Wormhole: A Powerful Data Mashup

David Widen
Zhangfan Xing
Jet Propulsion Laboratory, California Institute of Technology

Abstract

The mobile platform is quickly becoming the standard way that users interact with online resources. The iOS operating system allows iPhone and iPad users to seamlessly access highly interactive web applications that until recently were only available via a desktop or laptop. Wormhole is an AJAX application implemented as a smart web widget that allows users to easily supplement web pages with data directly from the Instrument Operations Subsystems division (IOS) at JPL. It creates an interactive mashup using a website's core content enhanced by dynamically retrieved image and metadata supplied by IOS using the webification API. Currently, this technology is limited in scope to NASA data; however, it can easily be augmented to serve many other needs. This web widget can be delivered in various ways, including as a bookmarklet. The underlying technology that powers Wormhole also has applications to other divisions while they are running current missions.

Introduction

Wikipedia provides an easy way to access a massive amount of information on the Internet. While it provides a wealth of information to users, Wikipedia does not utilize massive data stores that would greatly improve the overall user experience. IOS (PDS) contains NASA data about objects in our Solar System collected by various missions such as the Mars Exploration Rovers. The information stored in IOS would greatly improve the public's understanding and interest in our Solar System if a software application made it easy to supplement Wikipedia articles with this data. Wormhole accomplishes this by taking images stored in IOS and allowing any user to access these images while on a Wikipedia page with the click of a button.

Goals and Purpose of the Project and Implementation

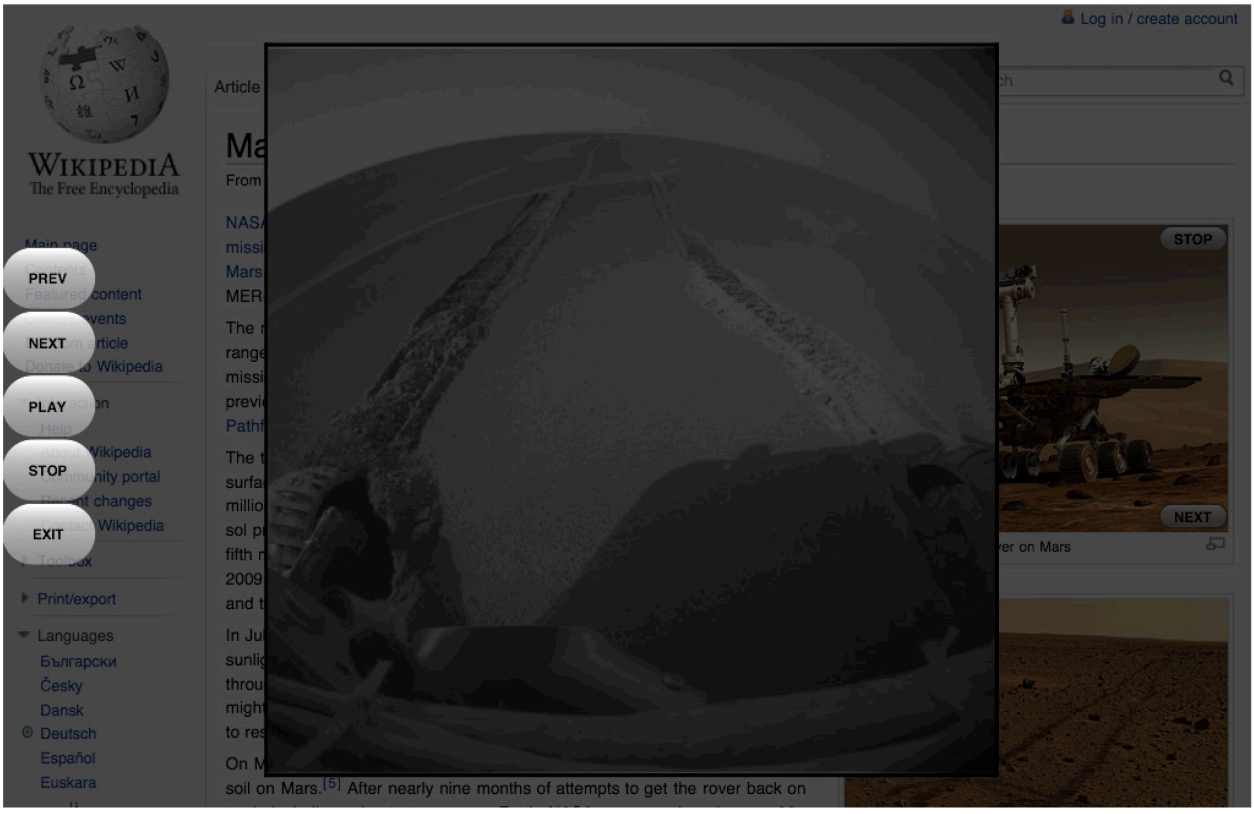
The main rationale of this project was that a person should be able to access non-classified NASA data in an easy way. Currently, huge amounts of NASA data are posted on the internet; however, it is almost impossible for the general public to access it due to the fact that one has to know exactly where to look. Searching can be quite difficult unless the user knows exactly what he/she is looking for. Wormhole solves this problem.

It is important that the general public can access this data because only a small sample of the actual research findings ever makes it to the public spectrum. People could easily find information about the final Space Shuttle launch, but it would be quite difficult to find out that what the Mars Exploration Rover (MER) was looking at as it was stuck in the sand on Sol 468. This type of information would show the public all of the hard work that NASA accomplishes and also fuel their interest in our Solar System and beyond. It may be exciting to read about what MER is doing on Mars, it is another thing entirely to see pictures taken through MER's cameras on a planet 77 million kilometers away from Earth.

The main goal of this software application was to provide a very easy to use user-interface. Current programs require the user to enter in very specific search criteria. The ultimate goal of this project is to use what particular Wikipedia page a user is on as the search criteria. Instead of searching for "Mars" with the date "May 25, 2007" and the instrument "Opportunity", a user would go to the MER Wikipedia page, click a button and immediately get data about Mars without performing a complex search. This reduces the barrier of entry for the common user.

A secondary goal of the project is to make it as easy to download and use as possible. This prompted the development of a "bookmarklet". A bookmarklet is essentially a JavaScript

program that is copied into the URL field of a bookmark. When the bookmark is clicked, it runs the associated code. This doesn't require any download or installation and runs as quickly as the user's internet connection. The final product's user interface is below.



This interface is optimized for use with iPad but also supports the web-browser. All the user has to do is use the given buttons to navigate through images, which is simpler than the current search mechanism.

My part of this project included the development and testing of the JavaScript program as well as designing the user-interface. I designed the software that performs three main tasks. First, the program dynamically embeds a script into the Wikipedia article that automatically makes a page request from a server containing all of the image data. Then, the program examines the data that was given to it and determines which images should be displayed to the user based on screen size and automatically resizes them if necessary. Finally, the program sets

up the user-interface and begins to asynchronously process user input. Currently, Wormhole can only integrate MER images with Wikipedia pages; however, future functionality for this software will include providing precise images to the user based on what page they are currently on.

I also wrote a data crawler to pull data from NASA IOS .img files and store them in a non-relational database, which will be used in future iterations of the project where image search capabilities are required. The database is implemented with MongoDB, and the data crawler is implemented in Python. Eventually several terabytes of IOS data will be index, currently approximately 380 gigabytes are indexed. A specialized search feature is currently being implemented that will allow very specific queries to be performed to improve the overall user experience of the program.

Career Impact

This project helped me realize that I enjoy software engineering as opposed to strict Computer Science research. My previous internship focused on high-level research. Although I accomplished that task, my passion is applying all of the analysis skills and experience I have gained from my classes and apply it to a hard software engineering problem. It reinforced my goal of earning a Master's Degree in Computer Science.

During this project, I learned many new skills. In terms of languages and programming paradigms, I learned: JavaScript, jQuery, AJAX, HTML, XML, Python, and NoSQL. I learned the importance of unit testing and how to perform an effective search from my cubicle coworker and also learned a lot of software development history from my mentor. I also learned the importance of setting goals and how to work on a project by myself which includes setting up a strict schedule and assessing how my current code fits within the entire scope of my project. I

also gained a lot of insight into the development process for software that is directly funded by NASA as well as through other sources.

Acknowledgements

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the NASA MUST program and the National Aeronautics and Space Administration.