Title: Using mashups to coordinate healthcare
Summary: IBM researcher Ohad Greensphan talks about the role of mashups in integrating disparate applications and Web-based technologies.
Featured researcher: Ohad Greenshpan
Duration: 6 minutes, 57 seconds

Introduction

Computer science has afforded researchers a wealth of applications and Web-based technologies to solve real-world problems. Unfortunately, many of these apps live in their own technology islands.

In this episode of Computer Science Spotlight, IBM research scientist Ohad Greenshpan talks about the mashups that he and his colleagues are using to address this technology "disconnect" in the healthcare industry.

Presentation

My name is Ohad Greenshpan. I'm an employee of IBM Research Lab in Haifa. I'm working in the healthcare and life sciences group.

My research focuses on mashups. Basically, a mashup is a Web integration technology that enables the user to assemble multiple applications into a single piece in order to achieve a certain goal.

Statistics say that there are 150 to 300 applications in one hospital. This is in small ones. In large ones, it can achieve the number of 3,000 applications.

So imagine yourself as a physician working in such a hospital trying to exploit all this data, being buried in all these applications, in order to get an answer about a specific question that you have.

So, currently these applications do not talk together. And the ultimate goal would be to develop enabling technology for all these systems to talk together.

Let's talk about two different scenarios. The first one that we are investigating is what we call “patient empowerment.” The goal is to provide the patient or the user at home or at [the] office the set of tools that would help him manage his data, collaborate with others, provide his knowledge to other patients on the Web.

So, let's take a simple example of a user that has the following applications on the screen:

Okay. The first one might be his personal health record; for example, his list of allergies, list of medications.

Even this personal health record application can be an integrated application retrieving data from various personal health records that he has in different hospitals. But let's treat this personal health record application as one application.

Another application would be, for example, a map. Another application would be an SMS alerting system. Another system would be an analytics component.

The patient goes to his personal health record. He sees something strange about his measures. He sees some irregular pattern. He sees that he has a headache or his heartbeat went high or blood pressure went high.

So instead of going to Google, he would like to use this system to come up with new knowledge. He would like to click on the list of measures that he has, or the irregular one, then see how it reacted along the years in the analytics component that we provide him.

He would like to take this list of medications and drag and drop it on the analytics component and see the correlation between the two.

Then when he sees something weird, he would like to grab

all of them and place it on the SMS widget and send an alert to the doctor or send a question. And then he would like to find the best doctor that can help him solve this problem.

There are these traditional integration approaches that take data, integrate it and provide it in a different, or in a uniform, way of representation. We have SOA, the Java J2EE solutions. We have databases that store data and provide it by using some UI. We have portals. Portals also collect data, integrate it in some way, and provide it using a Web interface.

But these technologies are still not sufficient to overcome this enormous number of applications on the Web today.

Another direction we're investigating is what we call the “future clinical workstation for physicians.”

Currently, systems that are designed for physicians tend to fail in phases of development and deployment, mostly because they do not fit the needs of a physician. A physician, for example, he has a very dynamic nature. He has to have a system that can support this change of context. So he can talk to a nurse and he will need some sort of data. And then he would like to switch and talk to

a patient and will need a different visualization and different view of the data. And then he would be on surgery and he will need other pieces of data.

So, we're trying to tackle this problem using the mashups to design the future clinical workstation for this community, which is very important.

So we first have to define what is an application. Then we need to define what would be the best platform to connect all these applications.

We will need to affect development of standards. For all these applications to talk to each other, they will need to speak the same language.

A current problem that I'm investigating now and we're going to present it in the upcoming ICDE 2009 conference is a mashup-based system that enables you not just to place your applications on the screen, but also to take advantage of all this intelligence on the Web. The research question would be how to use all these accumulating knowledge or intelligence on applications, connections, how to use it in order to assemble your mashup.

I think that [the] healthcare domain is developing very

rapidly today, but still is not mature to accept this kind of technology that is very, very popular with maps and other use-cases on the Web. So, investigating how the healthcare domain can be ready to adopt this technology. This, I think, will keep me busy for the next year or two.

Five years from now, the next step would be to understand how we can turn this into reality and how to place IBM as the leading vendor in this market with this promising technology.

You've been listening to Computer Science Spotlight, an IBM prodcast. Now available on iTunes.

Series producer: Barbara Finkelstein

Music: "Hot Swing" by Kevin MacLeod