Uso de Software Livre em Laboratórios de Engenharia

Luciano Rottava da Silva
Dept. de Automação e Sistemas
Univ. Federal de Santa Catarina
CEP 88040-900
e-mail:rottawa@lcmi.ufsc.br

Daniel Martins
Dept. de Engenharia Mecânica
Univ. Federal do Paraná
CEP 81531-990
e-mail:daniel@demec.ufpr.br

Resumo

Este artigo descreve as possibilidades de aplicação e limitações atualmente existentes no uso de software livre em laboratórios de Engenharia.

Abstract

This article describes the currently existing possibilities of application and limitations in the use of free software in Engineering laboratories.

1 Introdução

Em laboratórios de Engenharia muitas atividades e ferramentas são importantes. É impossível ser exaustivo em um assunto tão vasto quanto este. A escolha dos tópicos aqui feita é ilustrativa. É muito provável que o leitor possa citar outros aspectos de *software*, no caso *software* livre, que seriam tão ou mais vitais para um engenheiro, pesquisador ou acadêmico de Engenharia que os que serão aqui elencados. A ênfase dada neste artigo está nos aspectos de intercomunicação, na transmissão de idéias, técnicas ou imagens de uma forma inteligível e aceita pelo grupo de trabalho.

Os tópicos escolhidos aqui formam a necessidade de comunicação interna e externa (*intranet* e *Internet*) por meio eletrônico. Também no aspecto comunicação por meio impresso que passa pela editoração e processamento de textos. A documentação, validação e passagem de modelos e simulações também será abordada. Por último a passagem de idéias, figuras e processos associadas às várias ferramentas *computer-aided* serão brevemente descritas devido à sua importância e à falta de suporte que nesta área o *software* livre disponibiliza.

2 Histórico

O desenvolvimento de *software* livre é um movimento que vem sendo difundido e ampliado desde o final da década de 50, quando pesquisadores do MIT (Massachusetts Institute of Technology) e outras universidades americanas desenvolviam teorias nas áreas de computação e inteligência artificial. Para validar estas pesquisas, escreviam programas e os colocavam à disposição de todos. Neste contexto o termo *free software* foi criado, significando que qualquer pessoa tem a liberdade para usar, estudar, compartilhar e distribuir este tipo de programa.

Foi nessa época também que um dos grandes protagonistas desta história se tornou conhecido, Richard Stallman. Programador e pesquisador, fundou a FSF (Free Software Foundation), órgão que mantém o projeto GNU. Este projeto tem como objetivo principal desenvolver *software* de alta qualidade, sem custo algum para o usuário final. Um dos mais populares editores para UNIX, GNU Emacs, vem deste projeto. Os compiladores gcc e g++ também são outro bom exemplo.

Originou-se uma rede de desenvolvedores distribuídos por todo mundo, corrigindo *bugs* e incluindo novas funcionalidades. Mais do que isto, criou-se uma nova mentalidade no desenvolvimento de programas. Este projeto provou que *software free* não significa necessariamente de baixa qualidade nem cheio de *bugs*. Pelo contrário, como o número de programadores é infinitamente maior que o possuído por *software-houses* famosas, a correção de problemas é mais fácil e rápida.

Nas próximas seções discutiremos mais especificamente as vantagens e desvantagens de alguns *softwares* muito utilizados nos meios acadêmicos, principalmente na área das Engenharias.

3 *Intranet* e *Internet*

A tarefa de prover recursos para uma rede sempre foi um nicho dominado por máquinas UNIX. Isto vem acontecendo com sucesso desde o final da década de 70, quando foram disponibilizadas as primeiras versões deste sistema. Só que nesta época as implementações existentes não eram *free*, muito pelo contrário, apenas grandes universidades e grandes centros de pesquisa tinham condições de ter uma cópia licensiada da AT&T. Mais tarde, a partir da licença da universidade de Berkeley, vários grupos de desenvolvimento foram criados, cada um deles com uma filosofia própria, priorizando na maior parte das vezes algum aspecto que consideravam mais importante. Com isto, muitas versões apareceram, e hoje se dispõe de uma gama de opções bastante grande. Só para citar alguns exemplos da linha BSD, temos FreeBSD, OpenBSD e NetBSD.

Agora com a popularização da Internet, a tendência é que o número de máquinas que passe a utilizar este tipo de sistema aumente muito. O *software* livre tornou viável o uso de redes internas (*intranets*). Isto otimizou a utilização dos recursos tanto de *software* quanto de *hardware*. Além disso a perda de dados foi minimizada (*backups* distribuídos), foram criados meios de comunicação entre os usuários (*e-mail*) e políticas de segurança e acesso.

4 Editores/Processadores de Texto

Em atividades científicas e de Engenharia a comunição de forma escrita é muito importante. Por isso a característica de reusabilidade dos textos é fundamental. Um relatório técnico escrito há mais de cinco anos não pode ser perdido pelo simples fato do processador ou editor de texto não o considerar suficientemente novo ou atualizado para tanto.

Dentro deste enfoque padrões de texto baseados em ASCII como TEX, LATEX, HTML ou simplesmente texto ASCII puro (*plain text*) são os formatos mais perenes e mais compactos. Muitas das comunicações entre membros de um grupo e até mesmo relatório não necessitam de figuras ou diagramas sendo o texto ASCII o mais viável. Aquelas que necessitam devem usar um dos formatos anteriores ou mesmo outros que se baseiem nestas características.

O absurdo vigente atualmente é o envio de formatos binários, muitas vezes carregando consigo vírus, com formatação para um certo editor de uma certa versão específica e com um tamanho muito superior ao necessário.

Certos editores *free* permitem uma utilização quase ilimitada de recursos em textos ASCII. Ferramentas poderosas estão disponíveis como por exemplo: correção ortográfica, tradução de textos on-line, reformatação inteligente associada com o estilo ou formato do texto e interface com outros sistemas externos. Usando-se estes editores de forma integrada com processadores de texto como o LaTeX pode-se obter qualidade tipográfica muito superior à alcançada com editores de texto comerciais amplamente difundidos no mercado.

5 *Softwares* de Simulação

Em Engenharia é fundamental a simulação de sistemas antes e após qualquer projeto ou especificação de produto ou equipamento. Nesta linha, certos *softwares* matemáticos são vitais para o engenheiro ou estudante. A explicação do modelo, sua validação e, talvez mais importante. a passagem deste modelo entre os membros do mesmo grupo que trabalham em simultâneo ou serão sucessores no projeto é vital para minimizar o retrabalho e aumentar a eficiência do trabalho coletivo.

Deste modo, alguns *softwares* são muito importantes e, entre eles destaca-se o Matlab da MathWorks Inc. Seu uso se consagrou e seu formato de trabalho se tornou padrão entre os membros da comunidade acadêmica internacional bem como algumas empresas. Até a versão 4.x existiam similares *freeware* muito bons tais como o Octave e o Scilab.

O Octave é o que mais se parece com o Matlab. Sua estrutura de dados é quase a mesma e muito poucas adaptações são necessárias para se usar programas Matlab anteriores que atendam o formato até a a versão 4.x . Certas limitações são associadas à interface gráfica e ao uso de menus e interações com o usuário. Por outro lado, o Octave apresenta certas vantagens como uma possibilidade de usos de estruturas mais avançadas de programação como orientação para objetos. A compatibilidade com o Matlab 5.x é no entanto bastante limitada já que o próprio *software* comercial possui limitações com suas versões anteriores.

No entanto toda a comparação cai por terra quando se pensa na utilização de interfaces gráficas interativas tais como no pacote (*toolbox*) Simulink integrado ao Matlab. Neste pacote, os sistemas são apresentados graficamente com diagramas de bloco facilmente manipuláveis, fazendo com que o usuário se concentre apenas no trabalho que deseja realizar ao invés de se preocupar com detalhes inerentes à ferramenta. O Scilab apresenta uma interface similar chamada Scicos, mas que fica ainda muito aquém das capacidades do Simulink.

Resumindo com relação aos *softwares* de simulação, relativamente pouco trabalho precisa ser feito para que estas ferramentas estejam disponíveis e sejam utilizadas em escritórios de Engenharia ou ambientes acadêmicos de forma mais sistemática. Mas, infelizemente, a prática atual ainda passa pela aquisição de *softwares* comerciais em casos em que a interatividade seja absolutamente imprescindível, mesmo que os preços sejam relativamente altos. A tendência, no entanto, é que com a popularização do *software* livre, os grupos que dão suporte a estes produtos contem com mais apoio e passem a desenvolver ferramentas que apresentem características similares às encontradas na versões comerciais.

6 Ferramentas *CASE* para Engenharia

Aqui estão, por enquanto, as maiores deficiências dos sistemas livres em relação ao sistemas comerciais. Isto devido principalmente ao desinteresse demonstrado por parte das empresas

desenvolvedoras deste tipo de *software*. O aspecto custo é proibitivo. O preço cobrado pelos programas por parte dos poucos representantes sempre foi significativamente maior se comparado à de outras plataformas. O motivo alegado é que sistemas UNIX e seus clones (dentre os quais está o Linux) são plataformas de alto desempenho, utilizadas apenas em missões críticas e sistemas computacionais avançados. Isto vale também para uma implementação *free* de um UNIX rodando em um micro 486 DX (Linux) ou em uma Harvard-Graphics de última geração.

Esta atitude dificulta seriamente a utilização de sistemas UNIX como plataforma de pesquisa e ensino, pois um centro de pesquisa ou laboratório de uma universidade dificilmente terá recursos para investir num produto destes. Opta-se, então, utilizar a mesma ferramenta de projeto em um sistema reconhecidamente inferior, do ponto de vista de estabilidade, confiabilidade e custo.

Um exemplo disto são os pacotes para projetos, como alguns CADs para engenharia civil e mecânica e simuladores de circuitos elétricos e eletrônicos para a engenharia elétrica/eletrônica.

No caso de sistemas CAD que trabalham com gráficos, a problemática maior talvez seja ainda o padrão de formato das figuras. Apesar de existir uma norma ISO definindo um padrão (chamado STEP), este como tantos outros, ainda não vingou em virtude de interesses comerciais dos vários fabricantes de CAD. Um projeto de CAD livre existe mas ainda engatinha e não viabiliza ainda seu uso, mesmo que de forma experimental, nem mesmo na academia.

7 Conclusões

Muito se pode fazer em Engenharia, na academia ou na indústria, com *softwares* livres. Muitos projetos estão em andamento e poderão definir melhor o futuro nesta área em breve.

Certas áreas como por exemplo a edição já tem um nível muito desenvolvido e são completamente indiferentes à qualquer "inovação" feita por *software-houses*.

Pode-se notar que a maior parte dos problemas passa pela existência ou não de normas de comunicação ou padronização de formatos. Muito se deve fazer ainda neste sentido pois os padrões não podem ser proprietários e devem ser eficientes. Somente grupos independentes de interesses comerciais poderiam ter idoneidade suficiente para isto.

Referências

[1] Roger S. Flugel. Biomedical research and linux. *Linux Journal*, pages 18–20, april 1998.

[2] Michel Goossens, Frank Mittelback, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, 1994.

[3] Przemek Klosowski, Nick Maliszewskyj, and Bud Dickerson. Linux in a scientific laboratory. *Linux Journal*, pages 12–20, july 1998.

[4] Leslie Lamport. *LaTeX: A document preparation system*. Addison-Wesley, 1996.

[5] Wolf-Rainer Novender. Sceptre: Simulation of nonlinear electrical circuits. *Linux Journal*, pages 62–66, july 1999.

[6] Doc Searls. The linux position. *Linux Journal*, pages 36,37, july 1999.

[7] Andrew S. Tanenbaum. *Computer Networks*. Prentice-Hall, third edition, 1996.