Connecting Applications - Plone as a Tool for Connectivity to other Resources

David Breitkreutz[1], Russell Sim[2]
[1]eResearch Centre, James Cook University, Townsville, Australia, david.breitkreutz@jcu.edu.au
[2]Australian Research Collaboration Service (ARCS), Melbourne, Australia, russell.sim@arcs.org.au

INTRODUCTION

Collaborative research increasingly results in greater complexity through an ever-deepening richness of communication methods. Traditional, monolithic methods for the creation of complex, multi-purpose software packages have given way to more agile, inter-connectable products. This change is realised through the availability of common Application Programming Interfaces (APIs). The Australian Research Collaboration Service (ARCS) has sought to integrate a variety of applications together into its online Content Management System (CMS). This CMS, powered by Plone, is integrated with the ARCS Database Service, Google Applications, and other open source products, as a means of delivering a content-rich, collaborative environment.

This presentation presents a case study involving one of the research customers. that ARCS is involved with. The authors discuss the customer's specific requirements for a complex collaboration environment and demonstrate how the Plone CMS software is adaptable to requirements, through the integration of a number of different products.

COLLABORATIVE ENVIRONMENTS - AN OVERVIEW

A collaborative workspace environment is one where related users can share a common computing system for the benefit of improved communication. One such commonplace environment is e-mail, and whilst not significantly complex, organisations and employees worldwide utilise such a service to share information, and do so quickly. In the traditional sense, e-mail is not strictly as 'collaborative' an effort as other communication methods, primarily because messages are individualised and can only be shared or forwarded on a one-by-one basis. ARCS has ascertained that such communication is tedious, coherence is difficult, and there is no central 'repository' that collaborators are able to access. From this, it is evident that researchers and organisations alike require a step up from such an environment.

When considering a system to improve over simple communication methods, it is evident that a large proportion of users require the same features. The most significant of these aim to improve over simpler tools - such as e-mail - and make the collaboration process easier, and more accessible to users. By taking a powerful CMS as a starting point, and integrating it with other commonly-used, higher-level tools, it is possible to relieve the burden imposed on the users.

ADAPTING TO REQUIREMENTS

As can be expected, each different research group typically has a distinct set of requirements for a given collaborative workspace. These specifications need to be tailored in such a way to appease the needs of the given set of people, and yet the resulting software features that manifest must be, at least in part, reusable. The reason ARCS has selected the Plone CMS to act as a basis of its development efforts is because this software is highly extensible and code reuse is typically straight forward. Being built upon the Zope Application Server, Plone and its extensions are constructed using the Python programming language. Some authors, such as Zelle [2], have referred to Python as having 'magical' qualities, and likening the Python interpreter as a degree of 'genie'. These qualities are evident when a CMS, such as that of Plone, needs to be customised as clients frequently have different requirements for their workspaces. Having a flexible programming language underpinning the software is a crucial aspect of being able to deliver a suitable system befitting specific requirements imposed by different organisations.

ONLINE SERVICE INTEGRATION

More and more organisations are moving towards using Google's various applications to deliver services to their members and employees. Using the Google Data (gdata) API, third-party software can connect and exchange information with the various services. In the case of ARCS, the relevant Python gdata API has been used to communicate and display data from Google within Plone, presenting users with quick and easy access to information they need. At present, connectivity exists between Plone and Google Docs, Google Calendar, and Google Mail (see Figure 1). In order to achieve this, shared authentication is established, with the user granting access as required, thus allowing the applications to share data. Beyond these current capabilities, additional functionality exists for Plone to be extended to integrate with Google Contacts, Google Maps, and more, through the use of each relevant interface.

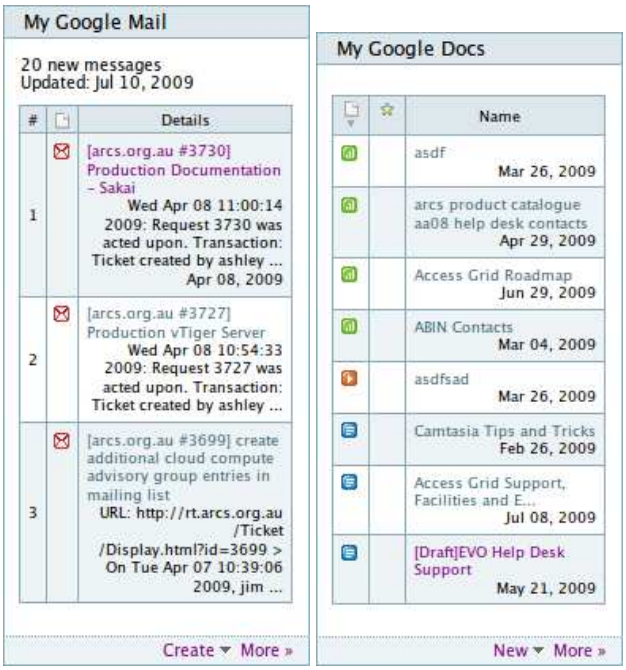

Figure 1: Examples of Google Apps Integration within a Plone Environment

Further to collaboration tools, another important aspect for integration is that of social networking. Users of online collaborative environments upon computer networks themselves represent a form of social networking [1]. With the advent of *web 2.0*, collaborators need to have immediate access to see what their colleagues are doing, in order to further their common goals.

One such way of achieving this goal is the integration of the OpenSocial API [3] within the Plone environment. Collaborating together across multiple software products can be a stumbling block for users if they are restricted to one set of *friends* and otherwise tedious for developers due to the programming required. Adhering to an open, common API like OpenSocial for such integration reduces the inherent complexity for maintaining the social network and the applications themselves. Such functionality also allows for easy access across multiple platforms and software to a common set of information, and the ability to update information wherever one is. Another benefit of using such a technology is that there are a number of larger social networking sites that offer support, including LinkedIn, MySpace, orkut, and Yahoo!. Implementing this within Plone would improve its collaborative qualities and allow users quicker access to the information they require about what their colleagues are doing at any given moment.

Overall, the only limitations on such integration are the degrees to which the services are exposed through their APIs., and the extensibility of the APIs themselves. For services that are controlled and administered by a third party, such as Google's various applications, the limitation is also constrained to what their Terms of Service allow.

SUMMARY

ARCS aims to leverage the above-mentioned emerging APIs within a *web 2.0* framework in order to create customisable, integrated platforms for eResearch collaboration. Using a *Service Oriented* model, ARCS aims to combine the highlighted technologies in such a way that they can be tailored for each research group's requirements.

Details of the work described here is accessible at: http://www.arcs.org.au

REFERENCES

1. Wellman, B., *An electronic group is virtually a social network.* Culture of the Internet, 1997. p.179-205
2. Zelle, J.M., *Python Programming: An Introduction to Computer Science.* Nature, 2003. Franklin, Beedle & Associates, Inc.: Sherwood, United States of America.
3. *OpenSocial - It's Open. It's Social. It's up to you.* Available from: http://www.opensocial.org/, accessed 27 June 2009.