Reefknot : A Calendaring Toolkit in Perl

Kirrily "Skud" Robert

e-smith, inc.

1. The desperate need for open source calendaring tools

Linux or other open source operating systems can largely replace proprietary systems as network or file sharing servers. Many businesses use them for this purpose. However, there is one network service provided under Windows NT with which open source solutions can not compete: calendaring and workflow.

Microsoft Exchange is a mail and groupware server for Windows NT. The preferred client for connecting to it is Microsoft Outlook. Together, they provide a suite of groupware/workflow tools which many businesses find essential to their smooth running.

The problem is that Exchange/Outlook calendaring (& etc.) are tightly coupled to the Exchange mail server, the IIS web server, and Windows itself. This means that open source software must be able to replace *all* of them in order to be accepted. A proposed solution which uses (for instance) Linux or BSD, Apache, and Sendmail will be found unsuitable because it does not support calendaring. Since all the other Microsoft software is required to support calendaring anyway, Microsoft-running businesses see no need to replace any of it.

It is not the intent of this white paper to discuss *why* one would want to replace NT with Linux or BSD, IIS with Apache, or Exchange mail server with sendmail, postfix, qmail, or whatever. It is assumed that the reader understands the relative costs, benefits, and problems of each. We will, therefore, assume that replacing an all-Microsoft

environment is considered a desirable outcome.

A number of efforts have begun to address the need for open standards and software for calendaring and workflow. Foremost among these is the IETF standards-track document for calendaring, RFC 2445, which describes a core object specification for Internet Calendaring and Scheduling. This specification is referred to as "iCalendar".

At the time of writing, there is one reference implementation of the iCalendar specification. In order for iCalendar to become an actual standard (as, for instance, TCP/IP is) there need to be at least two independent reference implementations of the specification.

The official standardisation of RFC 2445/iCalendar (and subsequent pressure on software houses to support the standard) is not the only benefit of developing another iCalendar implementation. More iCalendar implementations will:

- help us (the development community) refine our understanding of calendaring systems and improve the available technologies
- provide wider choice for users
- promote hybrid vigour by cross-fertilization of ideas between projects
- provide a base for project management systems (another much-needed area of software in the open source arena)

2. The toolkit approach

The Reefknot project has decided to take a "toolkit" approach to developing iCalendar-compliant software. That is, we hope to develop a set of tools and libraries which make it easy for people to develop iCalendar software. We hope that this will help rapidly multiply the number of calendaring tools and systems available, giving users even greater choice and letting us further reap the benefits outlined above.

It is our hope that the Reefknot toolkit will allow a competent developer to put together simple, interoperable calendaring tools in only a few lines of code.

The buzzword list:

- interoperability
- platform independence
- standards compliance
- component architecture
- fast development of custom applications

3. Why Perl?

The initial choice of Perl as a development language was based on the early development team's familiarity with the language. However, we strongly believe that this choice stands up well on the basis of functionality, not just personal bias.

- Perl is widely and freely available for many platforms
- Basic iCalendar libraries already exist for Perl
- Perl is one of the foremost languages for Internet development. Modules exist for most network protocols and many client/server systems are written in Perl
- Perl is ideally suited to manipulating textual data; iCalendar data is textual
- Perl is well suited to application development. CPAN modules exist for:
  - CGI development for the World Wide Web
  - X11 GUI development (Gtk)
  - Windows development
  - Database connectivity (abstracted through the DBI interface)

- Perl is known to a large number of programmers and is easy to learn. Even inexperienced Perl programmers should be able to build Reefknot applications with little difficulty.

Other languages suggested for this project include C, C++, Java, and Python. While we recognise the benefits of these languages and do not intend to suggest that they cannot fulfil the features listed above, we know that we (the core project team) will get a working system out the door faster in Perl than in any other language. If others wish to build on our design to implement Reefknot-like toolkits in other languages, we encourage them and will assist them in any way possible.

4. Immediate plans

As of the end of November 2000, we hope to begin on the following tasks:

- investigate the current state of the art by examining as many existing calendaring systems as we can get our hands on
- seek out real end-users (not necessarily developers) to tell us what they really want in a calendaring system
- liaise closely with the iCalendar working group
- liaise closely with other active open source calendaring projects, such as Evolution
- develop an architectural design for Reefknot
- begin work on Perl modules forming the base of the Reefknot toolkit
- begin work on applications based on the Reefknot toolkit

It is hoped that within 6-12 months we will be able to release a stable toolkit and sample applications built with the toolkit. These applications should provide sufficient features to begin to compete with existing calendaring systems.

5. Future directions

One direction which is of particular interest to us and to many of the people we've spoken to is project management, another application field in which there is no prominent open source candidate.

To the best of our knowledge, there is not currently a standard or specification for handling project management data. It seems likely that we will have to develop this specification if we choose to extend Reefknot along these lines.

Some users have expressed a desire for integrated contact management, as exemplified by the Windows application "ACT". This is another area which will probably warrant further investigation and development.

We hope that third parties will build upon the Reefknot project to extend its utility into other languages or to create custom applications. We see potential for a wide range of calendar backends and servers (Oracle, MySQL, LDAP, HTTP, SMTP, WAP...) and clients (WWW, X11, Windows, email, PalmPilot...), and will be encouraging others to develop these.

6. References and contact

Reefknot website

http://reefknot.sourceforge.net/

RFC 2445 (iCalendar)

http://www.ietf.org/rfc/rfc2445.txt

Perl

http://www.perl.com/