

Why give a talk about Smalltalk at a CocoaHeads meeting?

Because Smalltalk has had a very long and influential relationship with Apple. Many of the people involved with the development of Smalltalk were also involved with Apple and NeXT development. First a brief history.



Smalltalk was invented in the 1970s by Alan Kay and implemented by Dan Ingalls while they were at Xerox Palo Alto Research Center aka Xerox PARC. In 1979, Apple asked (paid) them to give a demo of their cool technology to someone we all know.



Steve was flush with the success of the Apple][, had heard about this really cool interactive environment Xerox had invented and paid them for access to their technology.

One of the best parts of the demo was when Steve Jobs said he didn't like the blt-style scrolling we were using and asked if we could do it in a smooth continuous style. In less than a minute Dan found the methods involved, made the (relatively major) changes and scrolling was now continuous! This shocked the visitors,

From "Early History of Smalltalk by Alan Kay Xerox Smalltalk Demo 1979

Just words if you don't know Smalltalk.

http://propella.sakura.ne.jp/earlyHistoryST/EarlyHistoryST.html#p38

By now it was already 1979, and we found ourselves doing one of our many demos, but this time for a very interested audience: Steve Jobs, Jeff Raskin, and other technical people from Apple. They had started a project called Lisa but weren't quite sure what it should be like, until Jeff said to Steve, "You should really come over to PARC and see what they are doing." Thus, more than eight years after overlapping windows had been invented and more than six years after the ALTO started running, the people who could really do something about the ideas, finally to to see them. The machine used was the Dorado, a very fast "big brother" of the ALTO, whose Smalltalk microcode had been largely written by Bruce Horn, one of our original "Smalltalk kids" who was still only a teen-ager. Larry Tesler gave the main part of the demo with Dan sitting in the copilot's chair and Adele and I watched from the rear. One of the best parts of the demo was when Steve Jobs said he didn't like the blt-style scrolling we were using and asked if we could do it in a smooth continuous style. In less than a minute Dan found the methods involved, made the (relatively major) changes and scrolling was now continuous! This shocked the visitors, especially the programmers among them, as they had never seen a really powerful incremental system



To the left is roughly what Steve would have seen at the demo. To the right is the Apple][startup screen from a similar time period.

Smalltalk influenced the Mac, Windows Java, Xcode, IDEs, Ruby, Objective-C, NeXT



Influenced the Mac GUI but not the mac programming language. Is Smalltalk more than a language or a gui?

```
exampleWithNumber: x
"comment"
| y |
true & false not & (nil isNil) ifFalse: [self halt].
y := self size + super size.
#($a #a "a" 1 1.0) do: [ :each |
        Transcript show: (each class name);
            show: ' '].
^x < y
Is it a language?</pre>
```

Next adopted Objective-C with message passing.

Used the Builder IDE

Was a Dynamic system like Smalltalk but there is no Smalltalk style browser except with F-Script.

A method that illustrates every part of Smalltalk method syntax except primitives. It has unary, binary, and keyboard messages, declares arguments and temporaries, accesses a global variable (but not an instance variable), uses literals (array, character, symbol, string, integer, float), uses the pseudo variables true, false, nil, self, and super, and has sequence, assignment, return and cascade. It has both zero argument and one argument blocks.

Influenced Objective-C syntax and message passing. Java, and many other languages.

Smalltalk is a Live Everything an object Everything integrated	ing Environm	ent
Graphical Windows & Menus	Environme Automatic Memory Mgmt	ent Live Object Debugging
Hardware Independent	mple Syntax	Object/Knowledge Persistence
The purpose of the Smalltalk project is to provide computer support for the creative spirit in everyone. <u>DanIngalls</u>		

Smalltalk is the "Apple" of programming languages.

Integrated from top to bottom.

Problem is, Smalltalk is in the Apple '90s still waiting for a Steve to come along and make it popular.



Everything is an object.

"Byte compiled objects" - my way of saying incrementally compiled byte code.



Message passing is very robust. It does not care about types just whether or not an object responds to a message. Response is voluntary and up to the object responding which means the message can't crash the object. You can pass a message asking a collection to increment. The collection knows better than to increment past the end of the collection. So no buffer over run or under run. There are no array by pointer accesses so no getting the wrong object or garbage because the pointer offset was wrong due to assuming the wrong type size or passing the end. Types don't matter only protocols. Collections have a standard set of messages to which they will respond. This is the Collection protocol. In Smalltalk, the class hierarchy is a common way of defining a protocol. The abstract superclass methods are the protocol which all subclasses implement in their own special way.



What are the common developer pain points?

```
The write -> compile -> debug -> write loop for every change?
```

Inability to "live change"?

Crashes and debugging due to "types"?

....?

Twilio, Seaside debugging demo.



Like a block of clay, you mold Smalltalk bit by bit, into the object you desire.



1970 - Alan Kay development of Smalltalk at Xerox Palo Alto Research Center (PARC)
1979 - Xerox PARC demos Smalltalk to Steve Jobs and others
1983 - Objective-C, object oriented C with message passing based on Smalltalk
1984 - Apple introduces the Macintosh Computer with a GUI from Smalltalk (Do It -> Ok)
1988 - NeXT acquires Objective-C
1988 - NeXT creates Interface Builder (& IDE Project Builder?) (Xcode ancestors)
1996 - Apple acquires NeXT (or NeXT takes over Apple :-)
2001 - Apple introduces OS X (Smalltalk -> NeXTStep -> OS X)
2011 - Apple delivers Automatic Reference Counting (ARC)
2011 - OS X Lion includes document and desktop persistence
2011 - Bertrand Serlet leaves Apple, worked at Xerox PARC, NeXT, Apple OS X
2014 - Apple introduces Swift with interactive workspace style "Playgrounds"

2011 - After 32 years, Apple has a computer which fully implements most of the Smalltalk experience for the average user.

Note, Apple could of always had "Playgrounds" in Objective-C with a utility like F-Script which allows sending Objective-C runtime messages and introspection of the runtime.



Squeak is an open source direct descendant of Smalltalk. Written by Dan Ingalls the creator of the original Smalltalk implementation. Squeak uses a GPL and so is not commercial use friendly.

Pharo is the latest open source Smalltalk good for commercial use. It was ported and re-licensed from the Squeak GNU license to a commercial friendly MIT license.

Cincom is a commercial business and owner of the original commercial Smalltalk from Xerox PARC and Digitalk. They sell VisualWorks.

Seaside is a web application server framework written for the Smalltalk platform. Includes block continuations and closures. Meaning it handles session information rather than the programmer.

Amber is a Javascript implementation of Smalltalk which can be used to implement client or server side javascript interfaces and applications.