

# **A Contact Manager for the New Web**

by Dan Kurtz

Faculty advisor: Professor Stephen Hockema, Faculty of Information Studies

**PROJECT OS I OA STUDENT EXPERIENCE PROGRAM**

**Winter 2006 Final Report**

April 1, 2007

# Project Goals

The principles, processes and tools collectively known as “Web 2.0” have been widely adopted by website creators on the Internet. While there are many criteria for determining which sites deserve the Web 2.0 label<sup>1</sup>, the most visible indicator is the presence of a set of tools common to most members of the category. These tools include RSS feeds, freeform tagging, open APIs and open data standards.

Most of the web-based applications that emerged from the first-generation web boom have now evolved into Web 2.0 versions. Online maps were once displayed on static pages that had to be refreshed with each new action; now maps can be navigated dynamically without page reloads<sup>2</sup>. Calendar systems have undergone a similar revolution, with the current crop of online calendar systems sporting overhauled interfaces and sharable data<sup>3</sup>.

Contact management applications, however, have been lagging in their adoption of these new tools. The “Contact Manager for the New Web” project aimed to correct this lag by using Web 2.0 innovations to improve the standard contact management workflow. This report describes the first prototype of Quaintance, a web-based contact manager with the following Web 2.0 additions:

- **Freeform tagging of contacts**, for fast, intuitive categorization and keyword-searching.
- Native handling of the **hCard microformat** specification<sup>4</sup>.
- **RSS feeds** for easy, flexible access to recently viewed and modified contacts.
- An **open developer interface** (API) which allows users to programmatically access and modify site data.

# Implementation

Quaintance was implemented between October 2006 and April 2007 by Dan Kurtz, with input and guidance from Prof. Stephen Hockema. It is written in Ruby using the open-

---

<sup>1</sup> One report on these principles can be found at [http://www.oreilly.com/catalog/web2report/chapter/web20\\_report\\_excerpt.pdf](http://www.oreilly.com/catalog/web2report/chapter/web20_report_excerpt.pdf)

<sup>2</sup> See Google Maps: <http://maps.google.com>

<sup>3</sup> See Google Calendar: <http://calendar.google.com>

<sup>4</sup> Described at <http://microformats.org/wiki/hCard>

source Ruby on Rails framework<sup>5</sup>. During development, a MySQL database was used as the backend, though the structure of Ruby on Rails is database-agnostic. Code revisions are stored in Subversion, an open-source revision control system<sup>6</sup>. The source code can be checked out from a public repository at <http://svn.brickswithoutclay.com/projects/quaintance/trunk>.

Like most open-source projects, development of Quaintance is ongoing. The current version is intended as a prototype. We prioritized new features over features that currently exist in contact-management applications. Features that users may expect in a mature contact-management application, such as search and photo storage, are currently incomplete in Quaintance.

## Features

Following is a description of the Web 2.0 features in Quaintance, and how they change the standard workflow of contact management.

### Freeform Tagging

Web 2.0 sites such as Flickr<sup>7</sup> and del.icio.us<sup>8</sup> allow users to describe content using lists of ad hoc tags, which result in a descriptive vocabulary uniquely tailored to users' sorting preferences. The Quaintance implementation of tagging allows users to tag contacts according to the scheme that best suits them. For instance, an academic who collects the business cards of five researchers at a conference may tag those contacts with the name of the conference or the contacts' research interests.

### RSS Feeds

RSS feeds normally deliver information about events, such as new postings to a blog or commits to a software repository. The RSS format is open and flexible, which allows feeds to be read and processed in a diverse array of contexts. Quaintance uses RSS to provide a portable list of the most recently viewed contacts, as well as the most recently edited contacts. Many applications and sites are now RSS-aware, meaning that users can access their contacts wherever they may be useful. Users who are comfortable with the Live Bookmarks feature of the Firefox browser, for instance, may subscribe to their most recently viewed contacts as a live bookmark, and access that contact information when they compose email in a web-based email system.

### Microformats

The current standard for machine-readable contact information is the vCard format<sup>9</sup>. Though widely supported, vCards are not human-readable, meaning that users who

---

<sup>5</sup> <http://rubyonrails.org>

<sup>6</sup> <http://subversion.tigris.org>

<sup>7</sup> <http://flickr.com>

<sup>8</sup> <http://del.icio.us>

<sup>9</sup> <http://www.ietf.org/rfc/rfc2426.txt>

wish to make their contact information public must maintain two different documents: a human-readable HTML file, and a machine-readable vCard file. Microformats are a new set of standards for HTML that allow common data types, such as events and contacts, to be marked up on a web page in a way that makes them readable by both humans and machines. An hCard is embedded directly into an HTML page, where it can be detected and parsed by applications.

The microformat movement is new even by the rapidly evolving standard of web trends, so it has not gained broad adoption yet. Where adoption has taken place, it is usually in the form of microformat availability, rather than processing. Popular sites such as Flickr mark up user profile pages using hCard, but there is a shortage of tools that can process this new wealth of machine-readable information. Most existing hCard-aware applications merely transform hCards into vCards for use in other contexts.

Quaintance supports hCards natively. Users import hCards by entering a URL containing contact information marked up according to the hCard specification. From there, the import process is no different from the process in other address book applications.

The system also supports the vCard standard. vCards can either be imported as a file or provided as a URL, as with hCards.

### **Developer API**

The Developer API makes Quaintance both a site that can be used directly by users and a repository of contact data that can be accessed and manipulated by other applications. Developers can use this API to synchronize contacts between Quaintance and other applications, or to arrange contacts in a manner not supported by the system.

## **Future Directions**

Quaintance is intended both as a usable application, and as a platform for experimentation with new web technologies. Given those goals, there are several dimensions along which the current prototype can be improved.

### **Updatability**

When Quaintance imports an hCard from a source URL, that URL is stored with the contact. This makes it possible for Quaintance to periodically check that URL and update its records if it notices a change in the contact information. However, this functionality is not yet implemented.

### **Compatibility**

There are numerous established standards for storing and accessing contact information. One broadly supported standard is LDAP, commonly used in corporate environments. Another is SyncML, which is implemented on many cellular phones and PDAs.

Because it is open-source, Quaintance has no incentive to keep users' data locked in and accessible only by an "approved" set of tools. Users should be able to store their data in Quaintance and access it using their preferred tools, even if those tools do not

support the Quaintance developer API. Quaintance should support a plugin architecture whereby developers can write modules that allow it to communicate using standard protocols.

### **Extensibility**

Most contact-management applications store the types of contact information found on a business card: phone numbers, street addresses, URLs, etc. However, there are now many other types of information that a contact may have publicly available.

Quaintance should make it easy to create plugins which can aggregate these types of data and display them with a contact, so that a contact's Flickr photos, blog posts and recently-bookmarked del.icio.us sites can be seen alongside phone numbers and email addresses.

## **Benefits to OSIOA**

The main deliverable for this project is a body of source code that implements, and proves the feasibility of, the ideas listed above. The code functions not only as a working contact manager, but as a platform for future innovation. Its openness allows for community-driven exploration and innovation that is not possible under the constraints of closed, centralized development models. We hope that users take advantage of the open architecture to build upon the existing functionality of Quaintance, and push the boundaries of contact management in general.