

DISCIPLE: A Framework for Multimodal Collaboration in Heterogeneous Environments

IVAN MARSIC

Rutgers — The State University of New Jersey

This paper presents a framework for sharing JavaBeans applications in real-time synchronous collaboration. A generic collaboration bus provides a plug-and-play environment that enables collaboration with applications that may or may not be collaboration aware. Research on knowledge-based quality-of-service management and multimodal human/machine interface is described.

Categories and Subject Descriptors: H.5.2 [**Information Systems**]: User Interfaces; H.5.3 [**Information Systems**]: Group and Organization Interfaces; D.2.2 [**Software**]: Design Tools and Techniques; C.2.4 [**Computer Systems Organization**]: Distributed Systems

General Terms: Human Factors, Design

Additional Key Words and Phrases: Synchronous groupware, CSCW frameworks, shared electronic workspaces, group communication, JavaBeans, multimodal interface

1. INTRODUCTION

As digital networking becomes ubiquitous, the opportunity grows for collaborative knowledge work through conferenced computing. The shift is from traditional productivity applications, such as word processors and spreadsheets, to computers as communication devices. In this context the machine takes on the role of mediator in human/machine/human communication—the ideal being to extend the intellectual abilities of humans through access to distributed information resources and collective decision making. However, the design of successful multi-user applications remains a great challenge. Significant issues include:

—concurrency control, consistency maintenance, and quality-of-service (QoS) requirements

Components of this research are supported by DARPA Contract No. N66001-96-C-8510, NSF Contract No. IRI-9618854, and by the Rutgers Center for Computer Aids for Industrial Productivity (CAIP). CAIP is supported by the Center's Corporate Members and by the New Jersey Commission on Science and Technology.

Address: Center for Computer Aids for Industrial Productivity (CAIP), Rutgers University, Piscataway, NJ 08854-8088, USA

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works, requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept, ACM Inc., 1515 Broadway, New York, NY 10036 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 1999 ACM 0360-0300/99/1200

- duplication of effort already expended on single-user applications when extending to multi-user domain, which often results in a failure to keep up with the latest features available in the single-user counterparts
- interaction through keyboard and mouse rather than the more natural ways humans use to communicate with each other

This report describes an evolving framework that addresses the above issues. The main characteristics of DISCIPLÉ (*D*istributed System for *C*ollaborative *I*nformation *P*rocessing and *L*Earning) are a layered architecture, explicit knowledge-based support for software modules, and multimodal human/machine interaction (see Figure 1). The paper reviews two areas of our research: software architecture of the framework and multimodal human/machine interface.

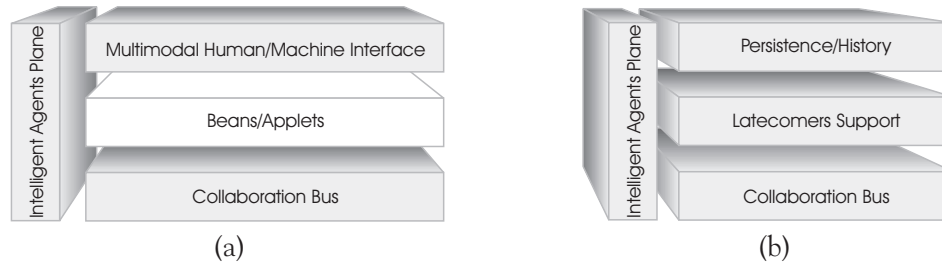


Fig. 1. Software components of the DISCIPLÉ framework on the client/user side (a) and the server side (b). (Applets are not part of the framework—they are supplied by the application developer.)

2. COLLABORATION BUS

An important goal of this work is to enable easy sharing of single-user applications, since the majority of applications continue to be developed for a single user. This is achieved through dissociating to the maximum the communication and group aspects from the application task. The framework is targeted for a particular class of applications—applications known as JavaBeans [Sun Microsystems, Inc. 1996], which include Java applets.

Our approach offers a single solution to what is currently viewed as two disparate problems: a toolkit for development of special purpose applications that are collaboration aware and a framework for sharing existing single-user (collaboration-transparent) applications. A key component of DISCIPLÉ is the *collaboration bus*¹ (Figure 2), into which both kinds of applications get “plugged” and the bus enables multi-user sharing [Marsic and Dorohonceanu 1999]. A major advantage is that it requires no code modifications in either the underlying Java platform or in the application. This is achieved by exploiting the Java delegation event model, where the bus acts as a listener for application events. The intercepted events are then processed for group usage and distributed to the remote peers.

¹The collaboration bus presented here has no relationship to the collaboration bus developed at the University of North Carolina [Dewan 1997].

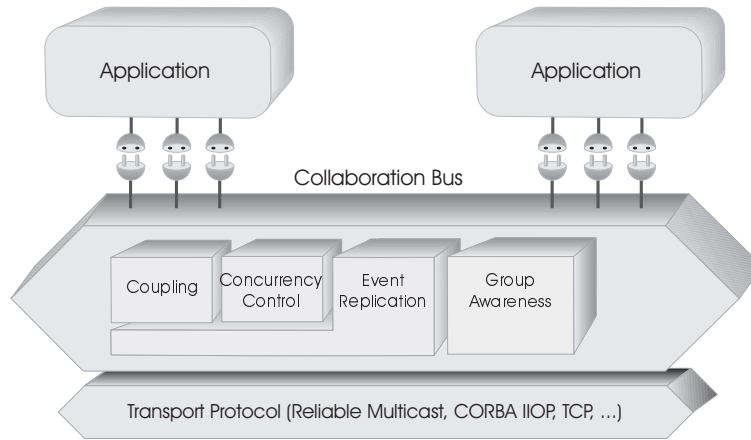


Fig. 2. The collaboration bus architecture. Each conferee runs a replica of the application. The sockets symbolize the user's actions that cause the transitions between the application's states. The bus provides various group- and communication-related services.

Figure 3 shows the current graphics user interface for the DISCIPLÉ desktop. The top left corner shows the desktop manager with hierarchical representation of the entire collaboration space. The hierarchy is as follows: organizations > places > participants. Places are persistent and may get more elaborate and complex with time (in terms of artifacts and participants' relationships). Workspace on the right corresponds to a place with artifacts, i.e., Java Beans/applets. The example Beans shown are whiteboard (back) and an image-guided medical diagnosis Bean (front) [Comanicu et al. 1998]. Bottom left shows a browser that allows the user to import the Beans from the local disk space or from the Web. The imported Beans can then be dragged and dropped to the workspace where they become shared by remote conferees. The user can also copy the Beans from a place to any other place on the desktop. Thus, sharing a Bean is as easy as pointing to its URL.

3. INTELLIGENT AGENTS PLANE

Classically, the knowledge about a software module's functionality is built-in implicitly or hard-coded into the module. This implies that no adaptation or learning is possible to adjust to the particular circumstances or evolving user requirements. A major theme of our research is to make the knowledge mechanisms explicit, rather than intertwined with the module's functionality, as well as dependent only upon the general functionality of the modules, rather than being application-dependent. Figure 1 shows intelligent agents spanning across all layers and providing knowledge support. An example of this research is our work on run-time learning of object migration policies, transparent to the user as well as to the application programmer [Marsic and Jonnalagadda 1999]. The novelty of the proposed approach is that the migration policy is derived by the system from an application's behavior at run time rather than from its software architecture at design time by the programmer.

The approach is focused on the run-time classification of objects as to their suitability for migration. For example, network-congesting calls might be avoided



Fig. 3. Screen snapshot of the DISCIPLÉ desktop.

if client objects and server objects are migrated to a common host prior to the calls (similar to downloading of Java applets). We conducted several collaborative sessions with a multi-user graphics editor and measured network traffic. For this we modified the CORBA Object Request Broker to intercept the remote calls and to record the various parameters to be used in classification. The data recorded over several sessions [Marsic and Jonnalagadda 1999] demonstrate that the behavioral patterns of alike objects are very similar across different sessions (Table 1). This finding supports the proposed idea of using an object's behavior in classification to devise a migration policy. We are currently building a decision-making system that will use these statistics to derive object migration policies.

	Lifetime	# of calls	Call frequency	Resource consuming tasks	Data transfer amount	Dataflow direction
Editor	long	few	uniform	few	low	out
Selection	long	few	uniform	very few	low	in & out
CreateOp	short	many	exp. ↓	many	low	in & out
ResizeOp	short	many	exp. ↓	many	low	in & out
MoveOp	short	many	exp. ↓	many	low	in & out
CutOp	short	many	exp. ↓	many	low	in & out

Table 1. Summary of the dynamic characteristics of software objects in a multi-user graphics editor. 'exp. ↓' stands for 'exponentially decreasing.'

4. MULTIMODAL HUMAN/MACHINE INTERFACE

The DISCIPLÉ human/machine interface (Figure 4) incorporates speech recognition and synthesis for conversational interaction, combined and synchronized with manual gesture sensing from a force-feedback glove [Burdea 1996] and gaze direction from a desk-mounted gaze tracker. The current system uses a finite-state grammar and a restricted problem-specific vocabulary. The speech recognizer is operated “hands-free” from speech captured by a fixed-focus microphone array [Flanagan and Jan 1997].

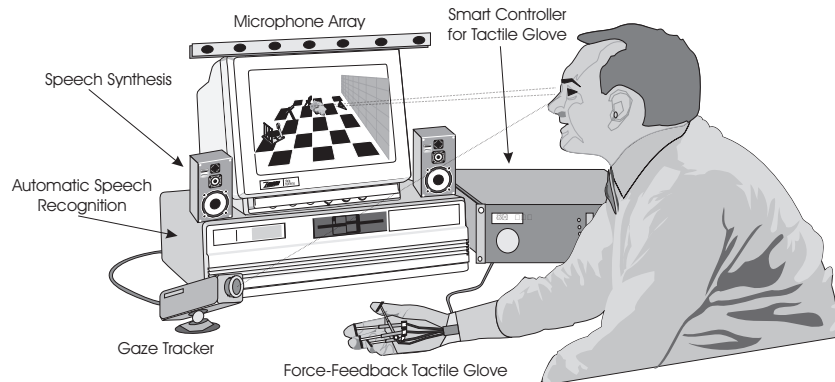


Fig. 4. Interface modalities for sight, sound and touch provide natural-like capabilities for cooperative manipulation of objects in shared workspaces [Medl et al. 1998].

Fusion of data from the modalities is accomplished by a slot-filling method in which a parse of the recognized text string is synchronized with the tactile or gaze input. Clearly, the sensory inputs can overlap in information content and exhibit redundancy. Clearly, too, in some instances a single modality is sufficient and natural, and can subsume the entire task. A central issue is the development of reproducible tests and quantitative metrics that reveal the synergies obtained from multimodal human/machine communication.

5. CONCLUSIONS

The article presents the DISCIPLÉ framework that facilitates the development and multimodal sharing of applications by geographically separate coworkers. Our continuing research investigates new methods for managing distributed objects in synchronous groupware, particularly with respect to partial failure survival, concurrency control, and application synchronization. Another major development is knowledge-based planning and learning to maximize the QoS (e.g., response timeliness, computational accuracy) in heterogeneous environments with different mixtures of resources. The ongoing human performance studies are aimed at quantifying the benefits of multimodal collaboration. Further information about DISCIPLÉ as well as source code and documentation are available at

<http://www.caip.rutgers.edu/disciple/>

ACKNOWLEDGMENTS

Research contributors to this project include Professors James Flanagan, Casimir Kulikowski, Peter Meer, Marilyn Mantei Tremaine, Grigore Burdea, Joseph Wilder, and Attila Medl. The students involved in the DISCIPLE project, Maurits Andre, Bogdan Dorohonceanu, Cristian Francu, Stephen Juth, Senthilkumar Sundaram, and Weicong Wang, were also critical to the evaluation of the ideas presented here.

REFERENCES

- BURDEA, G. 1996. *Force and Touch Feedback for Virtual Reality*. John Wiley & Sons, New York, NY.
- COMANICIU, D., MEER, P., FORAN, D., AND MEDL, A. 1998. Bimodal system for interactive indexing and retrieval of pathology images. In *Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV'98)* (October 1998), pp. 76–81. IEEE.
- DEWAN, P. 1997. Collaboration bus. University of North Carolina, Chapel Hill, NC. Available at <http://www.cs.unc.edu/~dewan/cb.html>.
- FLANAGAN, J. L. AND JAN, E.-E. 1997. Sound capture with three-dimensional selectivity. *Acustica* 83, 644–652.
- MARSIC, I. AND DOROHONCEANU, B. 1999. An application framework for synchronous collaboration using Java Beans. In *Proceedings of the 32nd Hawaiian International Conference on System Sciences (HICSS-32)* (January 1999). IEEE.
- MARSIC, I. AND JONNALAGADDA, L. 1999. Using network traffic statistics in learning object migration policies (1999). Submitted for publication.
- MEDL, A., MARSIC, I., ANDRE, M., KULIKOWSKI, C. A., AND FLANAGAN, J. L. 1998. Multimodal man-machine interface for mission planning. In *Proceedings of the AAAI Spring Symposium on Intelligent Environments* (March 1998), pp. 41–47. AAAI.
- SUN MICROSYSTEMS, INC. 1996. JavaBeans 1.0 API specification. Mountain View, CA. Available at <http://www.javasoft.com/beans/>.