INFORMATION TECHNOLOGIES COMMITTEE
MINUTES OF MEETING
FEBRUARY 7, 2006

[In these minutes: Open Source Software, Site Licenses, Webmail, Future Agenda Items]

[These minutes reflect discussion and debate at a meeting of a committee of the University of Minnesota Senate; none of the comments, conclusions or actions reported in these minutes represent the views of, nor are they binding on, the Senate, the Administration or the Board of Regents.]

PRESENT:
Andy Lopez, chair, Nancy Herther, John See, Dale Swanson, Deanette Schmidt, Eric Celeste, Linda Jorn, Douglas Ernie, Joan Hughes, Jim Waddell, Eli Mayfield, Mahmoud Sadrai

REGRETS: Mark Sanders, Alan Ek, Greg Laden, Stuart Speedie, Myron Lowe, John H. Miller

ABSENT: Stephen Cawley, Christine Blue, David Hauser

GUESTS:
Mark McCahill, director, University Technology Development Center; Phil Kachelmyer, director, University Computer Services; Kevin O-Rourke, manager, Internet Services Group

OTHERS: Bernard Gulachek, David Moretz, Shih-Pau Yen

I). Professor Lopez called the meeting to order and asked all those present to introduce themselves.

II). Members unanimously approved the December 6, 2005 minutes.

III).
Professor Lopez welcomed Mark McCahill, director, University Technology Development Center who was invited to provide the committee with information on open source software. Mr. McCahill began by providing a definition of open source software:
"Software whose source code is published and made available to the public, enabling anyone to copy, modify and redistribute the source code without paying royalties or fees. Open source code evolves through community cooperation.
These communities are composed of individual programmers as well as very large companies."

Open source software has rapidly gained popularity because the growth of the Internet has aided the formation of ad-hoc communities, which have fostered the advancement of open source software. The Internet is a perfect platform to bolster shared repositories of information. Mr. McCahill also noted the convergence of licensed open source software models e.g. GNU General Public License (http://www.gnu.org/copyleft/gpl.html) & The MIT License (http://www.opensource.org/licenses/mit-license.php), which represent two different philosophies around licensing code.

Open source software networks allows users to:
- Choose an alternative to closed/proprietary systems.
- Avoid restrictive vendor licensing agreements.
- Open up system architecture as a means to add value. Several large companies, e.g. IBM and Hewlett Packard, devote a significant amount of their resources on open source efforts/advancement.
- Possess more flexibility and control.

With this said, however, whenever software is not purchased from a vendor there are support issues to consider. It is at this point the user wants to judge the strength of the open source community.
- Enjoy the strength of the community, which has fostered its development.

Historical examples of open source initiatives over the years include:
- Developer tools such as Perl language and libraries (used for Forms Nirvana and EGMS).
- Operating systems – Linux servers and desktops.
- Application suites such as OpenOffice/StarOffice. (StarOffice is the commercialized version of the open source code developed by OpenOffice.org with some additional features).
- OIT Linux support in 2000.
- myU Portal.

myU Portal, is an example of a very successful enterprise class open source initiative, which currently gets on average 10,000 – 12,000 log-ins per day during the work week. Attributes of myU Portal include:
- Metadot, an open source software.
- Perl programming language.
- MySQL Database.
- Apache web server.
- Linux operating system.
- LVS load balancer.

Open source community tools include:
- Mailing lists with searchable mail archives.
- Shared repositories and versioning. A place where old and new code can be accessed.
- Bug reporting and tracking capabilities.
- Shared authoring of website documentation.
- Chat/IM, which allows for real-time conferencing.

Examples of open source tools that the University has applied outside of the software development arena include but are not limited to:
UMWiki – a shared website that permits multiple people to author web pages while keeping track of versions; Jabber, a chat client, which allows for real-time conferencing, etc.

Future directions around open source software promulgation at the University:
- Creation of a University community site to share reviews of open source software.
- Participation in the Croquet Project's Core Educational Partners initiative.
- Promotion of Squeak, a simulation and application development tool.
- Exploration around the feasibility of offering Moodle (http://moodle.org/) and Sakai (http://www.sakaiproject.org/), which are open source course management tools, as an alternative to WebCT Vista.
- Expansion of the University's source code repository infrastructure to include Subversion (http://subversion.tigris.org/).
- Incorporation of Mantis, a bug-tracking feature.

To summarize, Mr. McCahill stated that while open source software might not be the right choice in every situation, it should at least be considered.
OIT needs to take the lead in making an open source infrastructure readily available to the University community.

Questions/comments from members following Mr. McCahill's presentation included:
- Who decides if a user's open source enhancement recommendations are incorporated into an updated

version of the product? Usually core developers decide what will be included in the next revision. If there is a disagreement between the developers about what should be included, it is not uncommon for the product to branch off into different directions with the possibility they may collaborate again at some point in the future.
Also, the community that uses the product may influence whether an enhancement gets incorporated.
- It is commendable that OIT is exploring other course management tools options such as Moodle and Sakai.
- What channels do groups that create code need to go through to get permission to release it as open source at the University?
  Mr. McCahill recommended following the funding source (internal or external) for the code that seeks to be shared in an open source format to order to receive permission. An argument for putting material out in an open source format is that others will help maintain it. Also, it is often good for the University's reputation.
- Is the University creating its own open source community tools or does it use SourceForge.net, an open source software development web site?
  Mr. McCahill stated that he is making some of the tools available here so the University has the option of being able to handle smaller projects that would not be appropriate for SourceForge.net. The community that one wants to attract should influence which software development site is chosen.
- Open source software has matured at an impressive rate over the past 4 – 5 years. Serious applications can now be built using this type of software.

IV).
Next, Phil Kachelmyer, director, University Computer Services, shared information on software site licenses.
A software site license is an arrangement with a software vendor to obtain and redistribute software licenses for one or more of its products for departments.
Certain licenses are purchased on a site-wide basis, while others are purchased in large quantities to get a lower price.
University Computer Services works with multiple departments to aggregate the University's purchasing power.

The process for requesting a site license involves sending an email to software@umn.edu. In order for a site license to be considered, the following criteria must be met:
- More than one unit must be able to benefit from the arrangement.
- The negotiated pricing must be less than what the license could be purchased for through other channels.
- There must be no degradation in either the product or the service provided by the vendor.

Once a request has been received, University Computer Services investigates its viability.

Types of site licensing arrangements include:
- 'Common good' licensing – This type of licensing uses central funding sources to make its purchases. Examples of 'common good' licensing arrangements on campus include:
  - Microsoft Campus Agreement for departments.
  - Microsoft Campus Agreement for students.
  - Symantec Client Security/Symantec Anti-Virus for departments and individuals.

- Software licensing for which there is a chargeback to departments. Typically this type of licensing is purchased centrally with the right to distribute. Characteristics of this type of licensing include:
  - Departments pay a licensing fee to participate.
  - Most of these licenses are renewable annually.
  - A majority of these licenses can only be used on computers owned by the University.

  - Besides the Twin Cities campus, many of these licenses are available to the coordinate campuses.
  - Certain products may be licensed on a per-department basis, which allows the software to be used on all computers owned by a particular department. Other products must be purchased on a per-computer basis.
  - Some of these licenses offer rights for individual use.

- Software licensing for individuals. Examples of this type of licensing includes:
  - Mathematica
  - SAS
  - MircroSoft Student Select – This agreement gives students the right to purchase Microsoft software not covered in the Campus Agreement at a discounted price e.g. FrontPage, Project, Visio, OneNote.
  - MicroSoft Academic Select for faculty and staff – This agreement gives faculty and staff the right to purchase software at the same discount as Student Select e.g. Windows upgrades ($75), FrontPage ($47.91), OneNote ($16.71), etc.

For additional information on departmental purchases, please visit the following URL: http://www1.umn.edu/adcs/site/ and for student, faculty and staff purchases: http://www1.umn.edu/ucs/

A member recommended that University Computer Services make it easier for people to find licensed downloadable software by having one website that lists this information by product instead of licensing strategy.

V).
Prior to demonstrating GopherMail, Kevin O'Rourke, Internet Services Group manager, provided members with background information on this new Webmail client. The following information was shared regarding GopherMail:

- It is based on "Prayer" created at the University of Cambridge.
- It is integrated with the University's:
  - Central Authentication Hub (CAH).
  - Spam reporting.
  - LDAP Directory Services.
- It does not rely on an external web server or PHP infrastructure.
- It utilizes IMAP and holds open up to give connections similar to desktop clients.

GopherMail features include:

- Ultra fast response time.
- Saving draft messages.
- Attachment forwarding.
- Downloading of entire mailboxes (UNIX format).
- Searching based on sender, subject, date, size, and text.
- Spell checker.
- Threading option.
- Import/export of address book.
- Populate address books via directory search.
- View and forward HTML.
- Define roles.
- Define personal dictionary words.
- Aggregate operations on messages.
- Enhanced rendering of quoted text.

- Tailor folder views.
- Interface to server setting.
- Use nicknames (aliases) for composition.

Outstanding GopherMail issues that need to be addressed include:

- Spell check continues into quoted text on replies.
- Address book interoperability with WebMail 3 and WebMail Pro.
- LDAP look-up without sufficient criteria.
- Sort using ‘from’.
- Usability concerns e.g. renaming folders, logging back in, separate compose window).
- Help screen.

Next, Mr. O’Rourke provided members with a demo of GopherMail.

Member’s questions/comments regarding GopherMail included:

- Is there a limit on the size of individual GopherMail accounts? Users of GopherMail are allotted 1 GB (Gigabyte) of storage space.
- Does GopherMail allow for signatures? Yes, stated Mr. O’Rourke.
- Will OIT support WebMail Pro, WebMail 3 and GopherMail? According to Mr. O’Rourke, WebMail Pro will be phased out.
  GopherMail, because of its architecture, will be able to support older browsers like those currently served by WebMail Pro.
  In the not too distant future, another Webmail client, based on the University of Oregon’s Alpha Mail, will be released at the University. Alpha Mail offers some very interesting features.
- Is there an advantage to using WebMail 3 over GopherMail? Mr. O’Rourke stated that most Webmail users are very idiosyncratic about the features they like and dislike, and, therefore, it is often a matter of personal preference more than anything.
- The number of mail clients supported by the University should be limited because too many options will cause confusion.
  Mr. O’Rourke stated that the goal will be to move WebMail Pro users to the University’s new mail clients because this will save on the University’s central resources.
- Is GopherMail significantly faster than WebMail 3? Yes. GopherMail maintains a constant connection. WebMail Pro does not use a proxy, and is much slower than WebMail 3.
- Are there plans to make the new mail clients more handheld tool friendly? Although this request has not yet been addressed, there are plans to do so.
- Does GopherMail have a manual spam filter? No, however, there are plans to offer a server-side filtering option that will filter mail when it is delivered. This solution will apply to all mail clients, webmail clients and desktop clients.
  The next mail client will come with client-side filtering capabilities, but this is less desirable since the filtering will depend on which client is used and many people use multiple clients (e.g. a desktop client in their office, and a web client on their handheld device).
- Is consideration being given to formats like Google’s Gmail? Gmail uses virtual rather than actual folders. These virtual folders depend on stored google search strategies. OIT has no immediate plans to pursue this approach. WebMail Pro, WebMail 3, GopherMail and AlphaMail all use actual folders.
- Other than a resistance to change, are there drawbacks to changing mail clients? Conceptually the old and new mail clients are quite different.
  There are also browser support issues that are prohibiting some people from converting to the new mail clients.
- What percent of people are using old browsers? About 5% of people are using old browsers. While this may not seem like a lot, it is actually a very large number because it includes the entire University community.

- What percent of people are using one of the University's WebMail clients as their primary mail client? Estimates indicate that approximately 70% of the University community is using one of the University's WebMail clients as their primary mail client.

VI). Professor Lopez asked members for ideas for the March 1 agenda. The following ideas were generated:

- Business versus policy implications for providing students with access to purchasing electronic textbooks.
- Accessibility of Information Policy recommendations from the Disabilities Issues Committee.

In light of time, Professor Lopez indicated that an email message would be sent out concerning the status of the March 1 meeting. The meeting will be contingent on whether the appropriate guests can be scheduled.

VII). Hearing no further business, Professor Lopez adjourned the meeting.

Renee Dempsey
University Senate