Resurrecting the Dynabook Vision: Squeak

- What Steve Jobs Missed:
  The Dynabook Vision
- What happened over the last 20 years
- Why Squeak came to be
- What it can do…Dynabooks
- What GaTech Squeakers do...Collaborative Dynabooks

What Steve Jobs Saw

- Object-oriented programming
  - Class-based, methods, browsers for code exploration and entry
- A fast bytecode compiler
  - Smalltalk compiles to bytecode (like Java)
  - Bytecode interpreted by a ***small*** *Virtual Machine* (unlike Java :-)
    - PowerMac 308K, Win32 328K, Linux-Intel 276K, SGI Irix 609K, Solaris 816K
  - "Image" file contains all of object code, including development environment
    - 0.5M -> 5.5M
  - Runs comfortably in 4M, everything in 15M

What Steve Jobs Saw

- Development environment written in Smalltalk
  - Powerful tools for exploring system
  - Easily change code on the fly
  - Ease of porting
    - It’s all Smalltalk, all the way down

What Steve Jobs Missed

- Dynabooks
  - Alan Kay, Adele Goldberg, Dan Ingalls, Ted Kaehler
  - Goal: Creating the next "book" and the next "printing press"
  - Personal Dynamic Media, for learning and thinking
    - "A Personal Computer for Children of All Ages"
  - It's about *media*, Steve!
    - And individual control of that media

What happens next...

- Steve Jobs steals ideas from Xerox, and Bill Gates steals them from Apple
- Meanwhile…
  - Xerox sends Smalltalk-80 Release 1 to Tektronix, Sun, IBM, Apple
    - Gets ported all over the place
  - Alan Kay goes to Atari, then Apple
  - Adele Goldberg heads up Parcplace
    - Smalltalk-80 => ObjectWorks => VisualWorks
  - Digitalk Smalltalk/V, and others

Fast forward to 1995

- Alan, Dan, and Ted are at Apple
  - Today’s implementations get the Desktop User Interface *wrong!*
  - Current multimedia composition tools miss the point
- Smalltalk-80 was better than Director is today.
- “Control of one’s own software destiny"

Add some members to the team

- John Maloney: Musician, inventor of Morphic UI for Self
- Scott Wallace, inventor of MacApp: End-user programming

Enter...Squeak!

- Take the original Smalltalk-80 Release 1, and get it running again
- "We never want to program in C again."
  - Rewrite the virtual machine in Smalltalk
  - Write a Smalltalk-to-C translator
  - Generate a new virtual machine, with the source code in Smalltalk
    - Now it truly is "Smalltalk, all the way down"
- Release the whole thing to the Internet
  - With an Apple License that requires enhancements to be shared
  - Within a month: Ported to UNIXes, Win95/NT
  - Later: PDA's, set-top boxes, bare chips, SqueakOS
    - (Can run in a couple meg.)

Enter...Disney!

- Whole Squeak team leaves Apple for Disney Imagineering
  - It’s about *media*, Steve!
- Port Morphic to Squeak
- First Squeak game released at Disneyland last year
- First Squeak game released on Disney Blast! in next six months

Squeak Today

- Even better programming tools
  - Including end-user programming
- Media support
  - Going wild with text
  - Audio
  - MIDI
  - 3-D graphics
  - Flash
- Tools: Email, web browser, web server, solitaire and Tetris (*of course*)

Where the GaTech Squeakers Enter the Picture

- We want Dynabooks, but we want them *collaborative*
  - Ease of composing, alone and together
  - Ease of sharing, commenting, critiquing

Squeakers Projects: Shared Composition of Media

- Webservers:
  - PWS: Pluggable Webserver
  - Comanche: Bolot Kerimbaev
- CoWeb/Swiki: Jochen “Je77” Rick
  - Using CoWebs for Learning: Colleen Kehoe
- ICsQueak: Dean “Dino” Mao

Squeakers Projects: Ease of Constructing Media

- Music Analysis/Synthesis for Math Learning: Rodney Walker
- Music Editing Tools: Eddie Cottongim
- MAT (Multimedia Authoring Tool) and Jukebox: Aibek Musaev
- Tools for novice programmers: Jennifer Brown

Squeakers Projects: New Kinds of Media

- MuSwiki: Lex Spoon
- Prolog + Jini + Squeak: Bolot Kerimbaev

What Others are Doing

- Facial animation
- Speech synthesis and Text-To-Speech
- MathMorphs, at UBA, Argentina
  - “Interactive Mathematica”
- Regular expressions, YACC-like parsers
- (Recreating the Alto)

Where Squeak is going next...

- Video and music editing/synchronizing in Squeak
- “Replace HTML”
  - Drag-and-drop multimedia pages
  - Squeak in the Web
- Books: Textbooks, edited volumes

What Squeak is NOT for...

- Use Java instead for:
  - Database connectivity
  - E-commerce or other high-traffic sites
  - Platform-specific look-and-feel

What Squeak is GREAT for...

- New media applications
  - More flexible, faster, and more cross-platform than Director
- Inventing new media
  - Web-based, audio, video, whatever
- CS learning and research
  - *All* of the source is there, *all* in one language, and it runs on *any* platform
- Embedded and real-time projects
  - Interval just did a real-time OS in Squeak-OOPSLA99