# SoftBridge: An Architecture for Building IP-based Bridges over the Digital Divide

John Lewis, Bill Tucker and Edwin Blake
Collaborative Visual Computing Laboratory (CVC), University of Cape Town
Broadband Applications and Networks Group (BANG!), University of the Western Cape
{jlewis,edwin}@cs.uct.ac.za, btucker@uwc.ac.za

*Abstract*— **Users and computing equipment have varying capabilities that include the abilities to exchange text and audio, engage in high bandwidth communication, etc. As the technological options increase, effective communication between the different media has be able to bridge across these options: the bridges have to take into account network quality, end-user equipment capabilities as well as user preferences and capabilities. Our research proposes a Soft-Bridge, an abstracted communications platform for the construction of applications across disparate capabilities found in various Digital Divides. We outline the architecture and the requirements that the SoftBridge has to fulfill. An approach and some initial work is described.**

## I. INTRODUCTION

According to the International Telecommunications Union (ITU), about 4 million land lines exist in South Africa and only 20% of South Africans have cellphones. Even when Digital Subscriber Lines (DSL) roll out commercially in South Africa, the large majority will still not have access to state of the art applications running on powerful computers over high bandwidth networks. The bulk of Internet connectivity over the Internet Protocol (IP) will remain over cellphones and low end equipment connected via the Public Switched Telephone Network (PSTN). More and more, these users are precluded from participating in Collaborative Virtual Environments (CVEs) because these traditionally require expensive VR equipment (head mounted displays, dual-ported 3D graphics cards) and fast network connections.

Low end users would benefit greatly if they could be included in such environments. This could be achieved by either upgrading capabilities and bandwidth to match the content, *or* downgrading the content to match capabilities. We are concentrating on another approach, of adapting the content and its interface(s) to user capabilities.

By *capabilities*, we do not only refer to computer hardware and software, but also the capabilities of the user. For instance, if a user is blind, then the system would only deliver audio, or translate text into speech. And if they were deaf, speech would be converted into text. As another example, we can extract human communication from an information and graphics-rich CVE and bridge it to a low end user. Consider a text chat tool inside an immersive Virtual Reality (VR) CVE with an IP bridge to a text chat tool running in a web browser. The content (text) and endpoint (HMD and CRT) abilities remain, but the interfaces are bridged. User preferences and profiles can also be taken into account. For instance, if the sender is male, he may prefer that messages from him are read out in a male voice.

This adaption process can be performed by a system we call a "SoftBridge". A SoftBridge enables us to build applications that bridge vastly different access equipment (telephones, cellphones, handheld MobileIP devices, laptops, personal computers, HMDs) to communicate seamlessly using various communications media (text, voice, video, VR) without regard to underlying mechanics of the process.

The SoftBridge relies on the abstraction of communications mechanisms such as text, voice, video and VR. Just as a softswitch seamlessly integrates a variety of circuit switched and packet switched signalling, and transport protocols, the SoftBridge adapts, or bridges, a variety of communications mechanisms and tools. With the SoftBridge sitting on top of a softswitch, we can achieve converged services in the spirit of [1].

## II. COMMUNICATION TOOLS

### A. Existing Asynchronous Systems

Bridges between various communications mechanisms are quite common in today's converged environment. The most notable and useful systems are those used in the wireless world, like Wireless Application Protocol (WAP) and the Short Message Service (SMS). However, these systems tend to be non-realtime and somewhat asynchronous.

### B. Unified/Universal Messaging

Unified/Universal Messaging (UM) is the popular name given to most of the systems that attempt to allow different media to be accessed in a common way. These normally allow single point access to emails, faxes, SMSes and voice mail. The fundamental problem with these systems is that that are all either asynchronous, or at best semi-synchronous when Observed Response Time (ORT) is small.

### C. JabCast Secure Realtime Communications

According to their web site, "The JabCast Secure RC is the first technology that allows for realtime interactive text, file, and document exchange in a completely secure environment via the Internet using an array of Operating Systems." [2] However, after close examination of the specification, it is apparent that JabCast only allows real-

time communication using short messages, and still transfers other media asynchronously. No content conversion or adaption is attempted by this system.

### D. Instant Messaging

Instant Messaging (IM) systems, eg. Microsoft Messenger and AOL Instant Messenger, allow either synchronous voice or synchronous text, but the two are not integrated. You may use either voice or text, but not both. This is a serious limitation for allowing ubiquitous access to such systems, as some users may have both audio and text capabilities, and others may only have one of the two. An advantage of IM, though, is that the exchange is perceived as being synchronous as opposed to SMS where one is never quite sure if the other person received the message or not.

### E. Web Boards

Web boards allow for synchronous, semi-synchronous and asynchronous communications, depending on how they are used. For example, an online classroom may function somewhat synchronously during an agreed upon lecture time, and will be accessed in asynchronous mode outside the allotted hour. It is quite trivial to add voice to a web board, yet they are currently mostly text based. Threading the voice is also possible, and a mixed voice/text chat tool is currently under development. We intend to migrate this tool to be built over the SoftBridge architecture.

## III. The SoftBridge Architecture

We propose an architecture based on the "Data Flow" model of computation, consisting of the following components:

1. Endpoints
2. Content Adapters
3. Control/Location/Billing

Information will flow from endpoints through adapters (getting converted if necessary), and out to destination endpoints. The process will be controlled from the controller component that will also take care of location (Naming/Directory services) and billing. This flow is shown in Figure 1.

### A. Endpoints

Endpoints are the locations where humans interact with the system, and typically consist of devices such as personal computers, cell phones, telephones and wireless handhelds. These endpoints all have native IP functionality, or can be connected via a gateway, eg. an H.323 [3] or SIP [4] gateway.

### B. Adapters

Adapters convert information from a form produced by one endpoint into a form consumable by another endpoint. For instance, if endpoint A produces speech, and endpoint B consumes text, adapter AB will convert speech to text. These adapters can be located at the endpoints, or somewhere in the network in between. Ideally, we would like the location of adapters to be dynamic, allowing them to
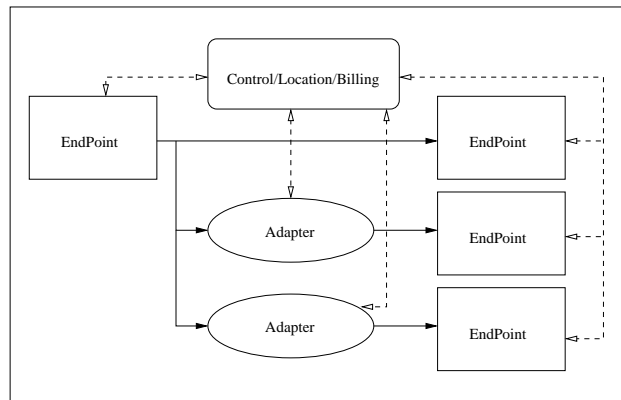


Fig. 1
The proposed architecture for the SoftBridge

be relocated as network conditions change to optimise performance and reduce overall ORT.

We do not intend to produce the software that does the actual conversion ourselves. Rather, we will make use of existing systems and develop drivers that will provide a common interface to them, as shown in Figure 2. For example, we are currently experimenting with Festival [5] and Sphinx [6]. In doing so, our goal is to provide the infrastructure to "plug in" tools as they evolve and improve.

We have identified the following conversions to be the most fundamental, and call them "basic" adapters :

1. Text to Speech
2. Speech to Text
3. Audio from Video

The following conversions and mechanisms have been identified as being more advanced, but still useful:

1. Virtual Reality to Video
2. Virtual Reality to Text tags
3. Text tags to VR
4. Language translators

### C. Control/Location/Billing

Control, location and billing (CLB) are combined into one component because of the closely coupled nature of these functions. In order to prevent the CLB component from being a single point of failure, it would be duplicated, and switched over if one of the instances fails.

### C.1 Control

The control function deals with the allocation of adapters and the synchronisation of components. It also coordinates the dynamic relocation of adapters as network conditions change. The network conditions are discovered by using QoS parameters such as jitter, latency and packet loss collected by a system like COLLAN, a distributed CORBA-based QoS collection and abstraction tool developed by Tania Paulse[7].

Endpoints register themselves with the Control component when they connect to the system. This allows the

Control component to keep track of all currently connected endpoints and simplifies billing.

## C.2 Location

In order for one endpoint to communicate with another, the source must first locate the target. To prevent illegal bypassing of the CLB component, an endpoint will verify with the control component that incoming connections are legitimate before accepting them.

## C.3 Billing

Ideally, we would like the billing function to make use of existing architecture like TINA to allow integration with the wider network and installed base. The billing function should make use of a unified Call Detail Record (CDR) structure. CDR formats are readily available for H.323 and SIP, and can be adapted to handle simple text transfer or any other type of data transfer.

## IV. METHODOLOGY

Because we will develop and trial applications built on the SoftBridge architecture, we will fuse two different but compatible research methodologies into a method for a critical perspective on ICT. The first methodology is the common approach of experimental computer science research [8]. This is based on the iterative development of research artefacts where the impact is assessed and the researcher uses the lessons from one cycle to improve the artefacts in the next cycle. This methodology applies to both the Soft-Bridge and its applications. The second methodology is that of *Action Research* which is now an accepted method in the field of Information Systems [9]. This method enables us to develop bridging applications relevant to targeted user communities and their needs.

## V. IMPLEMENTATION

We are implementing the system as a set of CORBA components using AT&T's OmniORB [10] broker, as shown in the UML diagram, Figure 3.

*Endpoints* with implement the `EndPoint` interface, and their references will be passed to the control component when they connect to the system.

*Adapters* will implement the `Adapter` interface, but this may be subclassed to provide more specific behavior.

*Drivers* implement the `Driver` interface. Drivers are owned by Adapters, and provide a common interface to underlying conversion software and API's.

*The Control/Location/Billing* component will implement the `CLB` interface, and will be the only component that endpoints are aware of at startup.

The Location function will make use of the CORBA Common Object Services Naming service (COSNaming) to locate endpoints and adapters. However, the endpoints will be prevented from accessing COSNaming directly. They will have to go via the CLB component to obtain object references.
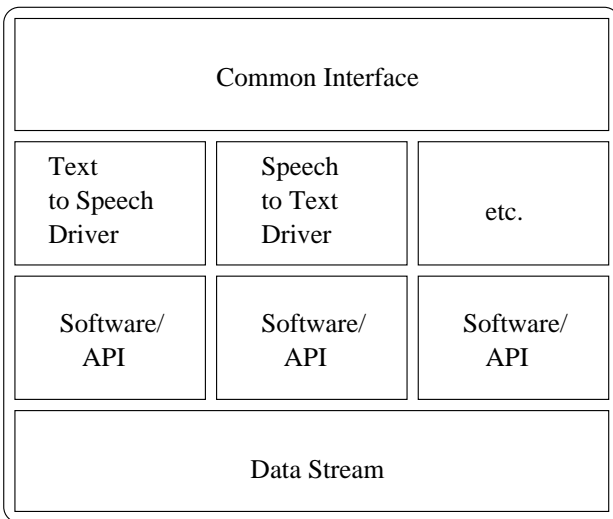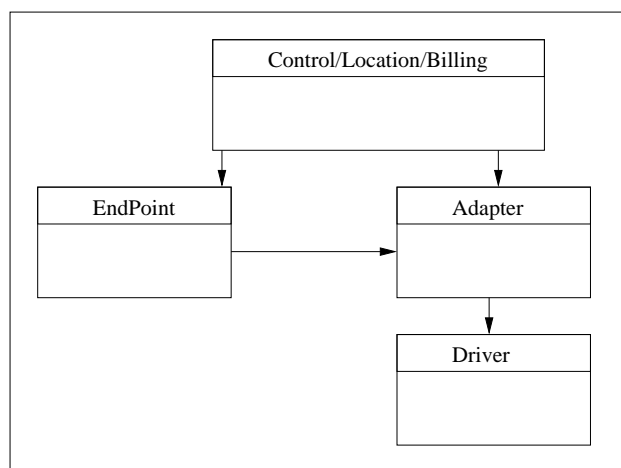


Fig. 2
THE PROPOSED DESIGN OF THE ADAPTERS



Fig. 3
THE CORBA COMPONENT SET

## A. First Prototype

We have developed an initial prototype that allows endpoints to connect to a server, and messages from an endpoint are broadcast to all other endpoints. Depending on how an endpoint is set up, the message is either displayed as text, or converted into speech. At this stage, the Adapters are co-located with the endpoints.

We are also currently implementing a speech-to-text adapter, which will allow speech-to-text conversion, followed by transmission of text, followed by text-to-speech, avoiding speech transmission entirely.

After we have completed these two basic adapters, we will implement a simple user interface, and perform simple pilot testing on a small number of users. From their responses, we will be able to decide on future developments and modifications to the prototype.

## VI. Evaluation

We will evaluate and test the system from both a technical performance standpoint, and a user/usability standpoint.

### A. Technical/Performance

The technical evaluation will involve measuring metrics like jitter, latency, and robustness. We will also be able to evaluate the effectiveness of the Dynamic Adapter Relocation system by generating traffic on the network and also testing on a normal network under load. We are particularly interested in the idea of mobile adapters, packaged as CORBA components to achieve network efficiency.

### B. User/Usability

In order to evaluate the usability of the system, we are going to construct three user applications to act as test cases. The proposed applications are :
1. Deaf Telephone [11] — Allows deaf users to communicate with hearing users.
2. VR chat — A tool that allows users in a VR environment to chat with users in text and voice environments.
3. Voice-Text web board — An Internet web board that will allow voice-text integration. This will be integrated and trialed with the Knowledge Environment for Web-based Learning (KEWL)system, a distance learning framework at kewl.uwc.ac.za.

## VII. Work Plan

We are developing the SoftBridge in parallel with the applications, as these are separate research efforts in themselves. They will be trialed using Action Research principles to improve their usability and efficacy. Once the SoftBridge is complete, we intend to port the applications to use its facilities. After the applications have been ported to the SoftBridge, the entire system will be fault tested, performance tested, and finally released for user testing. At this point we intend continuing development on the prototype, and eventually extending it to form the working system.

## VIII. Conclusion

We will develop a system and framework that allows us to build collaborative applications for users with very different communications equipment and capabilities. This will enable us to start building bridges across the "Digital Divide" that will enable participants from both sides to collaborate and communicate in meaningful ways.

## References

[1] C. Gbaguidi, "An architecture for the integration of Internet and telecommunication services," *In IEEE Infocom '99, New York, March 1999. IEEE.*, 1999.

[2] "JabCast SRC Realtime Communications System, www.jabcast.com," .

[3] A. Karim, "H.323 and associated protocols," 2000.

[4] A.B. Johnston, "SIP: Understanding the Session Initiation Protocol," 2001.

[5] P. Taylor and A. Black and R. Caley, "The architecture of the the Festival speech synthesis system," 1998.

[6] X. Huang and F. Alleva and H.-W. Hon and M.-Y. Hwang and K.-F. Lee and R. Rosenfeld, "The SPHINX-II speech recognition system: an overview," *Computer Speech and Language*, vol. 7, no. 2, pp. 137–148, 1993.

[7] T.M. Paulse and W.D. Tucker, "A Framework and Toolkit for the Collection and Analysis of QoS Statistics for Voice Traffic in Next Generation Networks," *Proceedings of the South African Telecommunications, Networks and Applications Conference, Stellenbosch, 11-13 September 2000 (CD-ROM publication)*, 2000.

[8] National Academy of Sciences, *Academic Careers for Experimental Computer Scientists and Engineers*, National Academy Press, Washington, D.C., 1994.

[9] D. E. Avison, F. Lau, M. D. Myers and P. A. Nielsen, "Action Research," *Communications of the ACM 42, 1 (Jan. 1999)*, pp. 94 – 97.

[10] S. Lo, "The Implementation of a High Performance ORB over Multiple Network Transports," 1998.

[11] M. Glaser and W.D. Tucker, "Web-based Telephony Bridges for the Deaf," *Proceedings of South African Telecommunications Networks and Applications Conference (SATNAC 2001),Wild Coast Sun, South Africa. Sep 3-5, 2001 (CD-ROM publication)*, 2001.

SoftBridge: An Architecture for Building IP-based Bridges over the Digital Divide

John Lewis †, Bill Tucker ‡ and Edwin Blake †

† John Lewis is with the Collaborative Visual Computing Laboratory of the Department of Computer Science at the University of Cape Town.

He is currently working on his MSc degree, after completing an Hons. degree involving parallelising graphics algorithms on clusters of Windows NT workstations, using Distributed COM. This explains his current enthusiasm for CORBA. His research is sponsored by the Telkom/Siemens/THRIP Centre of Excellence in ATM & Broadband Networks and their Applications.

‡ Bill Tucker is with the Broadband Applications Network Group of the Department of Computer Science at the University of the Western Cape. His research is sponsored by the Telkom/Cisco/THRIP Centre of Excellence in IP and Internet Computing.

† Assoc. Prof. Edwin Blake heads up the CVC lab, and is supervising the project.