



Creating Personal Mobile Widgets without Programming

Geetha Manjunath, Santhi Guntupalli, Thara S, Vinay Kumar K, Hitesh Bosamiya, Ragu Raman G

HP Laboratories
HPL-2009-94

Keyword(s):

Programming-By-Example, Widgets, Mobile, Interaction, HCI

Abstract:

Our goal is to radically simplify the web so that end-users can perform their personal tasks with just a single click. For this, we define a concept called Tasklet to represent a task-based personal interaction pattern and propose a platform for automatically creating, sharing & executing them. Tasklets can be deployed on the client, cloud or on telecom provider network - enabling intuitive web interaction through widgets, thin mobile browsers as well as from mobile phones via SMS and Voice. Our key innovation is a tool and platform that enables end-user to simplify personally valuable task on the web without any knowledge of programming.

External Posting Date: May 6, 2009 [Fulltext]

Approved for External Publication

Internal Posting Date: May 6, 2009 [Fulltext]



Published and presented at 18th International World Wide Web Conference, Madrid, Spain, April 20-24, 2009.

© Copyright The 18th International World Wide Web Conference, 2009.

Creating Personal Mobile Widgets without Programming

Geetha Manjunath
geetha.manjunath@hp.com
Santhi Guntupalli
santhi.guntupalli@hp.com

S Thara
thara.s@hp.com
Vinay Kumar K
vinaykumar@hp.com

Hitesh Bosamiya
hitesh.bosamiya@hp.com
Ragu Raman G
ragu@hp.com

Hewlett Packard Research Labs,
No 24, Adugod, Bangalore, India

ABSTRACT

Our goal is to radically simplify the web so that end-users can perform their personal tasks with just a single click. For this, we define a concept called *Tasklet* to represent a task-based personal interaction pattern and propose a platform for automatically creating, sharing & executing them. Tasklets can be deployed on the client, cloud or on telecom provider network – enabling intuitive web interaction through widgets, thin mobile browsers as well as from mobile phones via SMS and Voice. Our key innovation is a tool and platform that enables end-user to simplify personally valuable task on the web without any knowledge of programming.

Categories and Subject Descriptors

D.1.2 [Automatic Programming] H.5.3 [Web-based Interaction] H.5.4 [Hypertext/Hypermedia]

General Terms

Human Factors, Languages

Keywords

Programming-By-Example, Widgets, Mobile, Interaction, HCI

1. INTRODUCTION

The web has grown in multiple ways - content, complexity of navigation, number of web sites, web services and service providers. Current web interfaces require the user to remember websites, perform complex navigations, fill multiple forms and essentially figure out a sequence of actions that are needed to accomplish a simple task – often across multiple sites. Even when she succeeds in doing it once, the next time a similar goal arises, she has to recall and manually repeat the same exercise! This problem is even more acute in mobile context, when the device has interface challenges and user is very focused on getting a task done. We need to simplify this web interaction. By radically simplifying the usage of the web and providing familiar means of interacting with it, we believe more number of people will start deriving value from the web. Further, reduced interaction complexity of web can be a major productivity improvement for an IT-savvy user. Finally, 1-click access to personally valuable web tasks from mobile devices can be a compelling user experience of the mobile web.

An appliance or widget model of customized packaging of complex actions is a well-known paradigm of simplification. Since different users perform different personal tasks on the web, and possibly in diverse ways, use of pre-packaged widgets does not scale. Today, if a user wants to create a new widget that does



Figure 1: User Created Personal Widgets

what he wants, it requires development of a new web application, which in turn mandates programming knowledge of one or more of Java, JSP, Servlets, AJAX, JavaScript, Flex or some similar technology. Our goal is to break this limitation and enable the end-user to create simple ‘single-click’ widgets for his personal task without programming. We propose to demonstrate a tool and platform for creating, sharing and executing such personal widgets. The target user for our tool is an end-consumer with just simple web-browsing knowledge. Advanced user can, of course customize the behavior of the auto-generated widget.

2. OUR SOLUTION

Our key approach to solve the above problem is to package a user’s personally valuable web interaction across multiple web sites as a first class web-entity and provide an execution platform for robust execution of the same. For this, we define a concept called Tasklet to capture a user’s preferred way of accomplishing a task – compressing the sequence of web actions needed to perform a specific task in a user-specified way. These Tasklets can be user-created, virally shared, customized and composed with other Tasklets and web services.

In order to enable an end user to program a Tasklet, we use the technique of Programming-By-Demonstration. All the user needs to do is to perform the task (web interaction) once using her web browser by giving some sample inputs on all the relevant web sites. We record and analyze these browser actions to auto-generate custom personal widgets which compress all the required actions into a single-click interface. Further, using our cloud-hosted services, these widgets can be executed from a mobile device (through a thin web browser, SMS or even voice), despite certain device limitations.

For instance, if a user frequently goes on 2-day trips from Bangalore to Delhi by Jet Airways, she can create a custom Tasklet which takes the date of travel as input and books her regular flights and reserves a hotel room at her favorite hotel using her credit card details for payment – all this with a single click! Another example is a Bill-Payment widget with local-

language interface that navigates to the right web portal, performs the payment, and translates the results to the user's language. A Tasklet captures a sequence of web actions across multiple sites under a widget. It can also be used by portal providers and service providers to mobile-enable their business.

3. THE DEMO

We have a working prototype of the Tasklet authoring tool, Tasklet Repository Service and Tasklet execution environment for both Windows PC and mobile devices (both PDA and phone). The demo will be as follows. Firstly, we will demonstrate the Tasklet execution environment. We will show the working of a system-generated widget to perform the task of downloading a copy of an ACM research paper (*given a title, go to the ACM digital library and fetch the soft copy of the complete paper*). We use this desktop icon/widget which takes the title of the paper and returns a local downloaded PDF file. While the widget is performing the task, we will open its 'debug' window to pictorially show the user all the hidden web actions that is performed by the widget in order to complete the current task of fetching a research paper.

Next, we will show how one can easily author new widgets by just '*doing it once on the browser*'. The actions required for a specific web task is first recorded using a browser plugin. This recording file will be registered with our Tasklet repository service to generate a desktop widget automatically. During Tasklet registration, the user will be shown a list of possible input parameters to select from (*do you want to change the flight 'date', 'destination' in every run?*). On execution of the newly generated widget, these selected parameters will be used as inputs to the new task. One can note that the parameter list is dependent upon the type of task being performed and hence requires sophisticated analysis of the actions as well as web pages, which is performed by our repository service before the program synthesis phase. The Tasklet can also be composed with post processing services (Language translation). The Tasklet repository allocates a unique Tasklet URL to enable simplified triggering of Tasklet Execution from mobile devices. The repository services can be used to share this URL over SMS or email; accessing this URL results in cloud-based execution of the Tasklet. Two types of widgets will be generated – a thick widget that runs as a Windows Desktop application and another 'thin widget' that is meant for Mobile devices.

Specifically for demonstrating the creation of a new widget described above, we will choose the task of retrieving a person's horoscope from a website she trusts. We will use a web browser (with a plugin to enable recording) to *navigate to MSN page, go to the Horoscope page and extract content for my Sun sign*. While registering this new Tasklet with our Tasklet repository web service, we will demonstrate its features for sharing, parameterization and post processing. We will then show the auto-generation of Windows Desktop and Google Android widgets. On execution, these widgets will show that day's Horoscope in a local language (Hindi) – without requiring the user to open a browser or go to the website.

Lastly, we will show voice-based invocation of personal web tasks through our Tasklet Voice Gateway developed over HP Opencall Telecom Infrastructure. To show the voice demo, we will dial the Tasklet number from an SIP client, to hear the output - text of *my personal MSN horoscope* (that we created on the

spot). A second example of currency conversion that allows voice input for the task will also be shown.

4. RELATED WORK

Programmer-created mobile widgets and associated widget-stores such as Nokia [Widsets](#), Google [Apps](#), iPhone [AppStore](#) are becoming very popular. Developing a widget to do a personal task on any of the above platforms requires sophisticated web programming – which is not possible for a naïve web user or even for a tech-savvy person who has not kept abreast with web protocols and standards. Personalized portals, on the other hand, enable the user to customize only the layout of the website ([iGoogle](#), [My Yahoo!](#)) by selecting elements from existing services. Mashup tools like [Yahoo Pipes](#) allow users to compose new services by 'visually' specifying their information access needs (through a data flow graph) which may be non-trivial for a naïve web user. Some tools allow creation of widgets for web forms (Intel [MashMaker](#)) or some specific RSS feeds ([OpenKapow](#)) through PBE. Our solution goes beyond all of the above and enables end-users to simplify a complete task on the web with multiple 2-way interactions. We use a technique of programming-by-demonstration to exclude need for programming. Finally, web testing tools (like [iMacro](#), HP Mercury [QTP](#)), also provide a record and replay facility over websites. They have a very tight coupling between the recording and replay modules – and hence cannot work on mobile devices. Further, a lot more processing is needed to make a browser recording into a program and to accommodate changes in the inputs the next time the same task is repeated – bringing the need for parameterization.

5. CONCLUSION

In this paper, we presented our approach to radically simplify a user's personally valuable task on the web. We proposed the concept of a *Tasklet* to capture a sequence of web interactions needed to perform a web task. We presented a simple mechanism to create these Tasklets without explicit programming – through Programming-by-Doing. In this conference, we would like to demonstrate the creation, sharing and execution of these Tasklets on Desktops and mobile devices. Figure 2 shows some screenshots of the different phases of the demo. We believe that our proposal of end-user created widgets is unique and we look forward to test out the uptake of the proposed tool and solicit feedback from the expert Web developer community.

6. ACKNOWLEDGMENTS

We would like to thank our friends and colleagues at HP Labs India for their review comments and good Tasklet use cases. In particular, we acknowledge the interesting discussions and useful inputs from Ajay Gupta, Praphul Chandra and Nidhi Mathur. Special Thanks to Vimal, an interaction designer from HFI, Bangalore for the Portal design and Raghu for his support in the configuration of mobile internet devices.

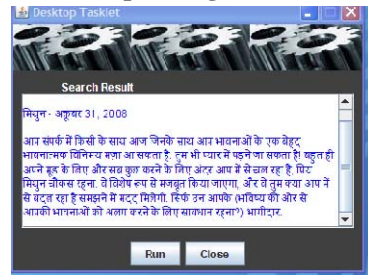
7. REFERENCES

- [1] Nokia Widsets <http://widsets.com>
- [2] Yahoo! Pipes, <http://pipes.yahoo.com>.
- [3] Intel MashMaker, <http://mashmaker.intel.com>
- [4] OpenKapow, <http://openkapow.com>
- [5] iMacro Macro Recording Tool, <http://iopus.com>

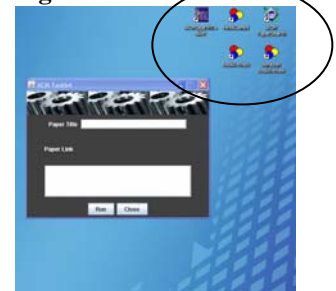


(a) Recording browser actions = Authoring a new personal web application (Tasklet)

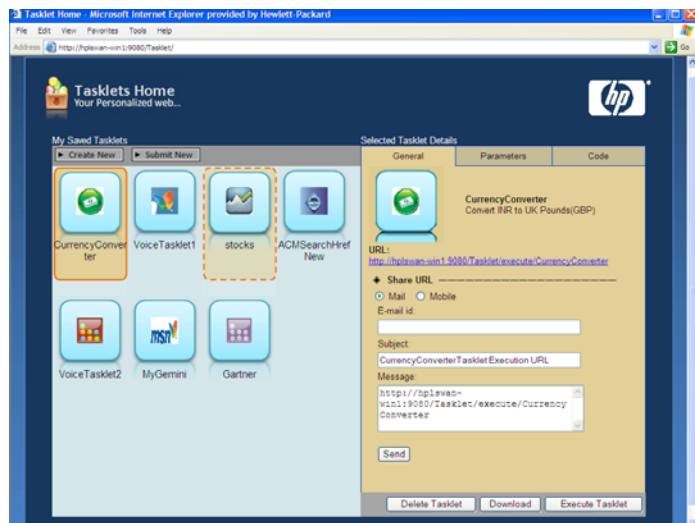
(b) System-Generated Gemini Horoscope Widget in Hindi



(c) Desktop Widget icons and ACM paper Search Widget



(d) A Mobile Widget on Android Simulator



(e) Tasklet Repository Services showing General Tasklet details and Parameters for selection

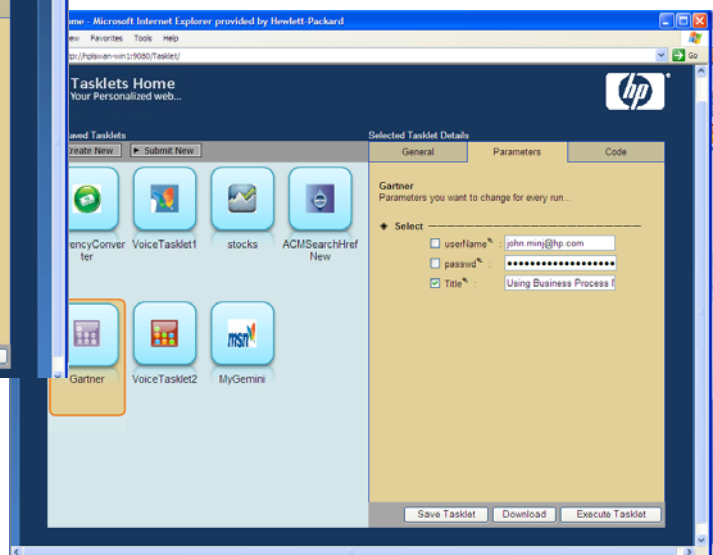


Figure 2: Screenshots of the proposed demo