status report for

A Sun Sparcstation-Based Speech Data Collection Platform

LDC Subagreement 5-24431-C
ISIP Project No. 02-95

for the period of July 1, 1995 to September 30, 1995

submitted to:

Linguistic Data Consortium

441 Williams Hall
University of Pennsylvania
Philadelphia, PA 19104-6305

submitted by:

Joseph Picone, Ph.D., Associate Professor

Institute for Signal and Information Processing
Department of Electrical and Computer Engineering
Mississippi State University
Box 9571
216 Simrall, Hardy Rd.
Mississippi State, Mississippi 39762
Tel: 601-325-3149
Fax: 601-325-3149
email: picone@isip.msstate.edu

EXECUTIVE SUMMARY

In the second phase of this project, we accomplished two goals: developed a detailed understanding of Sun's XTL Teleservices software platform, and trained our programmers on the relevant aspects of applications programming under Unix. As we anticipated, acquisition of a stable hardware platform from Linkon Corporation is becoming problematic. We have adjusted our efforts accordingly to try to offset the fact that both the hardware (an Sbus card) and the software interface (a device driver and C run-time library) are delayed.

The Linkon FS-3000 system, included in our proposal for this project, never really made it to product. The reason we were given was that the performance wasn't there — the board could only handle one channel, which made its price/performance point noncompetitive. The FS-3000 could also not do the teleconferencing application vital to SWITCHBOARD-style data collection. The FS-3000 was obsoleted and a new board, the FS-4000 has been developed. This board contains 4 DSPs and an analog telephone interface daughter card. It supports analog telephones in its default configuration. With the addition of a Newbridge Microsystems T1 card (remove the analog telephone daughter card and connect the T1 card directly into the Linkon board), the board can handle digital telephone lines in a software-transparent fashion (details of this remain to be seen).

The main problem with this "new" approach is that this system costs about $14K: $3K for the Linkon board, $1K for a Linkon software interface to the board (required), and about $10K for the Newbridge card (this is not a firm price yet). It appears the Linkon system can handle worst case one channel per DSP, which means a single card system with 4 DSPs can handle four telephone lines. It is possible that a single card might be able to handle 8 channels (the engineers have been rather vague about this). With multiple cards, however, a full T1 span can be covered, though it will be at an noncompetitive price (worst case, approximately $30K just for the T1-related hardware).

Our attempts to get an evaluation copy of this system failed even though prior discussions with Linkon led us to believe this was desired by both parties. Since the cost of the system is significantly higher than what was budgeted, we are in a bind. We have adopted the following strategy:

- purchase the analog telephone line portion of the system ASAP
- develop our applications based on analog phone lines
- delay acquiring a T1 line to save costs and not waste money on an unused line
- acquire the T1 portion of the system and complete the project

Linkon has promised to ship us a software interface by Oct. 10, and the hardware by Oct. 15. If the analog system is truly software compatible with the digital system, this plan should minimize cost (to offset the money we need to recover to buy the T1 card), and more importantly, minimize risk. If Linkon can't deliver the analog board, then we will certainly not pursue the digital board.

In parallel, we have studied Sun's XTL software platform, which is an alternate software interface that will support the Linkon board. This software costs $1.3K per CPU, and appears to be overkill for what we need to do. We hope to adopt its good features in our public domain version of an interface to the data collection platform.

1. HARDWARE STATUS: LINKON CORPORATION'S FS-4000

We have been tracking Linkon's progress in developing T1 interfaces for various platforms for several years. They had successfully produced a T1 card for a PC that was used by Ron Cole at OGI for several projects. This card lacked the teleconferencing capability that was required for this project. The Linkon FS-3000 system, originally included in our proposal for this project, was their first attempt to produce a Sun Sbus-based card. The major constraint in an Sbus design is physical space — Sbus cards invariably don't have enough space to put everything you might want in a subsystem (this has been a problem with many products I have seen). This is also Linkon's first attempt at supporting the Solaris 2.X operating system.

The FS-3000 product was originally planned to provide multiple voice channels per board, and to leverage the Newbridge Microsystems T1 card. Discussions with Linkon last Spring led us to believe Linkon was very interested in having their platform penetrate the ARPA market. We were considering the development of a limited number of systems, on the order of 3 to 6, available to ARPA-related vendors (such as LDC) for minimal cost. At the time, Linkon understood their price was comparable to the Intervoice system, and was very interested in seeing their system supplant the Intervoice system.

Another piece to this puzzle was the announcement last Spring that Linkon and Sun had reached a strategic agreement to co-develop a telephone services platform based on Linkon's products. Linkon, in return, would support Sun's XTL Teleservices software interface. This seems to have changed the game considerably. I suspect this agreement is now driving a large part of Linkon's R&D and strategic vision. Perhaps there is no longer a need to get visibility within the ARPA community?

At some point this summer, it became clear that Linkon's FS-3000 system would not provide the performance required to handle a T1. We were told the official reason was that one board could only handle one channel, which made its price/performance point noncompetitive. The FS-3000 could also not, therefore, do the teleconferencing application vital to SWITCHBOARD-style data collection. The FS-3000 was obsoleted and a new board, the FS-4000 has been developed. Of course, we learned this only this past month (probably after they had settled on a new strategy).

This board contains 4 DSPs and an analog telephone interface daughter card. It supports analog telephones in its default configuration. With the addition of a Newbridge Microsystems T1 card (remove the analog telephone daughter card and connect the T1 card directly into the Linkon board), the board can handle digital telephone lines in a software-transparent fashion (details of this remain to be seen). The A/D quality of the daughter card is supposed to be excellent (they seem to recognize the importance of this aspect of the system). This actually makes the system more attractive to some potential users (who are still interested in the analog telephone market).

The main problem with this "new" approach is that this system costs about $14K: $3K for the Linkon board, $1K for a Linkon software interface to the board (required), and about $10K for the Newbridge card (this is not a firm price yet). It appears the Linkon system can handle worst case one channel per DSP, which means a single card system with 4 DSPs can handle four telephone lines. It is possible that a single card might be able to handle 8 channels (the engineers have been rather vague about this). With multiple cards, however, a full T1 span can be covered, though it

will be at an noncompetitive price (worst case, approximately $30K just for the T1-related hardware).

Our attempts to get an evaluation copy of this system failed even though prior discussions with Linkon led us to believe this was desired by both parties. At first, they were going to send us an FS-3000 as an interim solution, and get us a pre-release of the software, so that we could begin prototyping the system. After several delays, I believe they probably would have trouble delivering this system, and really want us to focus on the new product.

Since the cost of the system is significantly higher than what was budgeted, we are in a bind. We have adopted the following strategy:

- purchase the analog telephone line portion of the system ASAP
- develop our applications based on analog phone lines
- delay acquiring a T1 line to save costs and not waste money on an unused line
- acquire the T1 portion of the system and complete the project

Linkon has promised to ship us a software interface by Oct. 10, and the hardware by Oct. 15. If the analog system is truly software compatible with the digital system, this plan should minimize cost (to offset the money we need to recover to buy the T1 card), and more importantly, minimize risk. If Linkon can't deliver the analog board, then we will certainly not pursue the digital board.

2. SUN'S XTL TELESERVICES 1.0 SOFTWARE PLATFORM

In parallel, we have studied Sun's XTL software platform, which is an alternate software interface that will support the Linkon board. This software costs $1.3K per CPU, and appears to be overkill for what we need to do. This software environment is geared towards high-end telecommunications services providers, and leans heavily towards multimedia technologies, a multiplicity of communications interfaces, and GUI-based interactive applications.

XTL first appeared under the name SunXTL in mid-1994. At that time, SunSoft decided not to market this software, and put it on the shelf. Sometime later, the hardware side of Sun decided to pick it up and base its enterprise systems business thrust on it. It has recently emerged as XTL Teleservices, and seems to have a new lease on life. We feel that its long-term future is questionable at best (several such products planned to support digital telephony have died in th not-too-distant past).

XTL teleservices is an object-oriented C++ based platform developed by Sun Microsystems for desktop call-processing applications. XTL was designed to provide an application programmer's interface (API) for the development of desktop applications, transparent porting between analog, ISDN, and ATM based technologies, basic building blocks of call processing (e.g. DTMF and silence detection), and other specialized services such FAX, modem, and video capabilities. The true power of XTL lies in its third-party extensible, distributed-object model. XTL is dynamically extensible to allow it to utilize new communication technologies and protocols as they become available.

The XTL architecture consists of the user application, the XTL API, the XTL provider interface, the service provider driver, and the service provider hardware. An overview is given in Figure 1.

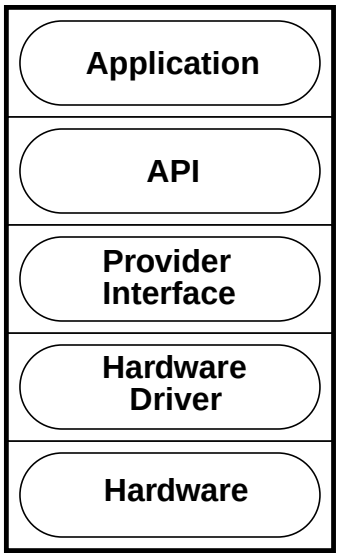

Figure 1. An overview of the hierarchy of interfaces involved in Sun's XTL teleservices platform. To take advantage of XTL, software should utilize the API level. It is our understanding that Linkon will still require the purchase of the driver-level software if their hardware is to be used with XTL (which means the software portion of the XTL system involves the purchase of about $2.5K per CPU in software).

The inner workings of the device drivers are hidden from the application by the API. Applications use the API to call C++ XtlCall objects. An XtlCall object generally represents a single phone call, and has methods that perform basic call functions such as querying the state of the call or changing the call's state. The XtlCall object also allows the application to access the data streams associated with the call. The extent to which the application can access and manipulate the data streams is dependent on the provider and the type of call.

The XTL provider interface contains the XTL Media Platform Interface (MPI) library. The MPI isolates the provider from the intricacies of the application and API by supplying the provider with a messaging system for commands and a common mechanism for making calls available to the application.

Any XTL application use a similar skeleton of code. The programmer needs to (1) derive subclasses from the XtlProvider and XtlCall classes, (2) implement event notification methods for the subclasses, and (3) bind the events to the command methods and new objects. This style is closely related to the standard X Window programming style. In Figure 2, we show a programming example from a Sun-supplied demo.

This code exhibits every important aspect of an XTL application, in brevity. It includes the standard XTL header files, xtlprovider.h and xtlcall.h. The program then creates a subclass (MyProvider) from the basic XtlProvider class. MyCall is an event handling routine that passes the event and kvl variable to DirectAudioCall until the event is DISCONNECT_EVENT. The main program basically just puts the program into a dispatch loop, passing control to the event handlers.

There are drawbacks to using Sun's XTL platform as the core of a telephony application system. First of all, XTL is not free software, it is licensed by host. Also, it relies on Sun tools, making

```
#include <stdio.h>
#include <xtl/xtlprovider.h>
#include <xtl/xtlcall.h>
[...]
class MyProvider: public XtlProvider {
  public:
  MyProvider(Exception*, XtlString, XtlAddress);
[...]
void MyCall::event_ind(CallEvent event, XtlKVList& kvl) {
  // preserve DirectAudioCall behavior
  DirectAudioCall::event_ind(event, kvl);

  // exit when call is disconnected
  if (event == DISCONNECT_EVENT)
     exit (0);
}
[...]
void main(int argc, char* argv[]) {
[...]
while(1)
  d.dispatch();
}
```

Figure 2. An example of an XTL application that shows the similarity to X application programming.

portability a problem. Furthermore, if XTL is not installed in the exact location on the system that Sun intended, it is nearly impossible to use. XTL requires Solaris 2.4 to run. Its availability on other platforms is not clear — especially non-Solaris platforms. XTL is an immense amount of code, being able to handle a great assortment of multimedia systems. XTL includes many features that would never be used by the LDC data collection system.

Due to the drawbacks of XTL, we plan not to use it as part of our telephony environment. Like XTL, our approach will be written in C++, object-oriented, and somewhat hardware-independent. The designers of the XTL system created a good general form of implementation for the system, so we will adopt many of the basic premises that XTL uses (though you could argue these features are inherent in any X-based GUI programming tool). The advantages of our code over XTL will be portability and less strict installation standards, as well as being shareware. We will create a smaller, robust, more dedicated system to handle a smaller range of applications more efficiently.

3. X WINDOWS APPLICATION PROGRAMMING TRAINING

One additional minor point about API's under Solaris. We have concurrently trained the two staff members working on this project in the development of applications using tcl (and its derivatives). They are also versed in several other GUI development tools under X. Presently, in conjunction with the LDC JEIDA Corpus project, we are studying the development of object-oriented interfaces to X windows application programming to support the application development portion of the code. We now have prototype code that implements some of the low-level device independence protocols. These same programmers are now developing a better understanding of interrupt programming under Solaris and C++.

4. NEAR-TERM PLANS

Our major concern at this point is whether we will ever see a stable hardware platform. A conference phone call is planned for the week of October 1 to settle this issue with Linkon. Assuming we pass this point, we should have hardware functioning by mid-October. This will most likely still leave the issue of system cost unresolved.

By delaying acquisition of the T1 line, we are saving approximately $750 per month. We hope we can use this money towards the purchase of the Newbridge T1 card. We will make every effort to squeeze this into our budget. However, we were told that Newbridge doesn't do evaluations either, so the likelihood of getting an evaluation copy is low. We are hoping we can strike a bargain with our available funds.

If the Linkon system supports both analog and digital interfaces transparently, we should only need a small number of months to debug the T1 portion of the project. We have discussed this at-length with Linkon, and they can't see any reason it shouldn't be "plug-and-play." In any event, from a functionality level, we should be able to demonstrate and debug the entire system using analog phone lines. We have three analog telephone lines in place, provided by our department at no cost, to support this evaluation. We expect to field a demonstration of the system at the time we submit the next status report.

Recent changes in the telecommunications structure at MS State might help us. We are negotiating with MS State's Telecommunications Department to see if they can provide us with a T1 off of the university telephone system at virtually no cost. This would save the money allocated for T1 line charges for the purchase of the T1 card. It appears this might be possible under the new university agreement, but we haven't received a final decision yet.

The next month of this project will be very critical to its long-term success. We will keep LDC advised on an immediate basis as soon as we receive any pertinent information.