

The Spoken Web - Software Development and Programming through Voice

Arun Kumar
IBM Research - India
4, Block C, Vasant Kunj,
Institutional Area,
New Delhi - 110070, INDIA.
kkarun@in.ibm.com

Sheetal K. Agarwal
IBM Research - India
Embassy Golf Links,
Block D, Domlur Ring Road,
Bangalore 560071, INDIA.
sheetaga@in.ibm.com

Priyanka Manwani
IBM Software Group
4, Block C, Vasant Kunj,
Institutional Area,
New Delhi - 110070, INDIA.
pmanwani@in.ibm.com

ABSTRACT

It has been a constant aim of computer scientists, programming language designers and practitioners to raise the level of programming abstractions and bring them as close to the user's natural context as possible. The efforts started right from our transition from machine code programming to assembly language programming, from there to high level procedural languages, followed by object oriented programming. Nowadays, service oriented software development and composition are the norm.

There have also been notable efforts such as Alice system from CMU to simplify the programming experience through the use of 3D virtual worlds. The holy grail has been to enable non-technical users such as kids or non-technical people to be able to understand and pick up programming and software development easily. We present a novel approach to software development that lets people use their voice to program or create new software through composition. We demonstrate some basic programming tasks achieved by simply talking to a system over an ordinary phone. Such programs constructed by talking can be created in user's local language and do not require IT literacy or even literacy as a prerequisite. We believe this approach will have a deep impact on software development, especially development of web based software in the very near future.

Categories and Subject Descriptors: H.5.2[Information Interfaces and Presentation]:User Interfaces–*Voice I/O, User-centered design*; D.1.0[Programming Techniques]:General

General Terms: Design, Human Factors, Languages

Keywords: VoiceSites, Spoken Web, Programming by Voice

1. INTRODUCTION

In the early days of computing, programming was quite a tedious task. For instance, the first general purpose computer called Electronic Numerical Integrator And Computer (ENIAC) required about two weeks to load a program. The process involved manipulating its switches and cables.

Right from those days it has been a constant endeavor of computer scientists to make programming simpler and provide abstractions that are as close to the user's natural context as possible. Assembly language programming [1] introduced human readable method of writing programs that

used symbolic codes rather than programming with binary bits. It was followed by several high level procedural languages such as FORTRAN, COBOL etc. Procedural programming works by specifying the exact steps required to be taken to execute the logic of the program.

Object oriented programming further simplified the task of the programmer by introducing the notion of objects, meant to be representations of real world entities, and were realized in Simula programming language and the Smalltalk environment. Service oriented computing [12] is a recent phenomena which tends to move away from artifacts in the Information Technology (IT) domain, represented by objects to entities in the business domain, realized in the form of services.

The efforts outlined above simplify the task of a trained programmer and make him more productive by empowering the syntax of the language. An orthogonal track of research is aimed towards producing programming languages and systems that make programming accessible to non-experts such as kids or people having no IT background. Such programming languages or environments focus more on the interface exposed to the novice developer and include systems such as Alice [5], Scratch [14], and SIMPLE¹, etc.

However, there is a large segment of human population that does not have access to information systems enabled by Internet and the Web. They either cannot afford computers or devices required to get access to information systems or are alienated due to low literacy levels. In fact, Internet access is available to just 22% [15] of human population.

Addressing the needs of such untouched population, in information technology domain and more specifically, in the context of online information and services, we have proposed World Wide Telecom Web (WWTW) [10] (also known as Spoken Web or Telecom Web) that is emerging as an alternate web for the underprivileged. Spoken Web is envisioned as being complementary to the existing Web and is navigable entirely through a voice based interface using an ordinary telephone. A node in the Spoken Web graph is a voice application called a *VoiceSite* [9] and is accessible over a simple phone call. VoiceSites are created by end users through a voice based interface [7], hyperlinked through *Hyperspeech Transfer Protocol* (HSTP) [2] and can be browsed through a Telecom Web Browser [3]. Such a web of applications, driven by spoken commands, opens up tremendous opportunities for billions of people in developing countries to whom the existing Web is inaccessible due to illiteracy,

Copyright is held by the author/owner(s).
WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

¹<http://www.simplecodeworks.com>

language barriers, infrastructural problems, low disposable income and lack of content or services that are locally relevant.

Spoken Web presents unique challenges in terms of application creation. First, application developers are assumed to be illiterate or semi-literate. Second, most are familiar with only their local language. Third, currently Spoken Web being entirely driven by voice there are natural constraints in terms of the interaction design that can be presented to the developer.

The Spoken Web Application Framework (SWAF) [7], addresses these requirements and provides an environment for creation of content and services in Spoken Web. In this paper, we present novel aspects of software development using SWAF that lets people use their voice to program or create new software through composition. We demonstrate some basic programming tasks achieved via simply talking to a system over an ordinary phone. Such programs constructed by talking can be created in user's local language and do not require IT literacy or even literacy as a prerequisite.

2. PROGRAMMING THROUGH VOICE

Typical steps in a program's lifecycle include the following:

1) Program creation through an editor, 2) Program compilation into an executable form, and 3) Program execution.

Further, creating a program makes use the following essential elements found in a programming language: 1) Variables - entities to store data, 2) Control constructs such as sequencing, iteration, and branching and 3) Procedures and functions. In this section, we demonstrate how SWAF enables programming through voice in Spoken Web.

Programs in Spoken Web

Programs created using SWAF are essentially voice applications called VoiceSites [9]. SWAF presents a voice based interface (in addition to a programmatic interface) for creating such applications. In the initial trials, it was observed that presenting an open ended interface left the unskilled user clueless about how to interact with the system. Therefore, SWAF enables program creation in a constrained fashion driven by pre-defined templates.

The program editor in SWAF is realized in the form of a meta-voice application that is generated from and driven by one of the pre-defined templates. The program editing activity essentially consists of selecting those components of the template that are needed in the program and then filling in the properties to define the behavior of those components.

Use-case Scenario

To illustrate the program creation process and to explain further, we present the example of a micro-business VoiceSite template. It is used by SWAF to let micro-businessmen such as plumbers, electricians, carpenters etc. create their own business VoiceSites. Let us assume that a micro-business VoiceSite template offers the following features: 1) Profile including information such as name, profession, service charges etc., 2) Personalized Messages for individual customers which they get to hear when they call the VoiceSite, 3) Links to other sites and 4) Appointments with an ability to receive an SMS when a customer fixes an appointment.

Program Lifecycle in SWAF

To create a micro-business VoiceSite with some or all of the above mentioned features, the creator calls the SWAF application and requests the micro-business template. Based

upon the template specification, SWAF generates and renders the voice based editing interface for the program creator. Essentially, SWAF provides options to the program creator to select desired components and guides him through the process of configuring the properties of the chosen components. At the end of the interaction, a customized *template instance* is created that consists of the specification of the new program (i.e. VoiceSite) along with required functional (e.g. SMS functionality) and UI components (e.g. appropriate language prompts). Template Instance is essentially the *compiled* program that can be executed.

Having created the compiled version, SWAF *loads* the program to make it ready for execution by deploying it on the application server and assigns a phone number to this application. The creator and his customers can now access this business VoiceSite using this new number.

Execution of the program is triggered by arrival of a phone call at the associated phone number and results in generation and presentation of the voice interface as per the program specification. The program exits executing once the caller hangs up or the call is terminated abnormally.

Programming Elements in SWAF

Next, we briefly illustrate the use of programming constructs in SWAF based program creation through the example template introduced above. The micro-business VoiceSite template offers the Personalized Messages feature in such a way that the creator can enable it for as many of his callers as he desires. This requires the creator to *iterate* through the Personalized Messages interface.

Branching construct is incorporated in SWAF either in the form of user controlled branching or system controlled branching. User controlled branching is realized in the form of menu-based interface presented to the user where user input decides the part of the program that executes. System controlled branching is typically conditional branching and depends on the values of one or more control variables (either system set or set based upon user input at different points in the program execution). Configuring a Personalized Message feature essentially inserts a system controlled branch, where the message is played only if there is a message configured by the creator for the caller who has called.

Procedure calls in SWAF are supported for local as well as remote invocations. The activity requiring sending of an SMS on booking of an appointment requires invoking the remote web service interface of an SMS gateway. It effectively constitutes a remote procedure call. Local procedure calls are made among various functional components that get bundled together to form a template instance.

3. RELATED WORK

Beyond the move from machine language to high level procedural and object oriented languages of today, several recent and ongoing research efforts have contributed towards bringing programming closer to the user. A rich survey of such efforts meant to increase the reach of programming is presented in [6].

Recent work done in services computing [8] and workflow domains (BPMN, BPEL) [11] are aimed to bring programming closer to the business user. They enrich the programming abstraction thus giving more power to the programmer while requiring lesser syntax. Moving an abstraction level up from objects as the basic unit of programming, these efforts allow business services to be treated as a first class language level construct.

Most notable among the efforts that design special programming interfaces for novices, is the Alice [5] programming environment. It uses a 'drag and drop' approach to let users select animated 3D objects and include them in their program representing a story. Students are able to learn basic programming through animated movie and simple video game creation. Scratch [14] is another programming language intended to let novices create their own interactive stories, animations, games, music, and art through the use of animated characters.

SiMPLE Modular Programming Language & Environment (SiMPLE) is a programming development system for non-professionals that provides its programming capabilities through a list of 23 keywords (4 keywords in an introductory version). Using these one could create various kinds of animations and interactive games. It is different from Alice as the developers need to write code even though it is much simpler whereas Alice lets the programmer select objects and configure their properties. Phrogram² also takes code-centric approach to programming and, similar to SiMPLE, keeps language is concise and specific.

There are several web application frameworks whose technical goals are similar to those of SWAF [7], i.e. enabling application creation through components. Reusable Dialog Components (RDC) [4] is a framework for developing voice applications that requires integration of UI components and business logic at design time. Hamlets [13] is an open source system for generating dynamic web-pages from XHTML templates. Java Server Faces (JSF)³ and ASP.Net adopt a component based approach to web application development. Apache Velocity⁴ is a Template Engine for Java. It provides a simple yet powerful template language to reference objects defined in Java code. However, all these are meant for making developers more productive and do not enable programming for novices. Also, none of them provides a voice based interface over phone call.

4. CONCLUSION

In this paper, we have introduced a novel mechanism of creating programs via voice interface. This mechanism is intended to enable non-professionals and non-literate people to be able to create programs by simply talking to a system over an ordinary phone rather than by writing code directly. The proposed method currently supports the typical steps covering the lifecycle of a program as well as basic programming constructs such as loops, branches and procedures. In future, we intend to expand the capability of the environment by adding support for features such as program revision, versioning and previewing etc. among others.

5. REFERENCES

- [1] P. Abel. *IBM PC Assembly Language and Programming*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.
- [2] S. Agarwal, D. Chakraborty, A. Kumar, A. A. Nanavati, and N. Rajput. HSTP: Hyperspeech Transfer Protocol. In *ACM Hypertext 2007*, UK, September 2007.
- [3] S. K. Agarwal, A. Kumar, A. A. Nanavati, and N. Rajput. The World Wide Telecom Web Browser. In *WWW '08: Poster Proceedings of the 17th International World Wide Web Conference*, Beijing, China, 2008.
- [4] R. P. Akolkar, T. Faruque, J. Huerta, P. Kankar, N. Rajput, T. Raman, R. U. Udupa, and A. Verma. Reusable Dialog Component Framework for Rapid Voice Application Development. In *SIGSOFT Component Based Software Engineering*, Missouri, USA, May 2005.
- [5] M. Conway, S. Audia, T. Burnette, D. Cosgrove, and K. Christiansen. Alice: lessons learned from building a 3d system for novices. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, 2000.
- [6] C. Kelleher and R. Pausch. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Comput. Surv.*, 37(2):83–137, 2005.
- [7] A. Kumar, S. K. Agarwal, and P. Manwani. The Spoken Web Application Framework User Generated Content and Service Creation through low-end mobiles. In *7th International Cross-Disciplinary Conference on Web Accessibility (W4A). Collocated with World Wide Web Conference*, 2010.
- [8] A. Kumar, A. Neogi, S. Pragallapati, and D. J. Ram. Raising Programming Abstraction from Objects to Services. In *IEEE International Conference on Web Services (ICWS)*, pages 864–872, 2007.
- [9] A. Kumar, N. Rajput, D. Chakraborty, S. Agarwal, and A. A. Nanavati. Voiserv: Creation and delivery of converged services through voice for emerging economies. In *WoWMoM'07 Proceedings of the 2007 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Finland, June 2007.
- [10] A. Kumar, N. Rajput, D. Chakraborty, S. Agarwal, and A. A. Nanavati. WWTW: A World Wide Telecom Web for Developing Regions. In *ACM SIGCOMM Workshop on Networked Systems For Developing Regions*, Aug 2007.
- [11] M. Z. Muehlen and J. Recker. How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In *CAiSE '08: Proceedings of the 20th international conference on Advanced Information Systems Engineering*, 2008.
- [12] M. P. Papazoglou. Service-Oriented Computing: Concepts, Characteristics and Directions. In *Proceedings of WISE*, Dec 2003.
- [13] R. Pawlitzek. Introducing Hamlets. <http://www-128.ibm.com/developerworks/web/library/wahamlets/>, Mar 2005.
- [14] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai. Scratch programming for all. *Commun. ACM*, 52(11), Nov 2009.
- [15] I. W. Stats. World Internet Users and Population Stats. <http://www.internetworldstats.com/stats.htm>, June 2008.

²<http://phrogram.com/>

³<http://java.sun.com/javaee/jaserverfaces/overview.html>

⁴<http://velocity.apache.org/>