



Anais do XXXIV COBENGE. Passo Fundo: Ed. Universidade de Passo Fundo, Setembro de 2006.
ISBN 85-7515-371-4

UMA METODOLOGIA GERMANO-BRASILEIRA PARA O ENSINO DE ENGENHARIA DE SOFTWARE PARA ALUNOS DE ENGENHARIA ELÉTRICA

Vicente Ferreira de Lucena Jr. - vicente@ufam.edu.br

UFAM/CETELI e CEFET-AM

Av. Gal. Rodrigo Otávio, 3000, Ceteli
69077-000 – Manaus – AM – Brasil

Alysson Brito - alysson.brito@siemens.com

Siemens SHC

Av. Solimões, SN, Distrito Industrial
69075-155 – Manaus – AM - Brasil

Nasser Jazdi - nasser.jazdi@ias.uni-stuttgart.de

IAS – Universität Stuttgart

Pfaffenwaldring, 47

70550 – Stuttgart – Alemanha

Resumo: *Este artigo descreve uma metodologia, com suas atividades e procedimentos, voltada ao ensino da disciplina engenharia de software para alunos da graduação do curso de engenharia elétrica da Universidade Federal do Amazonas – UFAM – na cidade de Manaus. Apesar de ser cada vez mais importante para a formação do profissional de engenharia elétrica nos tempos de hoje, a disciplina engenharia de software sempre foi pouco procurada pelos alunos deste curso na UFAM. Neste artigo, descreve-se como a introdução de um conjunto de ferramentas associadas a uma competição com robôs ajudou a aumentar o interesse dos alunos pelo tema e, principalmente, ajudou a manter até o fim o interesse dos alunos matriculados na disciplina, resultando num índice de desistência significativamente baixo. Um segundo efeito interessante da experiência aqui relatada foi a aproximação entre os alunos de ciência da computação e de engenharia elétrica, fato este incomum em nossa universidade. Este trabalho é um dos resultados do programa Unibral da CAPES, que financia um projeto de cooperação acadêmica e técnico-científica entre o Instituto de Automação Industrial e Engenharia de Software (IAS) da Universidade de Stuttgart na Alemanha e a Faculdade de Tecnologia da Universidade Federal do Amazonas.*

Palavras-chave: *Ensino de Engenharia de Software, Laboratório de Engenharia de Software, Uso de Robôs para o Ensino de Engenharia, Intercâmbio Acadêmico, Cooperação Brasil-Alemanha.*

INTRODUÇÃO

Encontramos na literatura uma série de relatos sobre as dificuldades em ensinar engenharia de software e disciplinas correlatas para cursos de engenharia em geral (BLAKE e CORNETT (2002), COWLING (2004), MCKIM e ELLIS (2004)). Historicamente, esta disciplina está fortemente ligada aos cursos de ciência da computação. Observamos, ainda, que os alunos que freqüentam cursos de engenharia têm formação acadêmica e objetivos profissionais diferentes dos alunos que freqüentam os cursos relacionados à informática e à ciência da computação. Nos tempos modernos, com equipes multi-profissionais sendo alocadas para resolver problemas relacionados ao desenvolvimento de software, surge o desafio de ensinar uma disciplina tão relevante para a indústria para estes dois tipos distintos de alunos.

Na UFAM, a situação não difere muito da abordagem clássica de outras universidades. Os cursos de engenharia elétrica e de ciência da computação estão sediados em faculdades diferentes. O primeiro, inserido na Faculdade de Tecnologia, tem uma formação mínima de cinco anos e oferece habilitações em Automação Industrial, Eletricidade e Telecomunicações, enquanto o segundo segue o currículo amplamente discutido pelos especialistas da ACM, formando profissionais após quatro anos de estudo. Ambos possuem um elenco de disciplinas obrigatórias para os alunos inscritos nos mesmos, assim como uma série de disciplinas optativas, às quais alunos de cursos diferentes podem freqüentar indistintamente.

O parque industrial de Manaus é composto basicamente pela indústria manufatureira e pelo setor de serviços. Estatisticamente, nos últimos anos, tanto os alunos graduados em engenharia elétrica como em ciência da computação não têm dificuldades de encontrar emprego. No entanto, com o recente advento de uma lei de incentivo ao desenvolvimento de pesquisas tecnológicas ligadas à informática, notou-se uma maior demanda por profissionais com conhecimentos sólidos em engenharia elétrica e que tenham habilidades consolidadas em desenvolvimento de software. Associado a este fato, observa-se que os dispositivos fabricados e desenvolvidos pela indústria localizada em Manaus seguem a tendência tecnológica de possuir cada vez mais software embarcado, o que demanda profissionais com habilidades nas duas ciências citadas. Não raramente, empresas locais contratam profissionais com este perfil, oriundos de cidades como São Paulo ou Rio de Janeiro, distantes cerca de 3500 km.

Apesar da necessidade da formação em engenharia de software por parte dos estudantes de engenharia elétrica ser claramente discutida na sociedade local, não foi possível convencê-los a participar das disciplinas oferecidas no curso de ciência da computação. Aspectos inerentes à formação básica destes dois tipos de profissionais e ainda aspectos sociais amplamente discutidos em (PARNAS (1999)) colaboraram decisivamente para esta rejeição. Uma das colocações mais freqüentes para justificar a falta de interesse em freqüentar engenharia de software diz respeito ao aspecto extremamente teórico da disciplina. Na verdade, os estudantes cometem um equívoco, ao entender que a grande quantidade de documentação necessária para o bom desenvolvimento de um projeto em engenharia de software é sinônimo de disciplina teórica. Os alunos preferem programar e ver algo funcional, a documentar formalmente o desenvolvimento de suas aplicações.

Outra colocação que não pode ser menosprezada diz respeito a certa rivalidade existente entre os alunos dos dois cursos. O número de candidatos por vaga do curso de ciência da computação na UFAM é muito maior que o de engenharia elétrica. Isto leva a crer que um curso é superior ao outro e influencia o comportamento dos alunos, dificultando a convivência no dia-a-dia.

Com o objetivo de aproximar efetivamente estes dois grupos de alunos, desde o início do projeto Unibral em 2002, decidiu-se que os alunos participantes deveriam estar vinculados aos dois cursos não permitindo uma nítida separação entre os mesmos. Este projeto contempla a troca de estudantes entre as universidades envolvidas. A cada ano, três brasileiros são enviados para a Alemanha e lá freqüentam aulas pelo período de 11 meses, enquanto 4 a 6 alemães vêm ao Brasil por um período de até seis meses. O projeto também prevê a troca de pesquisadores e, principalmente, a troca de experiências acadêmicas com a implantação de disciplinas relacionadas à engenharia de software e à automação industrial, pontos fortes do Instituto de Automação Industrial e de Engenharia de Software na Universidade de Stuttgart. Tendo o Unibral como justificativa, decidiu-se ainda implantar inicialmente a disciplina de engenharia de software para tempo-real. Esta seria ministrada no curso de engenharia elétrica e seria aberta para estudantes de ciência da computação. Desta forma, seria reproduzida a mesma disciplina oferecida em Stuttgart e sob condições similares, ou seja, ministrada na engenharia por um professor de engenharia e com participação livre de outros alunos da Universidade.

Nestas circunstâncias, tinha-se ainda a necessidade de tornar a disciplina atrativa tanto para engenheiros como para cientistas da computação. Precisava-se, então, transformar o laboratório em uma atividade interessante para estes alunos que historicamente não demonstravam interesse em aprender engenharia de software. Muitos educadores indicam que o uso de artefatos robóticos é adequado para solucionar problemas como os acima citados (MARTIN (2001)). Ressalta-se que o objetivo nunca foi construir um robô para depois programá-lo como em (VERNER e AHLGREN (2004)), mas sim, partir de uma plataforma já existente e desenvolver um sistema de software para resolver um determinado problema usando o robô. Em um primeiro momento, adaptou-se os experimentos utilizados em Stuttgart com robôs fabricados pela empresa Fischertechnik (Fischertechnik Web) para robôs da LEGO Mindstorms (FERRARI et al. (2003)). Finalmente, migrou-se para a experiência descrita neste artigo que utiliza robôs autônomos mais robustos e uma plataforma de programação mais próxima da realidade industrial.

Apesar de ser um laboratório com fins eminentemente didáticos, pretendia-se que os alunos experimentassem os mesmos problemas que são encontrados na vida profissional real. Adicionalmente, tudo isto deveria ocorrer no período de um semestre letivo (15 semanas de aula na UFAM). Tanto a parte teórica como o laboratório prático, deveriam ser oferecidos como uma única disciplina optativa para alunos de engenharia elétrica. As considerações apresentadas em (DAWSON (2000), SHAFER (2002) e GEHRKE et al. (2002)) foram realmente úteis e ajudaram a criar uma atmosfera de trabalho próximo da realidade. Considerou-se ainda, adotar procedimentos semelhantes aos apresentados em (ELLIS e HISLOP (2004)) a fim de atender às expectativas de estudantes que consideram engenharia de software uma disciplina não muito atrativa.

Os alunos compuseram times de trabalho que tinham por objetivo não somente escrever todos os documentos inerentes da produção do software, mas ainda desenvolver um algoritmo para os robôs que fosse vitorioso em uma competição. A experiência do laboratório em Stuttgart considera o importante fator da formação de times multi-culturais (BORCHERS (2003)) inclusive com alunos com línguas maternas diferentes de inglês ou alemão. Este não é o caso na UFAM, uma vez que são poucos os alunos estrangeiros inscritos. Entretanto, foi decidido que todos os documentos apresentados tanto pelos alunos como pelo professor seriam escritos em inglês. Desta forma, aproximou-se mais este trabalho da realidade de mercado.

ORGANIZAÇÃO DO CURSO

O curso de laboratório de engenharia de software vem sendo realizado todo semestre de verão na Universidade de Stuttgart, especificamente no Instituto de Automação Industrial e Engenharia de Software (IAS), desde 1996 (Portal IAS). Este laboratório é baseado em um sistema com robôs desenvolvidos pela Universidade Técnica de Munique e utiliza módulos fabricados pela empresa Fischertechnik. Desde o início do projeto Unibral pretendia-se que um laboratório nas mesmas condições fosse implementado em Manaus.

Em uma etapa prévia à oferta deste curso em Manaus, três estudantes da UFAM participaram do intercâmbio acadêmico em Stuttgart, no ano de 2004. Esses estudantes participaram ativamente das aulas teóricas e do laboratório de engenharia de software juntamente com os estudantes alemães, o que permitiu um profundo conhecimento das particularidades dos cursos. Uma segunda tarefa realizada por estes alunos, foi construir uma nova estrutura de hardware e software, novos robôs e novo conjunto de ferramentas, para serem usados por ambas as instituições no ano de 2005. Esse novo sistema deveria substituir o então utilizado na Universidade de Stuttgart. Entre as características melhoradas desses robôs, destaca-se o fato deles serem mais robustos e do sistema de comunicação utilizar ondas de rádio e não mais infra-vermelho, uma das fontes de problemas anteriormente existentes.

Uma característica fundamental da abordagem realizada na Alemanha é que a parte teórica é sempre realizada em um semestre anterior à oferta do curso de laboratório. Além disso, nada garante que os alunos que tenham freqüentado a aula teórica freqüentem as aulas de laboratório. A estrutura adotada para o curso teórico e para o laboratório implementada na UFAM modificou este procedimento. Em Manaus, os dois cursos são oferecidos em um mesmo semestre, sendo obrigatória a participação dos alunos nos dois eventos simultaneamente. Isso tende a tornar as aulas muito mais interessantes para os alunos, porque (1) logo após as palestras, eles sabem que irão precisar de tais informações; (2) a prática solidifica os conhecimentos e também levanta dúvidas pertinentes. Outro aspecto interessante dessa abordagem é que a ligação contínua teoria-prática evita a criação de lacunas de conhecimento deixada pela abordagem monolítica, de um semestre teórico e outro prático somente no próximo semestre, reduzindo a curva de aprendizado de um ano para seis meses.

Requisitos do Laboratório

O objetivo do laboratório é construir um sistema (algoritmo) para controle de um robô que deve partir de um ponto pré-definido e se conduzir de forma autônoma em uma pista com obstáculos desconhecidos. Toda a estrutura do sistema foi disponibilizada para os alunos. Isto inclui a mecânica e o sistema de controle do robô, sistema de comunicação, bibliotecas de acesso às primitivas de controle do robô e até mesmo uma biblioteca gráfica. O desafio era desenvolver uma estratégia para guiar o robô no menor tempo possível com as poucas informações disponíveis.

Foi também requisitado aos alunos que seguissem um modelo de processo pré-definido. Este modelo é semelhante ao adotado pelo IAS no desenvolvimento de software por seus alunos. São contempladas as seguintes fases (SWD): Especificação do Trabalho, Projeto, Implementação e Integração e Entrega. Além disso, são gerados documentos de controle de qualidade (QA) em cada fase. A Tabela 1 sumariza os objetivos e os documentos resultantes.

Logo no início do curso, os alunos receberam um documento com os requisitos do usuário. Este documento não é completo, deixando dúvidas propositais para que os alunos venham a resolvê-las ao longo do trabalho. Todos os templates dos documentos necessários são fornecidos com exemplos de preenchimento e os alunos são orientados sobre como utilizá-los durante as atividades do laboratório. As fases e milestones também foram pré-

definidas, cabendo aos alunos adaptar e executar o projeto dentro das especificações impostas pelo processo. Este é mais um ponto forte do laboratório: os alunos são ensinados a entender e utilizar um processo de desenvolvimento particular, antecipando situações que eles irão viver na sua vida profissional no futuro.

Tabela 1 – Descrição das fases de desenvolvimento do laboratório.

Fase	Objetivo	Documento Gerado
<i>SWD 1 – Especificação do Trabalho</i>		
SWD 1.1	Determinar completamente os requisitos do sistema em acordo com o cliente	Especificação dos Requisitos do Sistema
SWD 1.2	Realizar uma análise detalhada dos requisitos apresentados	Modelo do Sistema de Software
QA 1.2	Avaliar os produtos desenvolvidos nesta fase	Revisão da Especificação
<i>SWD 2 – Projeto</i>		
SWD 2.1	Projetar a arquitetura do sistema de software a ser implementada	Arquitetura do Sistema de Software
SWD 2.2	Projetar e especificar completamente os diversos componentes de software a serem implementados	Especificação dos Componentes de Software
QA 2.1	Criar uma especificação de testes compatível com os componentes gerados e com o sistema como um todo	Especificação de Testes
QA 1.3	Avaliar os produtos desenvolvidos nesta fase	Revisão de Projeto
<i>SWD 3 – Implementação</i>		
SWD 3.1	Implementar os diversos componentes de software	Programas Fontes e Documentação Associada
QA 1.4	Avaliar os produtos desenvolvidos nesta fase	Revisão da Implementação
<i>SWD 4 – Integração e Entrega</i>		
SWD 4.1	Integrar os componentes e instalar o sistema completo	Sistema de Software Completo Instalado e Funcional
QA 2.2	Testar o sistema entregue em sua completitude	Protocolo de Testes do Sistema
QA 1.5	Avaliar os produtos desenvolvidos nesta fase	Revisão da Integração e Entrega

Sistemas de Hardware e Software Utilizados

O robô utilizado chamado de SOPIAS2 (Figura 1) é uma peça de hardware composta por um microprocessador, dois motores simetricamente alocados em um chassis de alumínio, 2 conjuntos de baterias e 12 sensores de toque. Eles executam movimentos retilíneos (frente e ré) e de giro (horário e anti-horário), um de cada vez e possuem ainda uma unidade de comunicação via rádio frequência.



Figura 1 – Fotografia do robô utilizado (SOPIAS2).

O software embarcado do robô é bem simples contemplando somente as rotinas de monitoramento dos sensores e de controle do robô propriamente, além de rotinas de comunicação com um computador de mesa. O software desenvolvido pelos alunos é executado em PCs clientes. Estes, são conectados via rede a um servidor Linux rodando um programa que funciona como um gateway, que é responsável por estabelecer a comunicação com os robôs. A Figura 2 representa um overview do sistema utilizado.

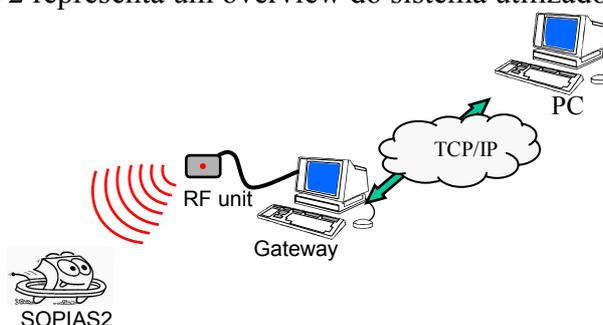


Figura 2 – Arquitetura do Sistema.

É disponibilizado para os alunos um conjunto de APIs com as funcionalidades necessárias para que eles acessem o robô. Isto faz com que o gateway robô-servidor e todo o sistema de comunicação seja transparente para os estudantes. Existe também um conjunto de APIs com funcionalidades básicas de controle do robô. A documentação descrevendo estas APIs também contém erros propositalmente simulando situações comuns da indústria.

Estas APIs estão escritas nas linguagens de programação C++ e Java, ficando a critério de cada grupo a escolha da plataforma desejada. Embora não seja obrigatório, sugere-se o uso de uma IDE comercial, o que facilita a programação e gerenciamento das linhas de código geradas.

A competição dos robôs acontece numa arena com uma área de 5x8 m. Os obstáculos são construídos a partir de blocos do tamanho de uma folha A4 numa dada configuração totalmente desconhecida dos alunos, obedecendo apenas ao critério que sempre será possível alcançar o gol. Em outras palavras, não é permitido bloquear totalmente a passagem dos robôs, devendo-se deixar no mínimo um espaço equivalente a uma vez e meia o valor de seu diâmetro (40 cm). Como o uso da arena foi limitado aos últimos testes, foi necessário disponibilizar um programa que simulava todo o sistema (robôs, obstáculos e comunicação) e que servia para as primeiras etapas da construção do algoritmo de controle. Este programa utiliza as mesmas APIs utilizadas pelo sistema real. Com este recurso, as diversas equipes podem elaborar seus algoritmos utilizando computadores comuns em qualquer ambiente de programação. A figura 3 mostra um screenshot da interface do simulador. No alto da figura estão localizados os dois robôs que competem (círculos), os retângulos são os obstáculos que devem ser contornados. Os robôs devem partir dos cantos superiores, um de cada lado (na figura com uma marca +), e chegar no centro da parte inferior da arena. Quem chegar primeiro ao gol, será o vencedor.

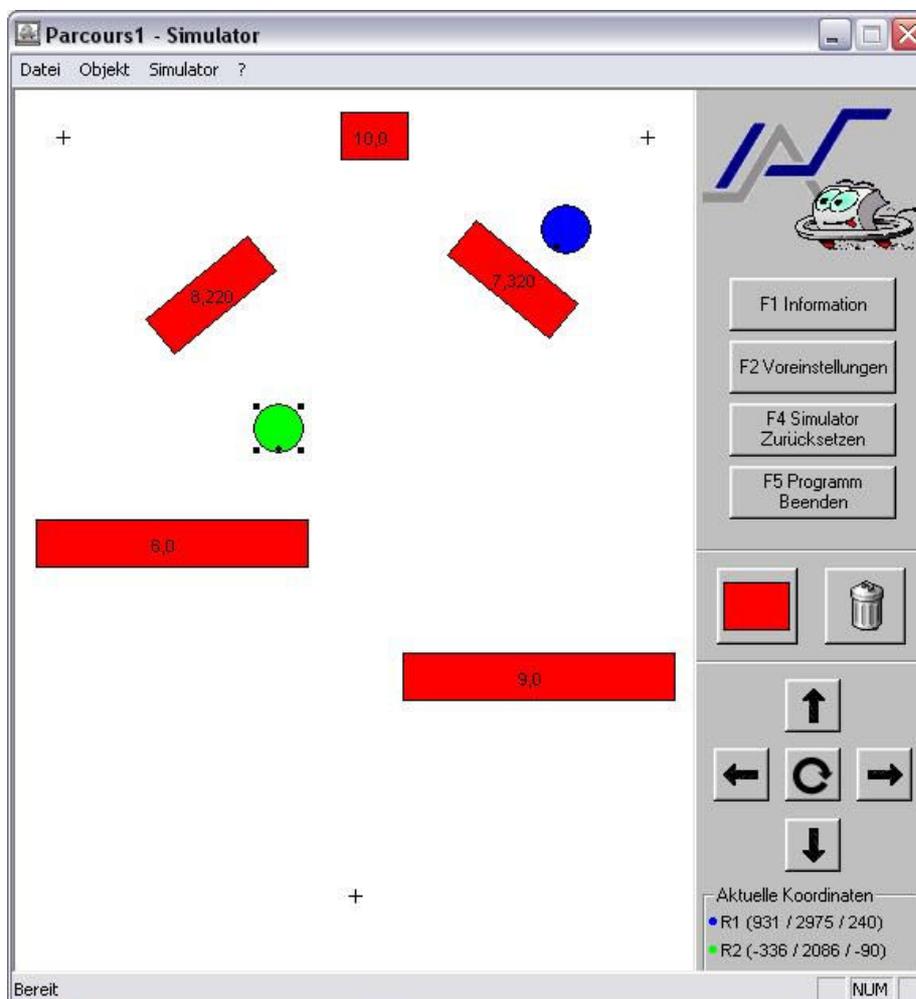


Figura 3 – Interface do Simulador do Sistema.

Alunos Tutores

Outro importante papel foi o dos alunos tutores. Na universidade alemã, um pesquisador assistente é responsável por cada equipe. Este pesquisador, normalmente, é um doutorando

com pesquisas relacionadas à engenharia de software. Entre outras atribuições ele é responsável por cobrar a entrega de documentos, acompanhar o andamento do projeto, agendar reuniões sempre que necessário, e tirar dúvidas teóricas e de implantação. Por serem efetivamente funcionários da universidade alemã, esses assistentes estão sempre à disposição para auxiliar os alunos.

Esta não é a realidade em Manaus. Não existe o cargo de assistente de ensino ou de pesquisa. Na implantação deste laboratório na UFAM em 2005, os alunos que desenvolveram o sistema na Alemanha, e que também participaram da experiência do laboratório no ano de 2004, foram convidados a exercerem este papel. Entre outras atividades, eles assumiram algumas aulas expositivas, resumindo aspectos práticos das fases de desenvolvimento.

Os tutores exerciam também o papel de clientes do projeto. Desta forma, eles definiam os seus requisitos, explicando o projeto sempre de uma forma não exaustiva para simular um ambiente real. O que é interessante ressaltar, é que alguns destes tutores não aprofundavam as explicações sobre os requisitos de propósito, fato que ocorre com frequência na indústria. Em outros casos, algumas informações só eram fornecidas na segunda ou terceira entrevista, requisitada sempre pelos alunos, quando estes julgavam necessário.

Os tutores também foram responsáveis por implantar um portal eletrônico de comunicação entre os alunos, os tutores propriamente e o professor responsável pela disciplina. Adotou-se a ferramenta denominada TeleEduc (Portal TeleEduc). O IAS também possui tal infra-estrutura de portal na Alemanha (Portal Lab Course IAS), porém o programa implantado para e-learning pela UFAM contém recursos ainda mais avançados, o que facilitou o uso do portal.

ATIVIDADES DO CURSO

O curso foi dividido em uma parte teórica com 60 horas-aula ministradas em no máximo 15 semanas. O laboratório continha oficialmente apenas 30 horas-aula. Foi acertado também, que ao final do semestre letivo, seria realizada a corrida dos robôs em forma de competição com uma equipe sendo consagrada como a que gerou o melhor algoritmo de controle do robô.

Aulas Teóricas

As aulas teóricas seguiram modelo clássico de ensino de engenharia de software. Por exemplo, uma das referências bibliográficas adotadas é o já famoso livro de Ian Sommerville (SOMMERVILLE (2004)). O material didático usado na Universidade de Stuttgart pelo Prof. Göhner foi gentilmente cedido e todos os slides e exercícios apresentados no semestre seguiram o roteiro do IAS (Portal Lab Course IAS). Isto forçou os alunos a lerem o material todo escrito na língua inglesa. As aulas expositivas, no entanto, continuaram sendo faladas em português. Foram aceitos alunos cursando o sexto semestre de engenharia elétrica ou o quinto de ciência da computação. Foi recomendado que os alunos possuíssem conhecimentos em C++ ou Java, mas não foi exigida comprovação destes conhecimentos através de disciplinas previamente cursadas.

Aulas Práticas

Os alunos inscritos foram divididos em grupos de trabalho contendo sete integrantes cada, mesmo número utilizado no IAS e recomendado na literatura. Ao todo, foram formados cinco grupos. A montagem das equipes foi deixada a cargo dos alunos que se organizaram livremente. A única recomendação foi que pelo menos três possuíssem algum conhecimento

em programação. Foi explicado que o resultado final do laboratório seria julgado pelo trabalho desenvolvido pela equipe, e não individualmente por cada componente. Aconteceu que a montagem das equipes seguiu um critério meramente pessoal, com amigos próximos participando da mesma equipe. Coincidentemente, sete dos alunos inscritos eram de ciência da computação e eles resolveram montar um grupo no qual nenhum aluno de engenharia participaria.

Tabela 2 – Atividades relacionadas ao laboratório.

Semana	IN	ET	PR	IM	IE	Atividades
1						Seminário de introdução. Introdução em engenharia de requisitos.
2						Introdução à gerência de tempo, divisão de grupo e discussão dos requisitos. Planejamento do projeto, e especificação de requisitos. Documentos: Especificação dos Requisitos do Sistema
3						Introdução à análise do sistema. Análise OO, definir estratégias do robô, planejar testes. Documentos: Modelo do Sistema de Software Planejamento de Projeto
4						Introdução ao projeto de sistema. Projeto, planejar a integração e preparar os casos de testes. Documentos: Especificação dos Componentes de Software Arquitetura do Sistema de Software
5						Introdução à implementação e ao teste. Documentos: Especificação de Testes Protocolo de Testes
6						Implementação e teste dos módulos.
7						Documentos: Programas Fontes e Documentação Associada Protocolo de Testes do Sistema
8						Demonstração do protótipo do sistema.
9						Teste do sistema e revisão.
10						Documentos: Revisão da Integração e Entrega
11						Corrida dos Robôs
12						Relatórios finais e festa de confraternização

As atividades de laboratório foram detalhadas em um cronograma que foi seguido o mais fielmente possível. Estas atividades com os documentos exigidos e sua distribuição no tempo estão apresentadas na Tabela 2. Os tutores ficaram responsáveis de cobrar a execução de cada atividade e dar suporte para que as mesmas ocorressem a contento. A correção formal e a atribuição de notas continuaram a cargo do professor.

Os papéis dos membros das equipes foram definidos previamente e longamente explicados para os estudantes na segunda semana de trabalho. Cada equipe decidiu livremente quem exerceria cada papel: Líder de Projeto, Controle de Qualidade, Gerenciador de

Configuração e Desenvolvedores. Quatro alunos desempenharam a função de desenvolvedores, cabendo cada uma das tarefas restantes a um dos outros componentes do grupo.

Competição de Robôs

A competição foi dividida em duas etapas. Na primeira, todas as equipes competiam contra o relógio em uma configuração de obstáculos constante. Na segunda etapa, elas competiam entre si na mesma forma que se organizam torneios de futebol. Foram atribuídas pontuações para os robôs mais rápidos. Os grupos deviam ainda fazer uma apresentação oral sobre o algoritmo de controle adotado. O evento foi aberto ao público em geral e realizado no Hall de entrada do Instituto de Ciências Humanas e Letras, um local com maior fluxo de pessoas.

RESULTADOS

Os resultados alcançados foram além das expectativas. Os alunos inscritos no curso mantiveram-se muito motivados até o último dia, quando ocorreu a competição dos robôs, talvez o maior fator de motivação existente. Houve um índice de abandono de apenas 12%. Em geral, constatamos um índice de mais de 35% em cursos regulares da engenharia elétrica na UFAM. Os alunos executaram as diversas atividades teóricas e práticas com empenho e nível técnico muito bom. Notou-se, ainda, o interesse de alunos de engenharia em saber quando a disciplina voltaria a ser oferecida e a quantidade de vagas disponíveis, o que faz acreditar que a procura será maior nos semestres seguintes. Uma análise dos resultados nos leva a considerar uma série de fatores relevantes que podem ser contribuições para trabalhos futuros:

- O uso de diversos documentos escritos em inglês gerou alguma confusão no início das atividades, havendo misturas de português e inglês nos textos produzidos pelos alunos. A iniciativa não foi bem recebida inicialmente, mas ajudou os alunos a entenderem a necessidade de se escrever e ler bem em inglês para trabalhar com engenharia de software.
- Os alunos nunca haviam seguido, sendo que alguns desconheciam totalmente, um modelo de processo. Explicar que seus documentos deveriam seguir necessariamente o modelo apresentado demorou mais do que o esperado. Além disso, esta foi a primeira disciplina do curso de engenharia elétrica a exigir um número tão grande de documentos.
- Os participantes não tinham muita familiaridade com aspectos mais avançados de programação. Apesar da estrutura do laboratório tentar tornar transparente certos aspectos como comunicação pela porta serial, criação de sockets, comunicação RF, etc, os alunos precisavam trabalhar com threads, APIS e IDEs comerciais. Até mesmo alunos de ciência da computação tiveram dificuldade no uso das ferramentas, uma vez que só utilizam softwares livres nas outras disciplinas.
- As atividades práticas de laboratório e teóricas de documentação exigiram mais tempo do que o previsto. Os alunos necessitaram de várias horas de reuniões fora do horário normal da disciplina para definirem tarefas e desenvolverem o trabalho a contento. A realidade da UFAM é que vários alunos trabalham em paralelo aos seus estudos, e, portanto têm pouco tempo fora de sala para se reunirem. Isso gerou uma carga de trabalho maior no final do projeto, quando os alunos ficaram nos últimos finais de semana dentro da instituição executando as tarefas da disciplina.
- Apesar do trabalho em equipe não ser novidade para nossos alunos, esta foi a primeira vez que eles interagiram de forma realmente intensa e com necessidade de participação de

todos integrantes. Além disso, as tarefas estavam bem definidas demandando um trabalho individual intenso. Isso exigiu intervenções dos tutores, orientando os alunos a não exercerem mais de uma função, não misturar papéis, e cumprir o que foi acordado nas reuniões das equipes.

- Diferentemente do que ocorre na Alemanha, os tutores não eram funcionários da universidade e, portanto, não tinham disponibilidade integral e muitos problemas tiveram que ser resolvidos não presencialmente, mas via portal do projeto.

CONCLUSÕES

Esta é a única disciplina relacionada à engenharia de software oferecida pela engenharia elétrica. Apesar disso, em anos anteriores a procura não passava de 10 alunos inscritos. Foi então divulgado que neste semestre a disciplina seria oferecida nos moldes da cooperação com a Alemanha e que incluiria atividades práticas em laboratório. O número de alunos inscritos chegou a 35, superando as expectativas iniciais. Ao final da disciplina, vários alunos já estavam procurando informações a respeito de como se inscrever no curso no ano de 2006. Isso supõe uma mudança de atitude dos alunos de engenharia elétrica frente ao ensino de engenharia de software.

As dificuldades acadêmicas encontradas pelos alunos foram seguramente superiores às encontradas em outras disciplinas e, mesmo assim, o índice de abandono foi bem menor que a média. Apenas 5 alunos inscritos abandonaram a disciplina antes do término do período. Conclui-se, então, que a utilização dos robôs como objetivo final do laboratório foi um fator decisivo para o bom aproveitamento da disciplina.

A composição dos grupos de trabalho não deve ser deixada a cargo dos alunos sozinhos. Ao definir previamente quem vai trabalhar em cada grupo, aproxima-se da realidade da indústria, uma vez que raramente escolhe-se um emprego a partir de critérios como amizade pessoal ou conhecimento prévio dos colegas. Fosse este o procedimento adotado no início deste projeto, não teríamos tido uma equipe composta somente por alunos de ciência da computação. Apesar disso, notou-se positivamente uma maior aproximação entre os alunos dos cursos de engenharia elétrica e de ciência da computação.

Cerca de 200 pessoas acompanharam atentamente a competição que durou aproximadamente duas horas. Este foi o primeiro evento da engenharia elétrica aberto ao público em geral, promovido por uma disciplina regular do curso na UFAM, o que gerou matérias em jornais da universidade e também do Estado do Amazonas. Outro aspecto interessante foi apresentar o curso de engenharia elétrica à comunidade local, uma forma de mostrar à sociedade que o investimento feito está produzindo resultados satisfatórios.

Agradecimentos

Nossos agradecimentos se dirigem primeiramente à Fundação CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, órgão do Ministério da Educação do Governo do Brasil e ao DAAD – Deutscher Akademischer Austausch Dienst, órgão de cooperação acadêmica alemão, que deram suporte financeiro ao projeto Unibral entre Manaus e Stuttgart. Sem o apoio recebido não seria possível realizarmos este trabalho. Gostaríamos de fazer um especial agradecimento ao Prof. Dr.-Ing Peter Göhner, diretor do Institut für Automatisierungs- und Softwaretechnik da Universität Stuttgart, por sua gentil cooperação, cedendo suas aulas e seus assistentes, e por sua disponibilidade de tempo apesar de todos seus afazeres, sem o que não teríamos tido sucesso no intercâmbio aqui relatado. Finalmente, agradecemos todos os alunos, brasileiros e alemães, que participaram deste curso e cujo comportamento ajudou a concluirmos nossas experiências com sucesso.

REFERÊNCIAS BIBLIOGRÁFICAS

- BLAKE, M.B.; CORNETT, T. Teaching an Object-Oriented Software Development Lifecycle in Undergraduate Software Engineering Education. In 15th CSEE&T **Anais**, 2002.
- BORCHERS, G. The Software Engineering Impacts of Cultural Factors on Multi-cultural Software Development Teams. In 25th ICSE, **Anais**, 2003.
- COWLING, A.J. The CrossoverProject as an Introduction to Software Engineering. In 17th CSEE&T, **Anais**, 2004.
- DAWSON, R. Twenty Dirty Tricks to Train Software Engineers. In 22nd INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), **Anais**, Limerik, Irland, 2000.
- ELLIS, H.J.C.; HISLOP, G.W. Techniques for Providing Software Engineering Education to Working Professional. In 34th ASEE/IEEE FIE, **Anais**, Boston, USA, 2004.
- FERRARI, M. et. al., **LEGO Mindstorms Masterpieces: Building Advanced Robots**, Syngress 2003.
- Fischertechnik Building Blocks for Life...: <http://www.fischertechnik.de/>
- GEHRKE, M. et. al. Reporting about Industrial Strength Software Engineering Courses for Undergraduates. In 24th ICSE, **Anais**, Orlando, Florida, USA, 2002.
- IAS - Institute of Industrial Automation and Software Engineering: <http://www.ias.uni-stuttgart.de/>
- MARTIN, F. **Robotic Explorations, A Hands-on Introduction to Engineering**. Prentice Hall, Upper Saddle River, USA, 2001.
- MCKIM JR., J.C.; ELLIS, H.J.C. Using a Multiple Term Project to Teach Object Oriented Programming and Design. In 17th CSEE&T '04, **Anais**, 2004.
- PARNAS, D.L. Software Engineering Programs Are Not Computer Science Programs. **IEEE Software Magazine**, November / December 1999.
- SHAFER, D. Practical Graduate Software Engineering Projects: Can Something Useful be Built in One Semester? In 32nd ASEE/IEEE FRONTIERS IN EDUCATION CONFERENCE, **Anais**, Boston, USA, 2002.
- SOMMERVILLE, I. **Software Engineering**. 7th Edition, Addison Wesley, 2004.
- VERNER, I.M.; AHLGREN, D.J. Robot Contest as Laboratory for experimental Engineering Education. **ACM Journal on Education Resources in Computing**, June 2004.

A GERMAN-BRASILIAN METHODOLOGY FOR TEACHING SOFTWARE ENGINEERING FOR ELECTRICAL ENGINEERING STUDENTS

Abstract: *This paper describes a methodology, with its activities and procedures, to teach the software engineering discipline to undergraduate electrical engineering students. This experience was developed at the Federal University of Amazonas in Manaus. This successful experience was developed under the UNIBRAL Program between the University of Amazonas in Manaus-Brazil and the University of Stuttgart in Germany. UNIBRAL is a program of cooperation between Germany and Brazil sponsored by CAPES Foundation, a body of the Brazilian Ministry of Education, jointly with DAAD - Deutscher Akademischer Austausch Dienst, a German academic cooperation body. Many Brazilian and German Universities are currently participating in this program. This program provides an opportunity to increase cooperation between these two universities aiming at introducing new curriculum guidelines and new courses, on the one hand, and increasing and renewing practical experiments, on the other. In this specific case, we introduced a practical experiment to help teaching software engineering based on a robotic platform. The results were very well motivated students and a very low drop out rate showing the success of our approach.*

Key-words: *Teaching Software Engineering, Laboratory Experiment on Software Engineering, Use of Robotic Artefact to Teach Engineering, Academic Exchanges, Brazil – German Cooperation.*