Data-Centric Collaboration for Wired and Wireless Platforms

Ivan Maršić

Department of Electrical and Computer Engineering and the CAIP Center, Rutgers University, Piscataway, USA

With the proliferation of mobile computing devices there is an increasing demand for applications supporting collaboration among users working in the field and in the office. A key component for collaboration in this domain is sharing and manipulation of information using very different devices and communications. We propose a novel, data-centric collaboration paradigm, where each user can obtain a subset of the shared data and the data may be visualized differently for different users. The data amount and the visualization technique reflect the user's interests and/or computing and communication capabilities. The users collaborate on and exchange data, and the data is dynamically transformed to adapt to the particular computing/network platform. The resulting design is simple yet very powerful and scalable. It is implemented and tested by developing several complex groupware applications.

Keywords: software frameworks, groupware, heterogeneous computing, adaptive applications.

1. Introduction

With the recent proliferation of connectivity for small mobile computing devices, using cellular links, wireless LAN, Bluetooth and other radio-based technologies, the door has been opened for development of systems for ubiquitous collaboration with wide range of applications, such as mobile commerce, in-time training and maintenance, law enforcement, medical emergency and disaster relief services. The major challenge with wireless connections is that they are, by nature, very dynamic, with the available bandwidth and the latency changing rapidly over time. Another issue is that users need to seamlessly move between different computing devices and use the ones that best suit their current needs, all during an ongoing collaborative session. Awareness of network, as well as computing and input/output resources is essential for optimizing the performance of collaboration systems in such contexts. The aim for collaborative applications is to adapt to the changes in the environment to best utilize the resources available. The following scenario illustrates the need for such applications.

An airport maintenance worker needs to check airplanes. The worker was recently hired and still needs training. He is using a handheld or a wearable computer with a wireless link while performing his task. A senior technician is located at the help desk and possesses an expert knowledge about airplane maintenance. He uses a desktop workstation in his work and may simultaneously assist multiple filed workers. He also needs sporadically to consult an expert engineer at an airplane factory who is using a virtual reality workstation. They are connected over a high-speed wired Internet link. All three types of users share the same engineering drawings, but they need to display them on unlike devices and they interact over unlike communication links. The data also need to be customized to meet the communication link constraints (available bandwidth) or computing constraints of the user's device (available memory). Due to their different roles and expertise, the users are also interested in different aspects or subsets of the data.

There have been heterogeneous platforms and applications in the past, but what is new is ubiquitous connectivity that allows them to access and share the same data and collaborate at the same time. Good design principles for environment-adaptive applications need to be derived in order to achieve rapid development for diverse platforms and quality of service with dynamically changing resources.

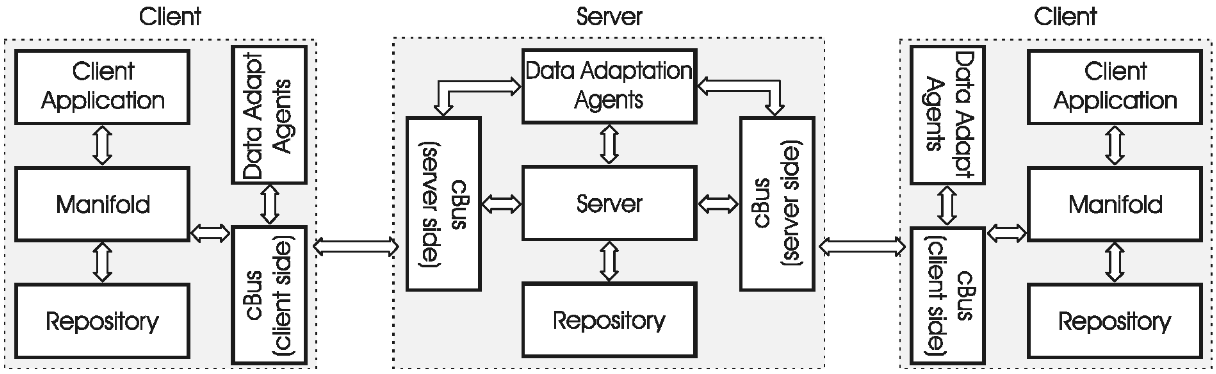

Fig. 1. Overview of the system architecture.

2. Related work

Collaboration in heterogeneous environments is a new research area and only few papers have been published. Among these [1],[3],[5],[6] are included. Heterogeneity is defined here in terms of platform capabilities, rather than operating system or programming language.

The MASH project [1] introduced application-level proxies for adaptation between heterogeneous clients communicating over heterogeneous networks. The authors applied their architecture to PDA-based Web browsing and to collaboration between desktop and PDA clients. In their system, a PDA essentially acts as a remote display for a proxy that maintains shared application state, making support difficult for disconnected client operations.

It is important that client devices are not just remote displays, i.e. the client should maintain application state, since a key requirement is that the system supports disconnected operation. Wireless links often provide only intermittent connectivity, so it is desirable to maintain a local state to enable the user to continue the work while disconnected, and to re-synchronize when the client gets opportunity to reconnect.

The XWeb project [5] addressed the problem of interacting with services employing a variety of interactive platforms. Since multiple users can simultaneously interact with a service, XWeb also supports collaboration. The key to the support for heterogeneity is that service and client implementations are independent of each other and of the particular mode of interaction that any user might chose, provided that they conform to the XWeb Transport Protocol (XTP). XWeb requires one-to-one correspondence between the data representations at the collaborating sites. This requirement prohibits the use of data compression and abstraction, which may be necessary for a scalable support of heterogeneous platforms. XWeb also does not support data adaptation for quality of service (QoS) management.

The research reported here is a continuation of our prior work [3]. It extends and refines the ideas presented therein.

3. System architecture

The system comprises four main components (Fig. 2): the repository, the collaboration bus with the data adaptation agents, the Manifold framework for developing collaborative applications and the task-specific application. The design of the server is relatively simple in the case where we have a single central server. It

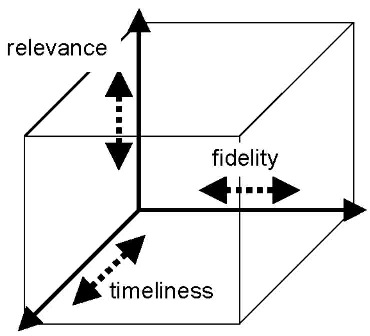

Fig. 2. Dimensions of data adaptation for quality of service (QoS) management.

just reflects messages (commands) between the clients and maintains the main repository.

Distributed Repositories

The global repository is situated centrally at the server site and stores all the shared data objects. It is the only repository for which persistence is currently supported. The repositories at the clients store the objects of interest to the local user and allow offline work. They are synchronized with the global repository when the connection between the client and the server is (re-)established [2]. In this way the same system can simultaneously support both synchronous and asynchronous collaboration. This is important when the collaborators work in different time zones or for other reasons cannot participate in synchronous sessions (e.g., unreliable wireless connections).

Manifold Framework and Applications

To facilitate the development of collaborative applications for heterogeneous computing environments, we have defined the Manifold framework [3]. The framework is based on the Model-View-Controller design pattern and consists of the base package with Java interfaces for the basic structure of an application. Default implementations for some of these interfaces have also been provided to help develop applications using the framework.

4. Data representation and adaptation

We believe that in most collaborative scenarios the users are mainly interested in data. The user wants to share data, whether synchronously with collaborators or asynchronously and alone from a different platform. Consequently, the problem of interoperating device-tailored applications is reduced to the problem of managing consistency of the data structures (application state) across different devices.

We use eXtensible Markup Language (XML) as a generic means for information representation and exchange. We expect that most of the Web data will be exchanged in XML and most of the collaborative applications will benefit from choosing this data format.

4.1. Intelligent data replication

Intelligent data replication aims at adapting the shared data to conserve network bandwidth or match the device's computing capabilities. Important dimensions of data adaptation are relevance, fidelity, and timeliness (see Fig. 1), where: (i) *relevance* is determined by user's interests and priorities; (ii) *fidelity* is dictated by computing platform's capabilities; and (iii) *timeliness* is determined by the requirements of the task. Other dimensions include modality, reliability, and security of data, but our focus here is on the above three.

In our framework, the user provides the relevance information to state his interests, the device and the application determine the fidelity of the data, and timeliness issues time constraints to data delivery, set by the requirements of the tasks. Logical evaluation of data with respect to the three dimensions results in a priority. This priority represents the relevance of data to the user in a certain environmental state. The more relevant the data is, the higher its priority will be, and the more certain it is to be sent. With different priorities for levels of details we can select at what level of detail the data should be sent. Sorting data by priority before sending addresses the timeliness: high priority data is sent immediately and low priority data is sent later when network conditions improve. The rules can also be set up to aggregate the data to send only updates about higher-level objects in a hierarchical data structure.

Interactions between the client and server components in the process of data adaptation can be seen in Fig. 3. When a user wants to modify a data object, an update command is sent from the client application to the server to update the global repository. A command will then notify other users who are subscribed to changes to the global repository and they will update their local repository. A *data adaptation agent* decides on what changes to notify the subscribing users about.

We introduce a selective and prioritized replication algorithm with dynamic filtering [7]. The criteria used for filtering depend on the environmental variables, and are therefore dynamic.

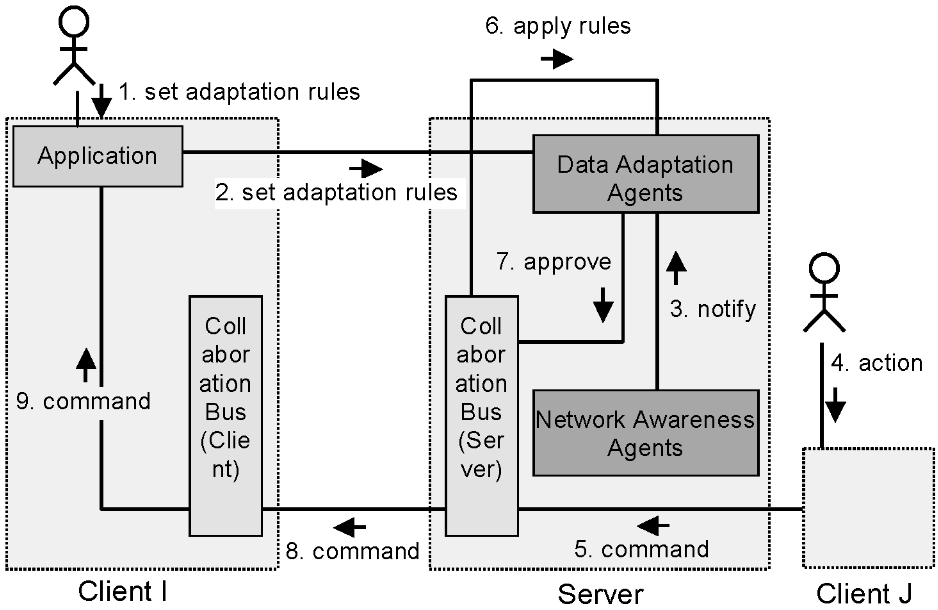

Fig. 3. The interactions between the framework components in the process of data adaptation to the context.

When these variables change, the selection criteria have to adapt to the situation. Fig. 4 shows the measured overhead of prioritized (conditional) data replication.

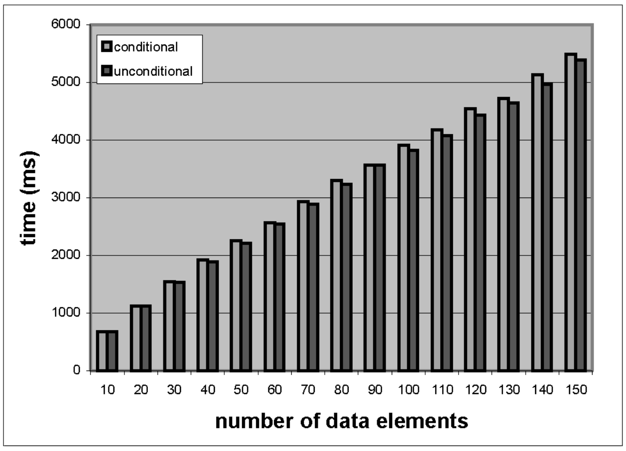

Fig. 4. Overhead of conditional replication.

4.2. Maintaining state consistency

The *application state* consists of all application's data structures that are relevant for the process of sharing. An example of application state is what is usually called *scene graph* in two-dimensional and three-dimensional graphics applications. It is a hierarchical description of scene objects, with scene objects, their parts, and parts of their parts represented at different levels of abstraction. Other examples include conceptual knowledge representation using relationships such as inheritance, spatial or temporal proximity, etc.

Suppose that two remote sites run an application tailored to each site's computer platform. As the users interact with the application, the application state transitions must be consistent to provide meaningful collaboration. The following diagram represents the problem. Let us assume that the application states at the two sites are consistent at a time t. The mapping between them is denoted as "map1".

$$[\text{ site1 state}(t)\ ] \xrightarrow{\text{map1}} [\text{ site2 state}(t)\ ]$$

$$\downarrow\text{op1} \qquad\qquad\qquad \downarrow\text{op2}$$

$$[\text{ site1 state}(t+1)\ ] \xrightarrow{\text{map2}} [\text{ site2 state}(t+1)\ ]$$

At the time $t+1$, the user interacts with the application at site 1 ("op1") and it transitions to a new state. To maintain consistency, site 2 must also transition to a new state which is consistent with that of site 1. We could either apply an operation "op2" to the old state of site 2 or a mapping "map2" to the new state of site 1. We assume that it is always more efficient in terms of network traffic to compute "op2" at site

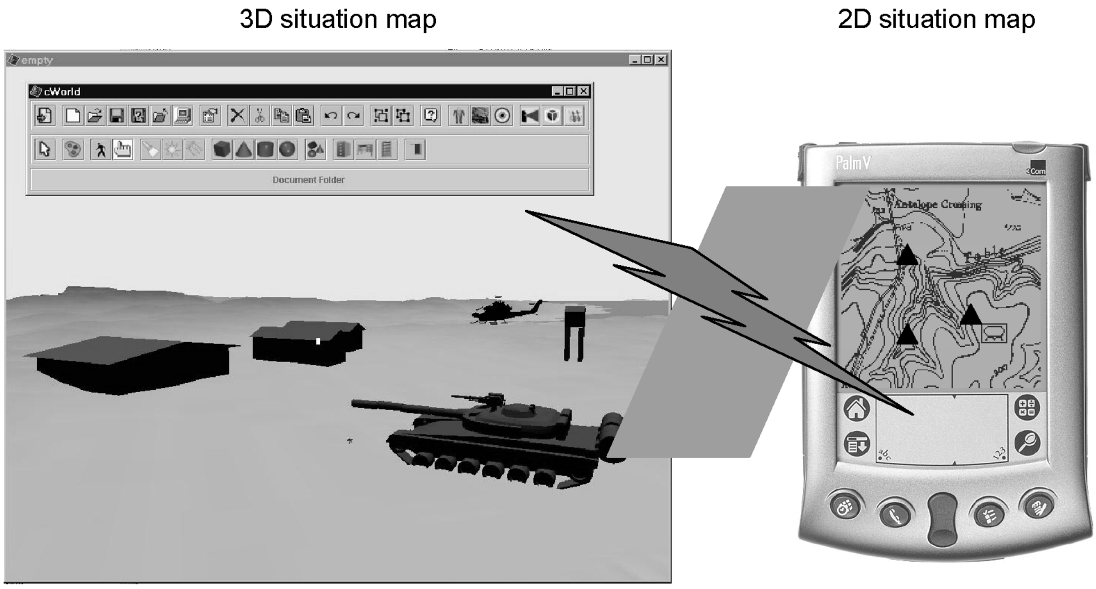

Fig. 5. Example application of mission planning on a situation map. The user on the left employs a high-end virtual reality workstation and the user on the right employs a Palm PDA.

1 and send it to site 2, rather than to compute the new state of site 2 using "map2" and then transmit the entire new state to site 2. This may not always be true, but the solution can easily be generalized.

If the platforms were homogeneous, the operations "op1" and "op2" would be identical and could be directly distributed across the collaborative applications. Due to platform-specific differences in state representations, "op1" first needs to be transformed to adjust to different domains. This transformation allows the interoperation of the applications.

Another important issue is how the mapping "map1" is determined, i.e., the issue of *defining* and/or *checking* semantic consistency. For this we are running human factors investigations to determine the rules for tailoring the data representation to fit platform capabilities. The set of rules generates "map1" mappings and needs to be checked for consistency. Once the rules are determined, the system needs to *maintain* semantic consistency. We derived an algorithm which guarantees that the state consistency will be maintained *assuming that the rules are consistent* [4]. Given the mapping "map1" and the operation "op1", the algorithm derives the mapping "map2" or the operation "op2" such that the application states remain consistent as they transition in response to users interactions.

5. Example Application

We developed two versions of a mission planning application: one that renders 2D graphics and the other that renders 3D graphics. The 3D version runs on a high-end workstation with Window VR virtual reality display. The 2D version runs on a PalmPilot V handheld connected to the Internet via an OmniSky cellular modem. Both versions support operations such as view a map, zoom in and out, place, move, rotate and delete objects (icons), freehand drawing, object property modification, etc. Rendering on Palmscape is monochromatic, as the J2ME Connected Limited Device Configuration (CLDC) does not support color displays yet.

The architecture shows significant reusability. More importantly, there is a great degree of design reusability. The basic infrastructure and application logic (design) are the same across platforms. The main difference is in the data: the data amount, types, fidelity, accuracy, etc., may be different across different platforms. Regarding the code, the processing machinery remains the same and most of the Java classes are reused across different platforms. The main difference is in the presentation and user interface code.

6. Conclusions

Computing platforms are rapidly becoming more heterogeneous. Our work provides a scalable architecture for the development of collaborative applications adapted to disparities in computing and communication capabilities of the collaborative participants. Collaborators share the same or semantically equivalent data, but view very different displays depending on their computing capabilities. The main characteristic of the framework presented here is extreme scalability allowing use on a wide spectrum of platforms, ranging from handheld access devices to high-end graphics workstations. It scales to support application logic with varying complexities of behaviors and visualizations with varying degrees of realism and media richness.

Acknowledgements

The research reported here is supported in part by NSF Grant No. ANI-01-23910, US Army CECOM Contract No. DAAB07-02-C-P301 and by the Rutgers Center for Advanced Information Processing (CAIP).

References

[1] FOX, A., GRIBBLE, S. D., CHAWATHE, Y., AND BREWER, E. A. Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives. *IEEE Personal Communications (Special Issue on Adapting to Network and Client Variability)*, 5(4), Aug. 1998.

[2] IONESCU, M., KREBS, A. M., AND MARSIC, I. Dynamic Content and Offline Collaboration in Synchronous Groupware. *Proc. 3rd Int'l Symp. on Collaborative Technologies and Systems (CTS 2002)*, San Antonio, TX, Jan. 2002, pp. 201–206.

[3] MARSIC, I. An Architecture for Heterogeneous Groupware Applications. *Proc. 23rd IEEE/ACM Int'l Conf. Software Engineering (ICSE 2001)*, Toronto, Canada, May 2001, pp. 475–484.

[4] MARSIC, I., SUN, X., AND CORREA, C. Maintaining State Consistency Across Heterogeneous Collaborative Applications. Submitted for publication.

[5] OLSEN, D. R., JEFFERIES, S., NIELSEN, T., MOYES, W., AND FREDRICKSON, P. Cross-Modal Interaction Using XWeb. *Proc. ACM User Interface Software Technologies (UIST'00)*, San Diego, CA, USA, Nov. 2000, pp. 191–200.

[6] TANDLER, P. Software Infrastructure for Ubiquitous Computing Environments: Supporting Synchronous Collaboration with Heterogeneous Devices. In: G.D. Abowd, B. Brumitt, S. Shafer (Eds.) *Proc. Ubicomp 2001: Ubiquitous Computing, Third Int'l Conf.*, Atlanta, GA, USA, Springer-Verlag LNCS-2201, Sep./Oct., 2001, pp. 96–115.

[7] VAN DER POEL, B., TAN, Y., KREBS, A. M., AND MARSIC, I. Prioritized Replication With Dynamic Selection For Mobile Environments. Submitted for publication.

Received: June, 2002
Accepted: September, 2002

Contact address:
Ivan Maršić
Department of Electrical and Computer
Engineering and the CAIP Center
Rutgers University
Piscataway, NJ 08854-8058 USA
e-mail: marsic@ece.rutgers.edu

IVAN MARŠIĆ is Assistant Professor of electrical and computer engineering at Rutgers University. He received the B.S. and M.S. degrees in computer engineering from the University of Zagreb, Croatia, in 1982 and 1987 respectively, and Ph.D. degree in biomedical engineering from Rutgers University, New Brunswick, NJ, in 1994.

Prof. Maršić's research interests are in mobile computing, groupware, computer networks, and multimodal human-computer interfaces. His research in these areas is supported by grants from NSF, DoD, and Industry. He has authored over 80 journal and conference papers, one book and 3 book chapters. He is a member of IEEE, ACM, IFIP WG2.7/13.4, and International Academy for Advanced Decision Support and has been a consultant to industry and government.