PROJECT NAME: Previous Contacts Identification Using Public APIs

PROJECT DESCRIPTION:

This proposal is based on a suggestion by Eaton posted on Groups.Drupal about Google Summer of Code 2008.

Drupal is a content platform that allow individuals, organizations and communities to express themselves and organize their knowledge and ideas. It has a wide range of uses thanks to a large number of extra features provided by add-on modules, which makes Drupal great for social applications, such as social networks and forums.

Many features, such as user profiling and user relationships, allows better interaction among site users. An issue though, is that users generally have their contacts hosted on other applications such as email services and can't take advantage of it. This problem can be minimized now that many contact based applications such as Gmail are providing APIs to export user's contact data to other application.

This project will design and implement a Drupal module that allows users to identify their contacts that already are in a Drupal site. We would implement this idea for Gmail contacts using Google Contact API, but the module will be designed to be easily integratable with other web services (other email clients and social networks that expose their APIs).

IMPLEMENTATION DETAILS:

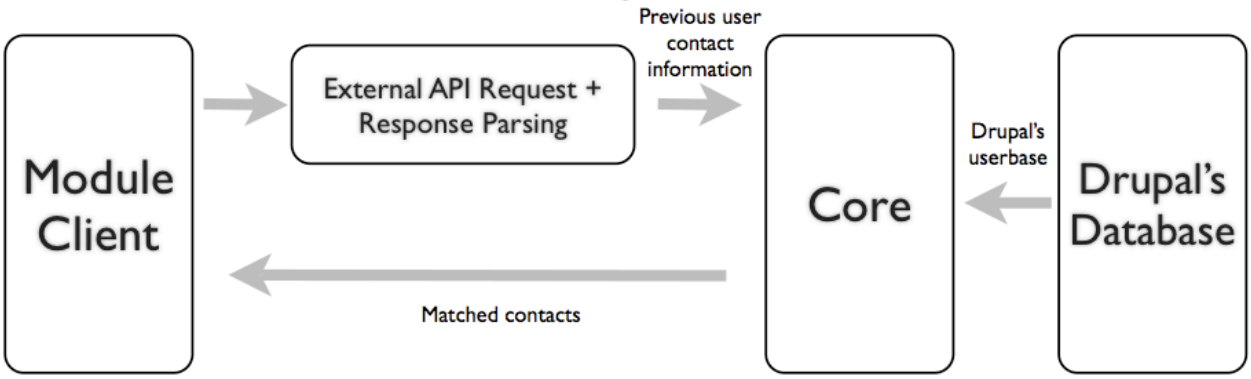


Figure above shows a preliminary architecture for the solution. This approach uses interfaces in order to support the implementation of many web services (e.g. gmail, hotmail, opensocial) in the same module. It also allows that many clients (e.g. invite module, user relationships module) use its service: contact matching.

While the core and its relation with the database will have one implementation, it's expected that we'll have many API support implementations and many clients using it. The scope of this project will be to implement the core, the gmail (Google Contact Data API) implementation and a "client" that shows the user which of his previous contacts are in the social networking site.

Core Overview

The core idea is simple: its input is a contact list provided by the API and its output is a list with contacts that were both in Drupal's user-base and the input list. This central feature implementation doesn't seem to be complicated, although we could spend some time to find the most efficient way to do it and with error handling .

On the other hand, determining format and structure of input and output will require more attention, since we want to support many external APIs and serve many internal clients. Input and output format definition needs further investigation, but possibly will be JSON, since it's widely used by todays web services APIs.

We don't want to limit the information handled just to names and emails, so we would investigate candidate API's to find common information (e.g. email, name, phone number) that would compose the input structure. Other interesting approach would be to support "wild-card" fields associated with each input contact that would be passed to the selected output. In this case, the only required field would be email (our business key).

We also want to provide flexibility to allow as many modules (clients) as possible using this service. The use of "wild-card" fields helps on that, but we should also provide different outputs, based on the needs of our candidate clients. For example, a output that include contacts not yet registered on the site could be used by a client that allow inviting your contacts to join the site.

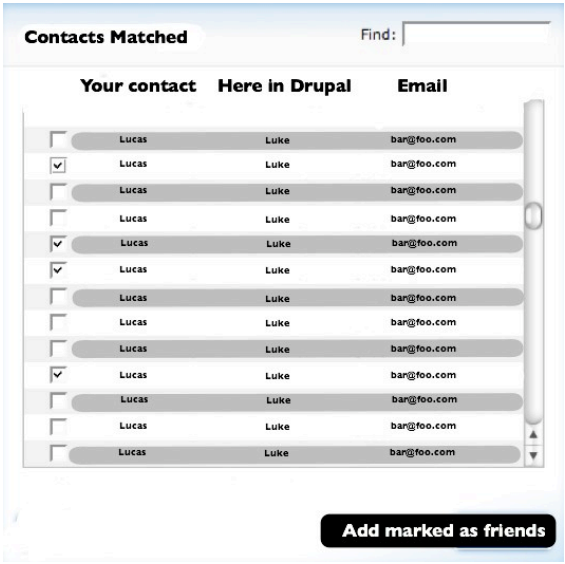
Google Contacts Data API Quick Considerations

GCData API allow web applications to perform CRUD operations on user Gmail contacts through HTTP requests. Access is granted after a AuthSub authentication, which redirects to a google page to authenticate user data. After that, the page receive a token to be sent with every request (the token can be stored in the session). We could implement that directly or user look for a ready solution on PHP language.

GCData API can return atom (default), rss, or json feeds. The feed's type choice will depend mainly on what types other API's use, but also on what is more reasonable to implement. It also allow adding contacts to gmail, so we could do the reverse functionality (export drupal contacts), but it would require some chances on what where proposing on core.

Client Preview

As Alex suggested on the idea topic, we would do something like Plaxo Address Book Access Widget (http://www.plaxo.com/api/widget_demo). When looking for friends, users would have an option to find previous friends, which will redirect to a google page for login and on return will see a list of friends already on the network. There would be links for user profile in each line and buttons, for example "add marked as friends" (if we do integrate with User RelationShips Module).



Deliverables

As we detailed above, the deliverables of this projects should be:

- A fully functional core (contact matcher) with a well defined interface for new web services implementations and an exposed API for new clients.
- A Google Data API implementation
- A UI (client) that requests user data and show matched contacts

BACKGROUND AND MOTIVATION:

Sorry about all the sites referred here being in portuguese. If you want more information about them, please don't hesitate to contact me.

I was introduced to PCs when I was 8 years old, in 1996, when my family bought a 486. Computers weren't that usual but I managed to learn the basics working in DOS/Win3 environments. One year later, I started using the internet to chat and play games such as Age of Empires. I wasn't that interested with technical stuff, but motivated by the game clan (www.onuclan.com) I was part of (my first experience with web communities), I learned HTML and made its simple and statical web page. Later on, I had contact with PHP and MySQL using Invision Board to create my clan's forum. In that time, I was just superficially using technology, but it influenced my decision to dive into computer engineering.

In 2005, I was admitted to Instituto Tecnológico de Aeronáutica (www.ita.br), one of the best engineering colleges in Brazil. Despite the strong academic demands from my college, during first two years I worked for a NGO (www.itajunior.com.br) that developed engineering solutions for small companies and other organizations. There I had contact (not as a developer) with many projects, including the development from scratch of a PHP+Smarty+MySQL CMS for ITA's alumni association (www.aeitaonline.com.br).

Last year, I was highly involved in a open source project called Doacoes (check our project page at http://www.ita.br/~lucas), which is a java application using PostgreSQL to be used for brazilian NGOs donation control (I developed the event administration module). I'm currently working in a Ruby on Rails project called ITA Alumni (check our blog at www.itaalumni.com), that aims to build a alumni social network with a donation module. I'm responsible for the profile system.

I also like to do experiments, my last one was a opensocial (html+javascript) gadget using google maps API that communicates restfully with a RoR server. It's called ovvo and will go public next month. This project was very enriching, since I could understand and use concepts as REST and AJAX.

As you can see, I have great interest in projects and organizations meant to be useful to many people and in service oriented web apps. That's why I believe Drupal and this project involving Google Contacts API would be a perfect fit. You can be sure I have all the motivation to overcome my small experience as a web project developer and make a high quality, very useful module to Drupal.

BENEFIT TO DRUPAL COMMUNITY:

By improving social networking features on Drupal, the sense of belonging to a community in such an online environment would grow even stronger. With their previous friends, members would interact and share ideas more enthusiastically. This multiplies Drupal's content generation and community involvement potential.

Drupal would be ready to use APIs that will proliferate in the next months (for example, Microsoft just launched Windows Live Contacts Data API for social networks). This can be one more reason to make a choice on Drupal's platform.

It's also important to highlight that, although this project implements an useful networking tool (finding previous friends), it also provides a service to any module that wants to use user's contacts.

PLAN OF ACTION

Task 1: Studying Drupal's policies, studying relevant parts of the documentation (specially about module hook system) and getting familiar with the source code that I would have to work on.
Duration: 2 weeks
Deliverable: Module's implementation model

Task 2: Investigate public contact APIs and open source applications that already implemented this idea (if any). Investigate potential clients for the service and what information they would need. Look for modules/code that could be reused.
Duration: 2 weeks
Deliverable: API/Core and Core/Client Interfaces

Task 3: Core Implementation
Duration: 2 weeks
Deliverable: Already Clear

Task 4: GCData API Implementation and Integration with Core
Duration: 2 weeks
Deliverable: Already Clear

Task 5: Client Implementation and Integration with Core
Duration: 2 weeks
Deliverable: Already Clear

Task 6: Through Testing and documentation of the work done
Duration: 2 weeks
Deliverable: Already Clear