

# The Listen Project: Progress and History

Aditya P. Mathur

Last Update: September 10, 2003 at BITS, Pilani, India

Information about the Listen and Jlisten project is available at:

<http://www.cs.purdue.edu/homes/apm/listen/html>

## 1 The system

The Listen system is intended to be a general purpose tool for the auralization of program behavior. Ideally it should be possible to auralize any type of program behavior using Listen.

### Versions of Listen:

Listen/C 1.0 (Dave Boardman with assistance from Bob Liu, Geoff Greene, and Jack Lowry): April 28, 1994. *This work was part of Dave's master's thesis. Dave joined HP, Atlanta, GA.*

Listen/C 2.0 (Cary Supalo, Sriram, Nate Nystrom, Katrina Pakan, Leo Rijadi, Vivek Khandelwal): May 30, 1995. *This work was part of Vivek's master's thesis. Vivek joined Nortel in Dallas, TX.*

Listen/C 3.0 (Cary, Sriram, Nate Nystrom, Katrina, and Leo) : December 19, 1995

JListen (Under construction by Ed Gilmore and Bryan Strait): Expected by December 15, 2002.

## 2 Progress

1. I have purchased a MAC IICi, a MIDI interface, Emu-Proteus/3, and Altech's Midi Pascal (total cost was approximately \$4500). I plan to build Listen on this system. A keyboard and some miscellaneous gear is yet to be purchased.
2. Elias Houstis feels that I should place all this equipment in the future SoftLab. I like the idea, it is yet to be approved by Rice.
3. Using demo.p file that came with the Altech Midi Pascal, I have learned a bit on how to program MIDI. I instrumented a program so that a user can listen to a loop in that program. The sound is initially turned off. But if the user clicks the mouse then the sound is turned on and one listens to a succession of G-E pairs of notes to indicate the beginning and end of the loop body. When the loop terminates a simple pattern (G-G-E-C) is played to indicate termination. I used the C major scale to auralize this loop.
4. A lot has happened since I last updated this file. Dave Boardman has joined Purdue in the fall of 1992. He started working on the Listen project. I am supporting him quarter time since

January 1993. He has developed a version of ATAC on the Ncube that can instrument programs with calls to generate a trace. When the instrumented program is compiled and executed on the Ncube, a trace file is generated. This file is transferred to Mac II. On the Mac it is played by interpretation through another program. The playback is via the Proteus/3. The system is wonderful but a bit inconvenient to use because of the trace file transfer and interpretation required. Dave did a wonderful demo to SERC affiliates on May 11 in the SERC lab. Dave is quite excited about the immense possibilities of using sound in program understanding or just for fun.

I started designing a language for specifying what kind of auralization is required from a program. I prefer to name it Listen-S (S for specification and LSL, Listen Specification Language for short). LSL will help in the specification of linguistic constructs that should be sounded and how they should be sounded. The language seems to be quite general in the sense that one may use it to write a complete score textually. When a program in Listen-S is interpreted a score is played.

5. Elias has placed an order for a Quadra/800 for this project..and also an Audiomeia II card. With this system we will be able to use voice and sound together and have a faster system to work with.
6. *September 8, 1993*: The Quadra 800 is now in place. Unfortunately Dave decided against using it. He prefers to program in a Unix environment and plans to use the Sun. I still plan to use the Quadra for the Listen system. We can at least port to Quadra whatever we do on the Sun. The Keynote MIDI software for the Sun has also arrived from AT&T. It is yet to be tested. We have a MIDI interface also, Kytronics, which is also to be tested.

The design of Listen Specification language (LSL) is complete. A tech report is being issued today. I gave a talk titled "A language to specify program-sound mappings". The talk was given as a part of CS 490S and CS dept colloquium. Jyoti also attended. She did not ask any questions..during or after. There were many many questions from students. One graduate student (I do not remember his name) has indicated that he would like to do his PhD in this area. I have advised him to begin with a Master's thesis. His work will supplement that of Dave's. Another student, Jack lawry, an undergraduate, has also indicated his intention to work.

The LSL tech report will be distributed to a large clientele. Dave has a list of over 20 people who have requested for a copy.

7. *October 4, 1993*: We had the first "expanded" group meeting on Thursday, September 28 at 4pm. Jack Lawry and Bob Liu, two undergraduates, have joined the project. They are doing this work as a part of their independent study course (2 credits each). I am expecting 12 hours per week work from each of them. Dave is the team leader and Aditya the advisor.

The meeting went on till 5 pm. Jack Lawry presented the data structure for the **occurrence**

list. More ideas were floated to improve the structure. Jack is scheduled to distribute the structure to all members Bob Liu and Dave Boardman discussed the tree structure generated by ATAC. Bob is now working on a module that will help in displaying a tree in a variety of textual (non-graphical) formats for debugging purposes.

Considering that this was the first “expanded” group meeting, it went off quite well. I would have liked more progress but I guess things will ramp up slowly.

8. *June 16, 1994*: There has been tremendous progress since January 1994. Dave organized the project very ably. As a result Lawry and Bob Liu have implemented LSL pre-processor and C-code instrumenter. Dave worked on it further and implemented more LSL features. Geoffery Greene has done a remarkable job of implementing a GUI. The GUI allows easy auralization. It generates LSL specs from user commands given through GUI.

Notify, atrack, and dtrack works and so does scoping. I have seen two great demos but only on small programs. Today is another demo in a few minutes.

Dave will leave us in about a month. He has a job with HP in their multimedia group. I do hope Dave had stayed over and done his PhD. Vivek Khandelwal has joined as full time researcher during summer. He will take over Dave’s job after Dave leaves. He plans to work on his Master’s thesis using LISTEN.

I plan to use LISTEN in CS 180 during Fall. This will be a nice experience and a test of the utility of using sound in classroom teaching of programming. Vivek will develop examples for me.

I have placed orders for several minor items so that I can do the demo in the classroom. The only problem I foresee is the availability of a SparcBook. The LISTEN system is NOT on the MAC. It is now on the Sun. The MIDI interface on the Sun is also working!

9. *November 12, 1994*: The following message was sent to me by Vivek:

From: Michael Pelz-Sherman <mspboss!zappa!mps@uunet.uu.net>  
To: mspboss!uunet!acm.org!sound@uunet.uu.net  
Subj: Review of ICAD '94

[deleted...]

For me, the highlights of the conference were: 1) Jonathan Cohen’s dem o of his “Out to Lunch” system for monitoring background activity, 2) Robert Stevens’ presentation of a system for providing blind readers with an “auditor y glance” at algebraic expressions, and 3) the presentation of two systems for program auralization: David Boardman’s LSL (Listen Specification Language) and

Dale Bock's ADSL (Auditory Domain Specification Language). These last two systems facilitate experiments with program auralization, or 'listening' to one's code. For example, entering and exiting a given function in a C program might start and stop a particular sound or sound sequence. Much work clearly remains to be done to determine the extent to which musical training is required before software developers can use sound effectively in this manner.

Unfortunately, both systems are currently only available for the C programming language. Both authors stated that porting their systems to other languages is quite feasible, however.}

10. *January 26, 1995*: The Spring 95 semester has begun. While flying to India I spend some time working out the R&D plan for this semester. The Listen group is the largest ever (but still no financial support). This semester we have Vivek supported through some left over SERC money, Katrina Pakan supported by Undergraduate Research Grant, Howard Chen is working just out of the love for this project, Nate Nystrom is continuing with us supported by UG research grant, Sriram is doing a CS 490, Cary Supalo is doing a CS 190 (1 credit), Leo Rijadi is doing a CS 490.

Vivek has already ported Listen to Sparc 5. Katrina is working on auralizing the output module of an application developed by SoftLab people (Elias' group). This is our first attempt to take Listen outside our own domain of applications. Sriram and Cary are working on modifying EMACS for use by a blind user. Cary will try it out and provide useful feedback. Leo will help Vivek in adding functionality to Listen. They are trying VDAP implementation first. Howard is enhancing the compile time GUI developed in Fall 94 by Geoffery Greene. Nate is continuing his work with runtime user interface. The interface a come a long way since Nate started working. He has already done his first demo.

I hope that an auralized emacs will be a nice tool. I hope to distribute it to other visually challenged users sometime in summer.

Dave and Vivek attended the ICAD'94 in Santa Fe, New Mexico. Vivek returned from the conference all excited. He met me at the Indianapolis Airport and we drove back together. He thought that our project had now made a mark in the auralization research community. There was one nice posting about Listen in the newsgroup from an independent ICAD attendee.

Geoff has got a nice job with Autometrics in Arizona. He did a wonderful job with Listen for about one year (1994).

The Listen paper submitted to ICSE 17 got rejected. Reviews were generally good but the paper did not make it. It has now been revised and submitted to COMPSAC'95.

The largest application prepared so far was the CS 352 lexan/parser auralization that I did with Listen. It was a wonderful experience. I auralized the compiler at various levels so that students could listen to the architecture. I then audio taped the sound output and presented it to the class with a nice storyline. It was more of a fun than something used as a teaching tool. Nothing has been published yet on this experience.

We hear from Dave once in a while. He does not seem to be enjoying his job. The multimedia group that he had joined wound up and he had to move to Atlanta from Boston. Dave and I have to still write the monograph on program auralization.

11. *June 6, 1995*: One more semester has passed since I updated this report. The Spring 1995 semester had the largest number of students working on the Listen project. While enroute to India in December 1994 I planned out the tasks for the Spring 1995 session. On return from India there was a meeting attended by Vivek Khandelwal, Nate Nystrom, Howard Chen, Katrina Pakan, Cary Supalo, Leo Rijadi, and Sriram Ramkrishna. Vivek was made the overall coordinator of the project, Chen would continue Geoff Green's work on compile time interface, Nate would continue his work on run time interface, Katrina would use Listen to auralize an application in Elias' softlab, Leo will assist Vivek in adding functionality to Listen, Cary will help in experiments with auralized emacs, and Sriram will auralize Emacs and conduct experiments with Cary. As a result of this effort, we now have Listen 2.0 made public and a master's thesis from Vivek. This is the second MS thesis on Listen..the first one being that by Boardman.

Vivek ably organized experimentation with Listen. He investigated the effectiveness of aural cues with and without visual cues. Results are as expected and appear in Vivek's MS thesis. Leo worked with Vivek and implemented the VDAP function of LSL. This is a useful feature for data auralization.

Nate did a wonderful job of programming the run time interface. The interface needs more work but in its present form it is a useful part of Listen. Chen began his work by removing errors from the compile time interface that was designed and programmed by Greene in Fall 1994. He then added new features to the interface. For example, functions call/return can be auralized using compile time interface. Both the run time and the compile time interfaces need more work.

Katrina started by identifying the application to be auralized. This was the front-end of a PDE solver in the soft lab. The idea was to auralize the front-end. Katrina dropped out in the middle of the semester due to some personal problems. This was a disappointment. I had expected a lot from her on this part of the project.

Sriram started very slowly with Emacs. Eventually he was able to auralize the cursor movement and search and replace functions of Emacs. He and Cary then did some preliminary and not well organized experimentation with the auralized emacs. Observations from this experiment suggest that Emacs auralized with non-speech sounds may be useful for blind users when used

in addition to speech sounds. Cary worked slowly but then he had registered in CS190 for only 1 credit. He surveyed the field of use of sound in programming environments. He write a brief report on it. Later he worked with Sriram and wrote his own report on his observations and feelings while working with the auralized emacs.

Vivek left for BNR, Dallas, Texas on June 3, 1995. Before leaving he made an ftpable file for Listen 2.0. We are now making it public.

I am now planning to ask for NSF funding to do some experimental work with auralized emacs and auralized debugger. I will meet Larry Scadder on June 13, 1995 at NSH to discuss the project in more detail. I have talked to Tricia from Audiology and Speech Sciences about joining in this experiment as a quarter time graduate student. She will help us in the design of experiments.

On Saturday April 8, 1995 Cary, Sriram, Nate, Katrina, and Leo participated in the Undergraduate day organized by the School of Science. This required them to write an abstract, prepare posters, and participate in a 4 hour poster session in the Stewart Center.

I have also had discussions with Doug Lee who is also a blind programmer working with a company in Illinois. He is wilkling to work with me on a part time basis to conduct experiments.

So, we are getting set for Fall 1995. I really hope NSF gives us funds from SGER (about \$50K) to carry on this research. It has been quite difficult getting students to spend a lot of time on this project. Whatever has happened has been due mostly to the voluntary work of students and their love for the Listen project. Chen worked without any money. Nate worked on hourly rate. Vivek, got half time support only during Spring though he has been working for about an year on this project. Katrina worked on an hourly basis. Cary, Leo, and Sriram worked for course credit. A lot of man hours have gone into the Listen project.

12. *August 19, 1996:* It has been an year since I updated this. Several things happened during 1995-6. (1) NSF did not award us the SGER grant. The reviewers said "NO". I did not agree with them but I have no choice. (2) Nate did a great job with the Sound Server interface. He left for Tektronix after graduating in Dec 1995. However, he is back to Purdue this semester to begin graduate work. (3) Howard Chen was the only person who worked all of 1995-96 on Listen. He has added some neat features to Listen so that one can generate some "real" music using Listen. His work is not yet ready for demo. I presume he will complete it soon and we will have Listen 4.0. (4) HP has a copy of Listen. Dave Boardman had asked for it. He has not done anything with it so far ( or at least I have not heard any news from him regarding Listen). (5) Two gentleman (Wilson and Lodha) are presenting a system named LISTEN at the NOV 1996 ICAD conference!!! This is not good but I guess this is how life works. (6) Cary Supalo worked with me throughout the year. He and Howard did some good experiments. I am yet to write about these.
13. *August 22, 2002:* Ed Gilmore and Bryan Strait have registered for CS 490 during this fall to work on Listen/Java. Here are the minutes of the first meeting we had:

LSL/Java group Meeting #1:

Present: Ed Gilmore and Aditya Mathur

Absent: Brayn Strait

Meeting duration: 10:30am-11:30am

Tasks for August 22-August 29.

- (a) Ed will begin work on generating an LSL/Java spec. The LSL/C spec will serve as the basis for LSL/Java.
- (b) Ed will simultaneously begin rewriting the LSL/C report to conform to LSL/Java.
- (c) Bryan (missed the meeting today) will select a Java parser, obtain its code and strip it of all code for generating the byte code. Bryan will thus generate a Java parser that parses Java programs but DOES NOT generate any byte code.
- (d) Both Bryan and Ed have been assigned space in the SERC lab. They will soon get accounts to work on machines in the lab. (Please obtain the key to the SERC lab from candy Walters.)

Next meeting: Thursday August 29 10:30am (this might need to change)

14. *August 28, 2002* Meeting #2 scheduled for August 27 did not take place due to confusion regarding the time. Here is a nice abstract of Bryan's progress.

TOOLS USED: I have decided to use javacc and Java Tree Builder, seeing that Java Tree Builder has been developed at Purdue under Prof. Palberg. Java Tree Builder constructs a set of classes using "visitors".

PROGRESS:

- Downloaded and installed jtb and javacc onto my lore account.
- Obtained a javacc formatted Java 1.4 grammar that is able to be used by jtb. Found here: [http://www.cobase.cs.ucla.edu/pub/javacc/java1\\_4c.jj](http://www.cobase.cs.ucla.edu/pub/javacc/java1_4c.jj)
- Used the Java 1.4 grammar in jtb to construct the classes.

NOTES:

The classes that jtb creates contain recursive "visitor" functions that are recursively called in the parser, building a syntax tree from the Java1.4 grammar. This will be very useful since the syntax tree can be traversed to obtain language productions. (Ex. function calls, "new" memory allocations, declarations, do-while/for loops, if-else. Everything)

I decided not to just take a full Java compiler and strip off the bytecode generation since there are a lot of compiler specifics to get into. I thought it would be a lot simpler and more efficient with building a syntax tree to make our "sound insertions" into the java code. This would

also somewhat gives us the ability to swap in and out grammars for different languages if we wanted to in the future if we wanted to. If anyone thinks otherwise, we can go about stripping off an already created java compiler.

#### THINGS TO DO:

- Get accounts setup and correct java versions and such on the SERC machines. Things seem to be working find on lore. Need get the jtb to work on those machines.
  - As soon as we get the LSL side done and we decide on our next step, I need to write some functionality to traverse the syntax tree looking for particular productions in which we will be adding our “sound” to.
15. *October 18/19, 2002* Dave Boardman and his wife Vicki visited us. Ed and Bryan gave Dave a demo of the initial prototype of Listen/Java. Dave came up with several ideas to bolster the Listen project. (1) Make it open source. (2) Issue a Boardman-Mathur challenge regarding the effectiveness of sounds in the analysis of program behavior. (3) Come up with several useful applications (network monitoring is one example.) Ed, Bryan, Dave, Vicki, Jyoti, and I had a fun Pizza dinner at Pizz King (the old “Garcia’s”).
  16. *October 25, 2002* Ed Gilmore has gone to the Undergraduate Symposium at Argonne National Lab in Chicago to present Bryan and his work on Listen/Java. Ed has a good set of slides and a demo. Ed wants to do his master’s and doctoral work on Listen.
  17. *October 29, 2002:* During today’s meeting some new ideas were floated that will improve Listen’s usability in testing/debugging. *Idea 1 (Bryan):* Add the ”Rhythmize” feature. For example, with this feature, a user can rhythmize a thread. Whenever a thread is assigned to a CPU and executes, a distinct rhythmic pattern is heard (different for each thread). *How does one implement this?* *Idea 2:* Generalize the LSL’s context specification feature. For example, in an LSL/Java specification one could declare: `context X package mypackage, Y package mypackage class myclass method mymethod`. Then you may use it as: `begin context notify all .....end context` This makes it easy to constrain the application of auralization specs to portions of a program.  
  
Ed had a good presentation at Argonne. His presentation slides are now posted on the Listen site. The demos that he gave will also be posted soon.
  18. *November 7, 2002:* Ed and Bryan have changed the name of the product to Jlisten. This appears to be an appropriate change and I like it. Bryan has also been able to work out a method to identify specific instance variables at run time. This will allow auralization of specific objects. He suggests using `System.identityHashCode(Object)` to keep track of variable instances within the Media Manager.
  19. *September 10, 2003* Ed and Bryan work did not make much progress as both left school after graduation. Chris Kanich did some work at Purdue during spring 2003 on reviving the JListen

project. Not much progress so far. Starting the fall of 2003, three groups of 5 students each have resumed work on the Jlisten project. In addition Jayaram is doing a 3-credit project with me while I spend my sabbatical at Purdue. The first key task is to develop LSL/Java. Jayaram has made some progress towards this. I expect a report on LSL/Java soon. One of the three groups consists of graduate students. I expect a respectable JListen 1.0 to be ready by December 2003. As suggested by Dave, I want to then place this project in the public domain so others could begin contributing.