The Development of Smalltalk
Topic Paper #7

Alex Laird
CS-3210-01
2/4/09
Survey of Programming Languages
Spring 2009
Computer Science, (319) 360-8771
alexdlaird@cedarville.edu

Grading Rubric		Max	Earn
	On Time/Format	1	
	Correct	5	
	Clear	2	
	Concise	2	
	Total	10 pts	

Peer Reviewer

ABSTRACT
This paper describes the development of the Smalltalk programming language.

Keywords
Development, Smalltalk, Programming, Language

1. INTRODUCTION
Smalltalk, yet another programming language originally developed for educational purposes and now having a much broader horizon, first appeared publically in the computing industry in 1980 as a dynamically-typed object-oriented language based largely on message-passing in Simula and Lisp.

2. EARLY HISTORY
Development for Smalltalk started in 1969, but the language didn't make a public appearance until 1980. It was developed for education purposes and namely for the Xerox families personal work stations [3]. It was the child of a research group led by Alan Kay, and the first version, Smalltalk-71, was based on a bet that a message-passing (communication between objects introduced in Simula) could be implemented in a page of code and was written in just a few days.

The idea that large problems could be broken down into smaller working pieces and that models of specific things could be represented by objects made programming a much greater ease [1].

3. OBJECT-ORIENTATION
While Simula receives the credit for being the first programming language to introduce the concept of objects and object-orientation, Smalltalk was the first language to be called "object-oriented," and it was the first purely object-oriented language. Unlike the later developed object-oriented languages such as C++ or Java, there is no distinction in Smalltalk between primitive data types and objects [2].

4. TYPING AND COMPILING
The only typing available in Smalltalk is dynamic. This means it is not imperative to type variables when you declare them, they will be typed dynamically on runtime. More specifically, they will be typed to whatever value is assigned to them at runtime. Dynamically typed languages are easier on the compiler because it has to make fewer passes and the brunt of checking is done on the syntax of the code.

Compilation of Smalltalk is just-in-time compilation, also known as dynamic translation. It means that upon compilation, Smalltalk code is translated into byte code that is interpreted upon usage and at that point the interpreter converts the code to machine language and the code is executed. Dynamic translations work very well with dynamic typing.

5. IMPLEMENTATIONS
Smalltalk has been implemented in numerous ways and has been the influence of many future languages such as Java, Python, Object-C, and Ruby, just to name a few.

Athena is a Smalltalk scripting engine for Java. Little Smalltalk was the first interpreter of the programming language to be used outside of the Xerox PARC. Smalltalk has even gone portable, available on Palm Pilots with Pocket Smalltalk. A static typing environment of Smalltalk has been implemented on Windows with Strongtalk. And there is even an open source version of Smalltalk called Squeak.

6. CONCLUSIONS
The above list of implementations only begins to touch on the numerous developments off of the Smalltalk language since its release in 1980. The language itself is still used today in programming quite few as it is one of the more powerful purely object-oriented languages available. Without Smalltalk's introduction of pure object orientation and its influence on future portable and interpreted languages such as Java and Ruby, programming wouldn't be what it is today.

7. REFERENCES
[1] Sebesta, Robert W., 2008. Concepts of Programming Languages. Addison-Wesley, Boston. ISBN: 978-0-321-49362-0

[2] Kay, A. C. 1993. The early history of Smalltalk. In the Second ACM SIGPLAN Conference on History of Programming Languages (Cambridge, Massachusetts, United States, April 20 - 23, 1993). HOPL-II. ACM, New York, NY, 69-95. DOI= http://doi.acm.org/10.1145/154766.155364

[3] Bennett, J. K. 1987. The design and implementation of distributed Smalltalk. In Conference Proceedings on Object-Oriented Programming Systems, Languages and Applications (Orlando, Florida, United States, October 04 - 08, 1987). N. Meyrowitz, Ed. OOPSLA '87. ACM, New York, NY, 318-330. DOI= http://doi.acm.org/10.1145/38765.38836