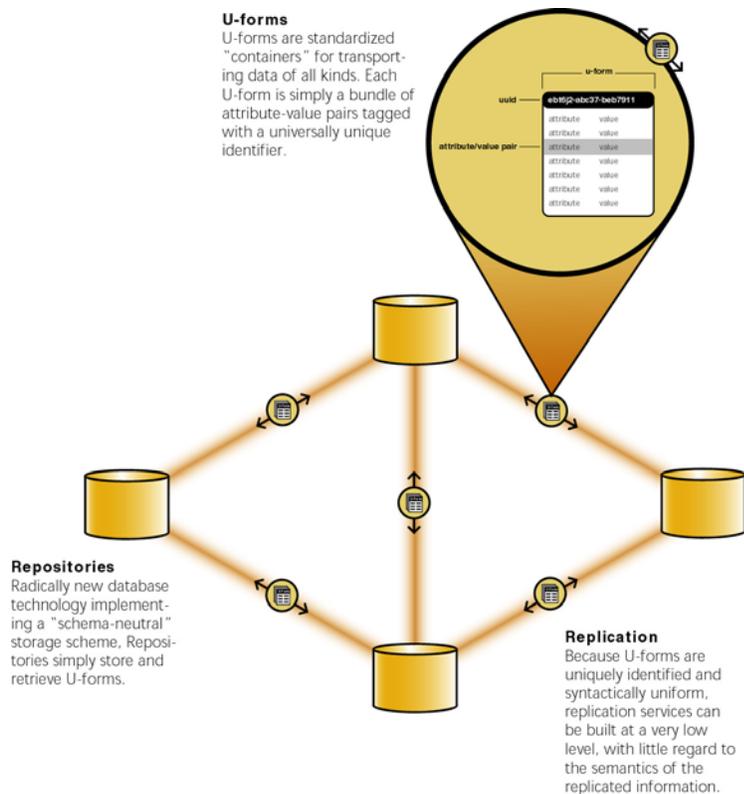


Figure 4 - Repositories store and retrieve U-forms, supporting the information-centric data model of Visage and the polymorphic collaboration available in Visage-Link

Repository Components

When we speak of the current implementation of the MAYA repository, we are actually speaking of two distinct entities; the TCP/IP repository server, and the library of code on which it is built. Both the server and library have been ported and tested on Solaris, Linux and Windows 95/98/NT.

Brief Overview of the VIA Repository

The VIA Repository is an information device whose purpose is the persistent store of data. This data is encoded as u-forms, or bundles of attribute name/value pairs, such that a particular value for an attribute can be referenced by its name. Each such bundle is keyed in the repository by a universally unique identifier, the UUID. What this implies is that any two u-forms in

different repositories with the same UUID actually reference the same entity. Operations on the repository typically involve getting and setting attributes using the UUID of the u-form, and the name of the attribute in question.

Library

The VIA Repository library is implemented in C++, and is itself is built on top of a commercial embedded database library, Berkeley DB from Sleepycat Software (www.sleepycat.com). It was chosen because of its mix of features and its low initial cost (free for evaluation).

The basic API of a repository library is intentionally kept very simple. Clients may get, set and remove u-forms, individual attributes of a u-form, as well as iterate over the contents of a repository. Some additional functionality is built on top of this API in order to allow the extraction of pieces of attributes (chunking), and for dealing with attribute values as structured data. There is also the provision for journaling. This is not to support transactions, but so that repository modifications are written to a separate disk file in order to aid in catastrophic crash recovery.

Repository Server

The VIA Repository server is also written in C++, and is a connection-oriented server; each client establishes a persistent TCP/IP connection and context with the server. There is some ability to do authentication and authorization on individual clients, and for managing access rights on u-forms. The messaging interface of the server mirrors that of the underlying library API. The server adds the additional capabilities to register interest in and receive change notifications on u-forms, and most recently the ability to perform queries.

Scalability

For DARPA's Genoa project, MAYA evaluated the repository. Based on the results of our experiments, we conclude that the repository behaves in a predictable and scalable fashion overall. The cost of accessing an arbitrary attribute in an arbitrary u-form, if the cost of reading in the u-form is ignored, is $O((\log n) * m)$, where n is the number of u-forms, and m is the number of attributes on a u-form. If m is suitably small, as we expect it would be, this reduces to $O(\log n)$.

Repository: Visage

MAYA has modified Visage to enable it to connect to the repository. (Before the Visage-Link project, all concepts were stored in memory or a single-user disk cache, which precluded collaboration.) This change allows us to visualize shared data sets and to immediately be notified of changes made by other Visage users and other clients connected to the repositories. A goal was to be able to have Visage bootstrap by requesting all the components that make it up from the repository. We have demonstrated Visage running on top of a repository, and have also demonstrated our ability to put visualization frames into a repository and re-render them. This is quite an important achievement, since it allows several users to share visualization—they need only request the concept representing the frame from the repository. It also allows users to view different (polymorphic) visualizations of the same data—in this case, the users ask for visualizations of the same collection, which is taken from the repository. We have demonstrated polymorphic collaboration in the past, but building this functionality on top of a shared repository is architecturally more sound and is on the critical path to far more interesting and useful shared experiences.

Repository: Clients

Building shared repositories with well-defined APIs and designing a standard message format are two of the primary elements in the Information Devices Architecture (IDA). The goal of IDA is to decompose computing into components which fall into three classes: Transducers (often renderers), Executors, and Repositories, all of which communicate using a standard message format. The existence of a repository has allowed us to demonstrate several executor-renderers which can communicate with the repository. A few are able to transduce data from other sources into the repository (for example, World Wide Web pages). Some of these are lightweight components which create visualizations of concepts in the repository; these are especially interesting in the domain of the Visage-Link project, allowing collaboration between users of Visage and users on other platforms and environments. Examples include visualizations built in Tcl/Tk and Java, intended to run on handheld computers and thin clients.

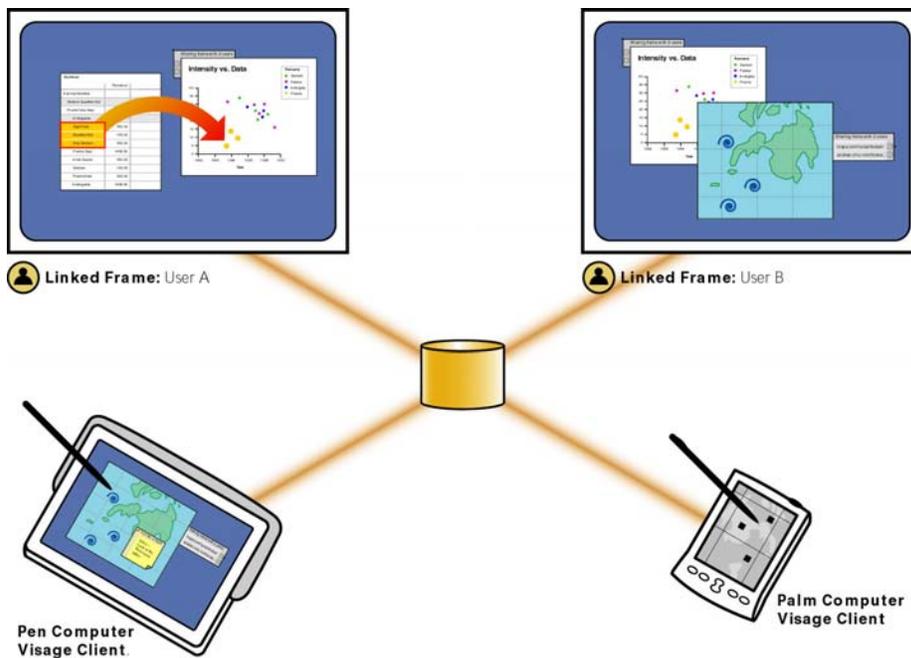


Figure 5 - This diagram shows how multiple heterogeneous clients can connect to the Visage repository, and users can collaborate even when running Visage-Link on different devices.

Integration: Rough-N-Ready

Funded partly through DARPA's IC&V program, BBN's Rough-N-Ready⁵ meeting transcription application creates annotated, indexed transcripts from audio input streams. (Francis Kubala of BBN is the principal investigator for Rough-N-Ready.) Rough-N-Ready provides the following capabilities:

- Processes recorded audio from broadcast news, meetings, etc.
- Produces partial transcripts
- Identifies entities spoken about (people, companies, etc.)
- Indexes words, concepts, and speakers
- Locates segments where each person is speaking

⁵ <http://www.bbn.com/speech/roughnready.html>

- Assigns multiple, ranked topics to segments
- Automatically creates structural summaries and stores them
- Quickly and easily retrieves relevant information
- Allows users to skim for topics without listening to the recording

MAYA was able to put data from the Rough-N-Ready meeting browser into a repository and build visualizations in Visage. The ad hoc navigation and depiction capabilities available in Visage using only stock capabilities are impressive, and complement the native capabilities within Rough-N-Ready. In addition, MAYA built a few customized frames that leverage knowledge about the domain and the data architecture. Conversations with BBN revealed a number of services which can be exposed by the Rough-N-Ready system to allow more complex interactions including query-by-example, topic extraction, and visualization of conversation threads.

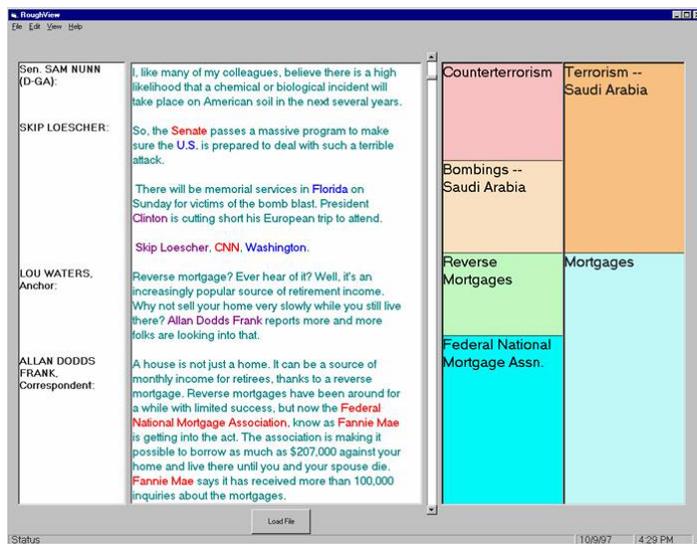


Figure 6 - Rough-N-Ready native interface, showing speaker, transcript, and topics

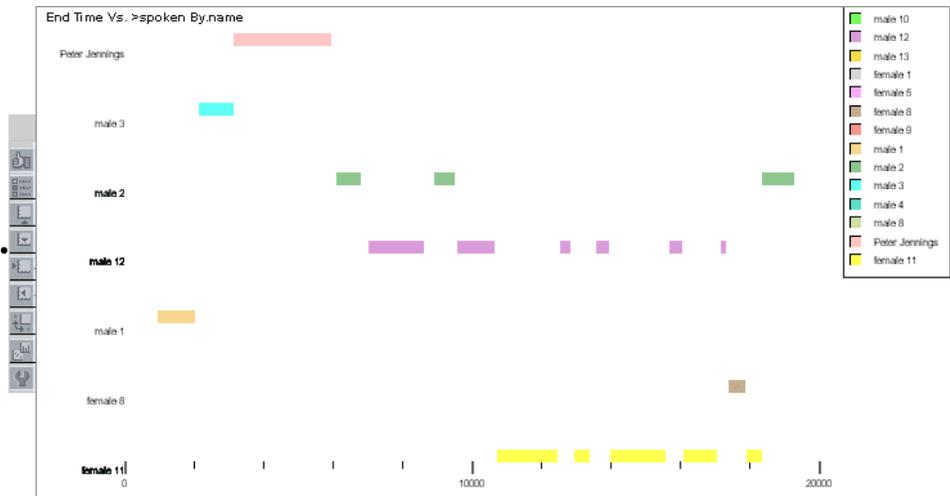


Figure 7 - A Visage plot frame, showing Rough-N-Ready data. Each bar shows a speaker "turn." The length of each bar is proportional to the time spent speaking; each turn can be used as the starting point for ad hoc analysis (for example, dragged to a Visage map)

Integration: CoMotion

MAYA Viz is a sister company to MAYA Design, founded to develop software systems that use visualization and collaboration techniques to improve analysis and decision making. MAYA Viz maintains core competencies in software engineering and information design.

"MAYA Viz's flagship product, CoMotion 2.0™, is a unique software solution that integrates collaborative analysis and decision-making capabilities in distributed, information-intensive environments. Using visualization and highly interactive information manipulation techniques coupled with expansive collaboration abilities, CoMotion delivers unmatched functionality that will revolutionize the way corporate teams work together." ⁶

CoMotion is a commercially available collaboration product derived in part from Visage-Link. The research into polymorphic

⁶ <http://www.mayaviz.com/products/>

collaboration performed under the Visage-Link project influenced CoMotion greatly, as can be seen in the language used to describe the product as well as in the user interface. In the MAYA Viz description of CoMotion, polymorphic collaboration is referred to as “deep collaboration:”

“In CoMotion, analysts and decision makers can move rapidly from retrieving data, drilling into the details of individual data elements, viewing different configurations and aspects of the data, to collaboratively analyzing and manipulating the information.”

“Unlike other commercially available collaboration software, which provides simple white-boarding and low tech replication of application screens, CoMotion supports deep collaboration, i.e., the ability to interact with other users over live data. Deep collaboration enhances productivity and effectiveness as people from various fields of expertise collaboratively navigate, assess, and understand complex information in highly visual and interactive representations.”

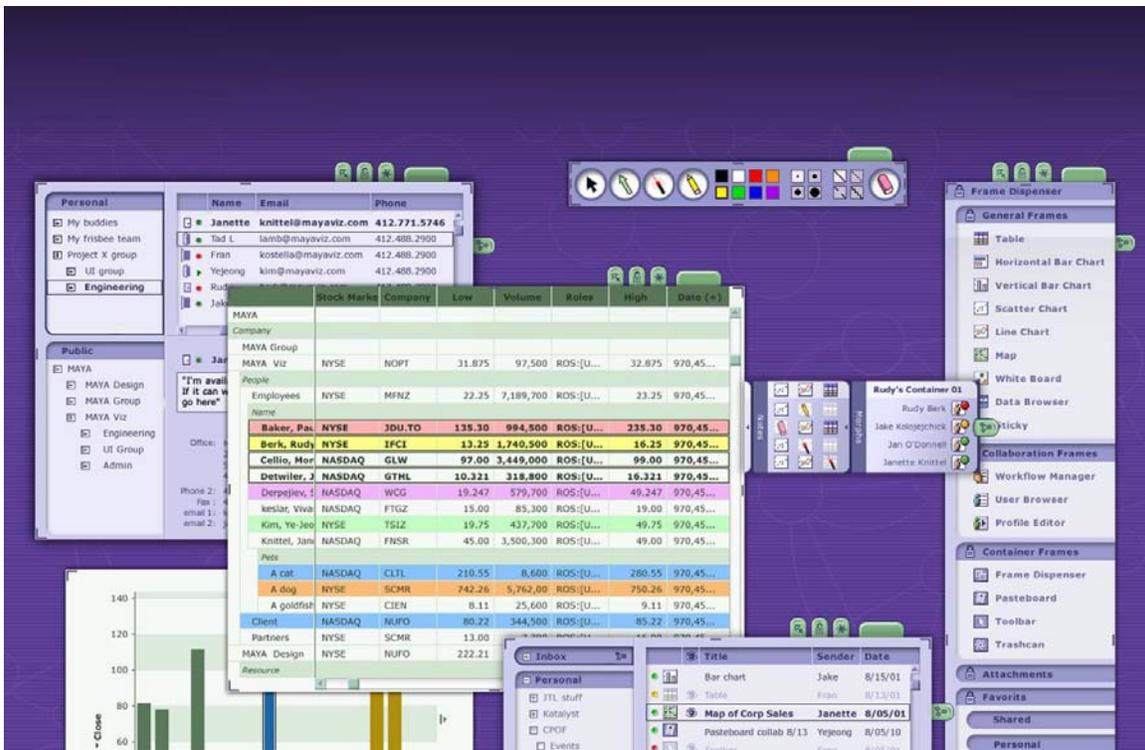


Figure 8 -- MAYA Viz's CoMotion product. Polymorphic collaboration, shared marking, and the drawer-like attachments that extend frames were all originally developed under the Visage-Link research project.

Attachments

In order to incorporate collaborative capabilities into Visage, MAYA had to consider ways to control the collaboration. This includes initiating sessions, terminating sessions, sharing frames, sharing data concepts, and visualizing the sharing status. Since Visage is an information-centric direct-manipulation environment, it was critical that the collaborative controls be designed and implemented in a manner consistent with the existing Visage paradigms.

The collaborative features needed to be considered as a way to *extend* the capabilities of existing Visage frames, and needed to be designed in a manner that would support not only existing

frames, but *future* frames as well. Specific goals for the collaboration controls included:

- Provide a mechanism to add controls to frames (control frames and contents)
- A modular system; users can easily add and remove controls
- Move away from the canonical "WIMP" design (windows have decorations)
- Users immediately see Visage as a dramatically different visualization environment.

MAYA designed an extension to Visage frames called "attachments." These attachments can be used to extend the capabilities of frames to include collaborative features, query tools, filtering tools, publishing capabilities, and more. The attachments connect to the visualization frames in order to extend them. They slide in and out like drawers, so they can be put away when not in use to reduce clutter and interference with the visualizations. A number of attachments were designed, but one of the most important is the "Sharing Attachment," which is the most direct embodiment of Visage-Link within the Visage environment. Users use the sharing attachment to initiate collaborative sessions with other users, to monitor the collaboration, and to control aspects of the collaboration such as whether or not selection is shared with other users.

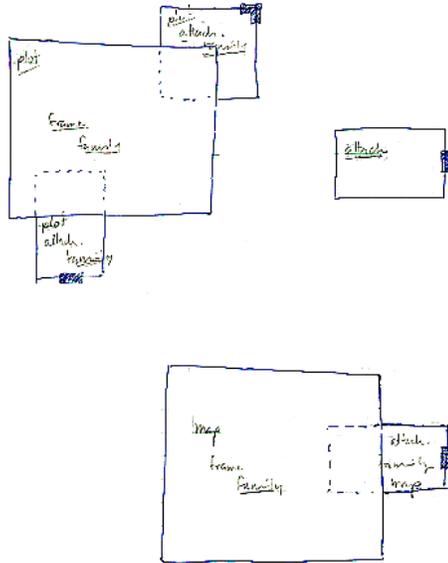


Figure 9 -- Early sketches of Attachments, showing how attachments connect to Visage frames to extend their functionality.

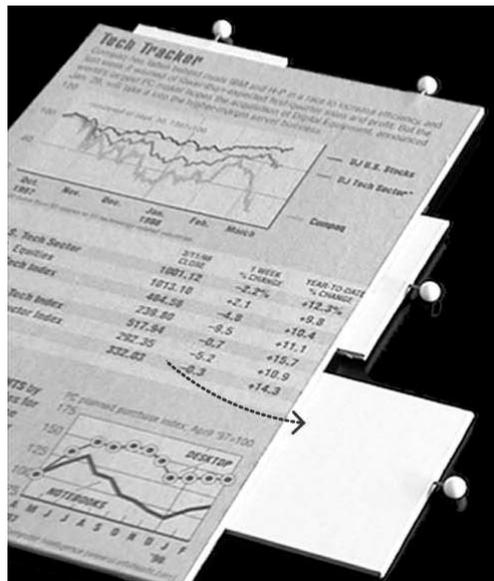


Figure 10 -- A physical model created to explore the relationship between Visage frames and the newly-designed attachments

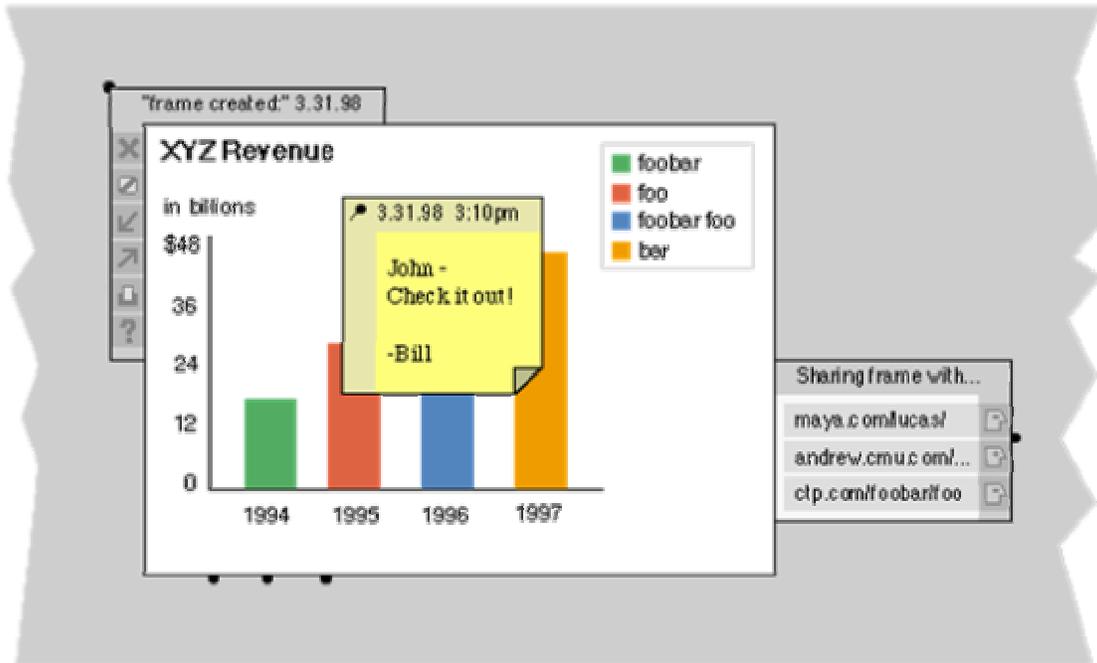


Figure 11 -- A Visage frame, with several attachments connected. Note that there are several different types of attachments; each extends the frame in a different way. The sharing attachment (Visage-Link) is connected to the right-hand edge of the frame; this user is collaborating with three other users..

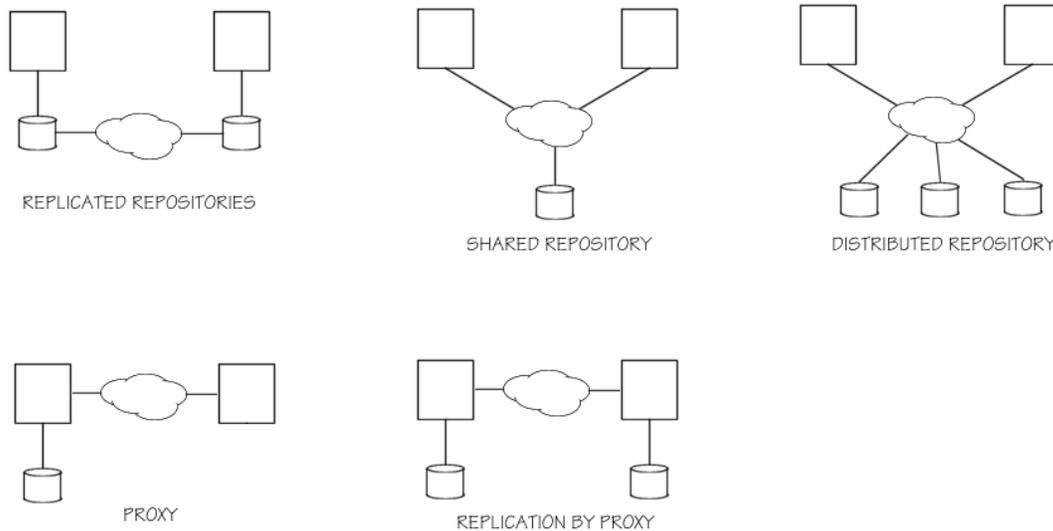
Repository Configurations / Shepherds

When researching Visage-Link, we determined that there were many ways that two Visage clients could access the same U-forms in order to collaborate. One of the most obvious methods is to have all clients access the same shared repository. Indeed, this is one of the first prototypes built, and a number of features were added to the repository in order to support this kind of access. Access control, change notification, locking protocols, and instrumentation were all added to the repository. Data structures to hold user information, authentication details, and in some cases, change history were all added.

Visage-Link is intended to work in heterogeneous environments across a wide range of devices with many levels of connectivity.

Palm Pilots, although gaining wireless connectivity had little or no network connectivity when the Visage-Link project was ongoing. Our experimentation with wireless networks using pen computers involved slow (1.5mb/s) networks with short range (we extended the range by using aftermarket antennas). The now-familiar and now-affordable 802.11b networks were not available. To this end, we investigated many possibilities for connectivity.

Our Palm Pilot client uses a custom-built conduit with a version control mechanism to allow the user to change data in the field and have it correctly synchronized with a larger repository when returning. The Palm Pilot Visage client has its own (small) repository that is compliant with the Visage Standard Message Format.



MAYA-Visage-Link-98001.a

Figure 12 - Several possible repository configurations, all of which support collaboration in Visage-Link. In these diagrams, squares represent Visage clients, cylinders represent repositories, and clouds represent networks.

Work on distributed repositories is continuing beyond the end of the Visage-Link project. MAYA's Shepherds project, funded by DARPA, continues to investigate the concept of a confederation of repositories kept synchronized by "shepherds" that move U-forms around so the right U-forms are always available to the proper clients. Ideally, a Visage client can connect to any repository as if it "the one," the "Grand Repository In the Sky," also known as "GRIS." The illusion of GRIS is kept up through the work of shepherds, constantly moving U-forms between the actual repositories in the confederation.

Shepherds are networks of autonomous agents that create the fiction of a single, universal collection of data by replicating information among local collections. MAYA is developing architectural principles and infrastructure components for building flexible, robust and usable asynchronous collaborative systems as well as new user interface paradigms for configuring these systems.

Out-of the box Collaboration

As research progressed on Visage-Link, it became increasingly important to make it as easy as possible for users to install Visage and begin collaborating. This involves identifying themselves, possibly pointing Visage to a common repository, and having Visage listen properly for other collaborators.

Goals

The ease-of-use goals for Visage-Link installation and collaboration initiation were as follows:

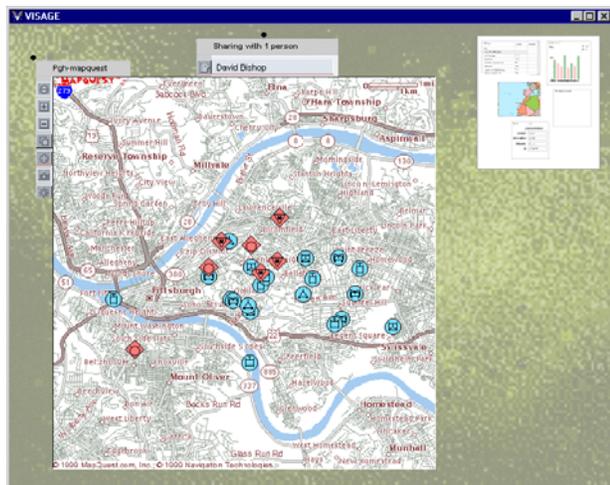
- To allow collaboration to require as little setup as possible (i.e. avoid tweaking preference files, changing U-forms or relations, restarting Visage)
- To allow users with little experience to collaborate
- To allow collaborators to discover each other

- To conform to design notes:
 - 99024 - The "Entity," "Person," and "User" roles
 - 99022 - Repository Portals and Network Services
- To allow the big list of all worldwide collaborators to be handled in some sensible way
- To allow multiple Visage clients to run on a single machine without precluding collaboration for anyone (minor goal: in most instances so far, there is a one-to-one user-to-client-to-machine relationship, especially for Windows machines.)

The Visage-Link project addressed each of these goals in elegant ways, and the designs supported further refinement in MAYA Viz's CoMotion as well as ongoing MAYA work on Visage, RenEx, and the repository.

Demonstrations

Over the course of the Visage-Link project, MAYA supported a number of demonstrations, briefings, and meetings. Some of the most impressive demonstrations were the yearly DARPA CVIM (Collaboration, Visualization and Information Management) demonstrations, where a number of researchers demonstrated their research,



often in an integrated way and always with a common thematic scenario and storyline. Some of the demonstrations supported by MAYA for the Visage-Link project were:

- *May 1998* – CVIM Demo, SPAWAR San Diego – Typhoon PAKA // Philippines // Humanitarian operation scenario (With Visage-Web)
- *April 1999* – Command performance progress demonstration at Rome Labs -- demo included Palm Pilot

client, first demo of shared Repository, wireless connectivity, Vadem Clio pen computer and Fujitsu Pen computer clients.

- *October 1999* – SPAWAR – Crisis Management scenario – rescue operation due to cliff-side cave-in at Point Loma
- *September 1999* – CVIM Demo – Humanitarian relief operation scenario - Typhoon hitting Bali
- *September 1999* – Scholtz, Foresti visited MAYA for a status report and demonstration – includes detailed architecture briefing and discussion
- *Rome Labs TEM 97* (Technology Exchange Meeting) – Presentation and demonstration. (First met BBN's Rough-N-Ready team; later integration between Visage-Link and Rough-N-Ready is covered elsewhere in this report.)

Repository Monitor

One of the interesting challenges during the Visage-Link research involved the storage of Visage itself within the repository. Of course, the repository is a database of sorts, and generally contains interesting data sets that are used for demonstrations, scenarios, testing, visualization, and data analysis. What is not as obvious, however, is that the Visage code itself is stored in the repository as U-forms. Visage is composed of manageable sections of code wired together in a visual programming environment to create the desired frames and behaviors. These bits of code, metadata about the code, and the connections between them are all stored in the repository.

This notion is critical to Visage-Link, for without it users could only collaborate about the data and could not share visualizations unless they had previously synchronized their Visage clients. With Visage itself stored in the repository, any visualization frame available to a user is available to all other users as well, even if they only created the frame a moment ago. In fact, it is possible for a user to modify a frame while they are viewing it and for all other users who are sharing that frame

to see the changes instantaneously. Note that this is potentially far more powerful and unique than merely sharing the data concepts, and is the basis for many of the “deep collaboration” and polymorphic collaboration features in Visage-Link.

The Great Irony

When the Visage code was initially moved from a file-system-based single-user storage area into the repository, Visage was launched for the first time and looked no different. One of the single most important architectural changes in the life of the project was impossible to observe.

In order to observe the repository in action, to see whether Visage was truly running “from the repository,” it was necessary to build a repository monitor. This monitor was also able to show collaboration traffic as users shared frames and data, and was a useful diagnostic tool over the life of the project. The monitor was designed to show the number of repository connections, the number of instantaneous messages (including change notification messages) to and from the repository.

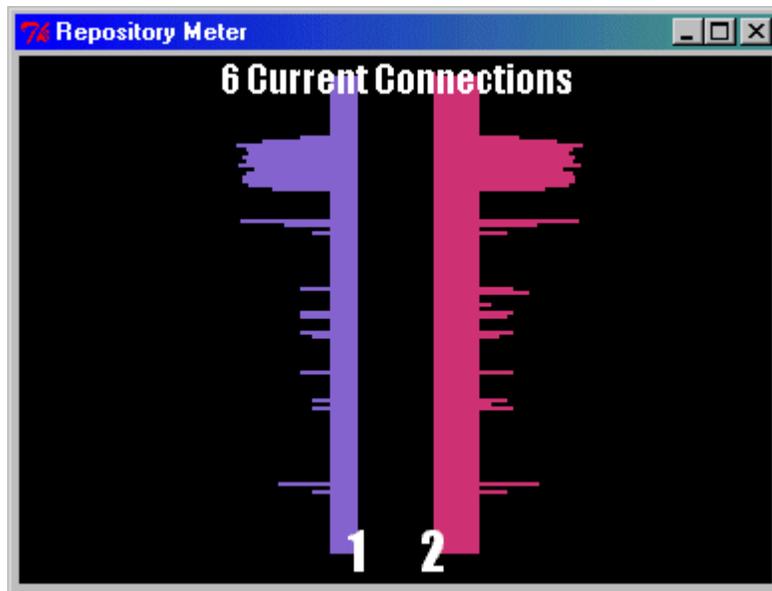


Figure 13 - The Visage Repository meter, showing the number of connections to the multi-user repository as well as a chart of message traffic to and from the repository over the last minute.

Shared Marking

Maintaining context among a collaborative group when it is unknown what kind of visualization other users are viewing is obviously difficult. Indicating focus is equally hard – a user with a map may wish to refer to items by their geography, saying, “The five trucks grouped together in the northwest,” which won’t indicate much of anything to another user who has organized a bar chart to show the truck’s capacities.

Many visualization systems include a notion of *painting* or *marking*, which is generally used to color data items based on some criteria or to highlight outliers or interesting features. In Visage, marking is used as well, and it is especially useful when the same data concepts are represented as elements in multiple frames that contain different visualizations. Marking all the high crime cities in a bar chart showing comparative crime rates will automatically and simultaneously mark them in a map frame and a spreadsheet frame. In a polymorphic environment, this is one

of the highest value features supporting data exploration and hypothesis validation.

Since Visage-Link relies on polymorphic collaboration (above and beyond polymorphic visualization), marking was extended to allow *shared marking*. When users morph their frames so they are viewing different visualizations, they can rely on shared marking to indicate focus. This allows a user to perform a visual query, like selecting the coastal cities on a map, then mark the selected cities to indicate to a user who isn't viewing a map which cities are being discussed.

In a sense, this is backing down slightly from the fully-polymorphic end of the collaboration continuum; it turns out that if users visualizations are too divergent, discussions are so abstract that the value of polymorphism is lost. Visage-Link addresses this with several features. First, by allowing the users to control how tightly linked their visualizations are, from isomorphic (when one user morphs their frame, other users' frames follow) to polymorphic (maximum autonomy to morph frames). Second, by including shared marking to provide users a method for indicating focus even when everything else about their visualizations is different.

An interesting discovery was made while developing the Visage-Link system: without shared marking, users invented ways to indicate focus on their own. Typically, one user would alternately remove and replace data elements from a visualization by repeatedly dragging them out of and back into a frame. Doing this makes the elements "blink" on other users screens, as their visualizations update to show the proper collection of data. An ingenious approach, but not exactly an optimal user interface, not a very scalable solution, and quite dependent on network speed.

Conclusion

A collaboration continuum was shown at the beginning of this report, where systems were placed according to how tightly coupled users' views are with each other. Visage-Link was built to explore the end of this spectrum where data presentation was completely independent for each user forcing the domain of discourse to be *only* the data.

Developing this system required careful thought to maintain the character of Visage and to leverage its drag-and-drop behavior and direct-manipulation "feel." It also required implementation of features to support users when collaborating in the polymorphic realm beyond relaxed WYSIWIS, including controls for varying the degree of frame coupling as well as focus indication schemes like shared marking.

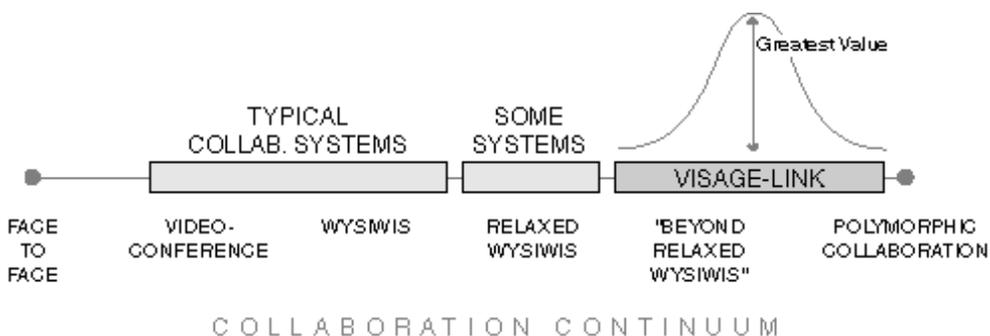


Figure 14 - The greatest value for collaboration beyond relaxed WYSIWIS comes before maximum polymorphism.

Polymorphic collaboration has great value as a deep collaborative device, as long as it is not pursued so fully that users are left in a realm that's too abstract.