NAP-Time Who Did What

What Benny has done:

1. Compiled kernel to support detection of USB flash drive of any file system.
2. Wrote a program to auto mount USB flash drive when it is being inserted into USB slave port and auto unmount when it is being removed.
3. Researched on installing NFS server on target, but was not successful when configuring TCP wrapper, did find a way to communicate between two boards through boa web server.
4. Modified madplay source code such that madplay takes its input from a named pipe instead of keyboard.
5. Programmed button interface such that pressing button can control madplay operation to pause, forward, back, resume, increase/reduce volume, and quit.
6. Wrote a how-to guide 'How to control madplay through button press'

Benny did crucial work in making Madplay more amenable to our purposes. He also created code for auto-mounting USB storage. He wrote a button interface as well.

What Thomas has done:

Thomas wrote a client for the first design concept, a client-server system architecture, which was subsequently abandoned. Following this, Thomas focused effort on the GUI for the music player. The challenge in this was making use of Benny's enhanced madplay which allowed use of that player through the GUI. After getting this player written and working successfully on both local and remote nodes, an additional goal which had been suggested by Cal was that SHA1SUMs be calculated for each mp3 file so as to gain the advantage of uniquely identifying all music on all connected nodes, which was envisioned to allow us to avoid redundancy that might occur where two mp3 files with different file names are in fact the same music. This too was accomplished, but introduced an intermittent, mysterious segmentation error that Thomas was unable to resolve successfully. Also, the on-the-fly calculation of SHA1SUM for numerous mp3 files resulted in long delays before remote nodes became available. With time running short, it was decided that an expedient choice would be to roll back the player to the previous working version.

Testing for Thomas at home was done using a board and a notebook PC separate from the Host PC and configured with a directory structure similar to that of the ARM boards. Although the roll-back was successful and remote and local songs became available in the player GUI, on the occasion of the final configuration meeting, the remote nodes did not display music. A debugging did not achieve a resolution, and Benny undertook to relieve Thomas of this final debugging effort due to workload considerations, (Thomas is taking 4 400 level CS courses and was given a surprise 1500 character Chinese essay requirement with 4 days notice for completion of his degree). Button functionality was also accomplished to the extent of navigation through the lists of mp3s.

Thomas assisted contributed to the project proposal, assisted in documenting meeting notes/whiteboards, scheduled one of the meeting reservations, and produced the PowerPoint for the in-class demonstration. Thomas stupidly suggested trying the new and unpredictable GIT at the beginning, after which Cal showed everyone the wisdom of using the tried and try SVN via Google Code.

Cal Woodruff:

- Did most project management tasks – created svn repository, organized most meetings, developed test system for integration testing during meetings.
- Developed overall system architecture and coding concept.
- Developed multicast core components, web tools, RTP integration, launcher, memo and intercom gui apps.
- Debugged player application.
- Wrote “Will it Build – Part 1” how to guide.
- Co-wrote project report.

From Subversion: who authored what code.
Reference: http://code.google.com/p/cmpt433-nap-time/source/list
http://code.google.com/p/cmpt433-nap-time/source/browse

Component	Description	Developers
madplay-0.15.2b	madplay modified to use fifo for input	Benny
usbAutoMount	auto detect usb keys	Benny
buttonControlmadplay	control madplay via fifo	Benny
buzzer, leds	basic tools for changing leds sounding buzzer for scripts	Cal
napscan	multicast node detection	Cal
naprtp	modification of oRTP test programs to work with Alsa/madplay	Cal
p2p	tools for website	Cal
p2p/shared	website	Cal
cgiparser	tool used for website cgi scripts	Cal
musicscan	grab list of shared music from other nodes in multicast group	Tom
gui:		
gui/launcher	main program to start other gui programs	Cal
gui/player	music player	Tom
gui/memos	record and send memos	Cal
gui/intercom	talk directly with other nodes	Cal
gui/scripts	support scripts used by gui	Cal