

3 Tecnologias para Aprendizagem de Programação em Grupo

Neste capítulo discute-se sobre as tecnologias e algumas metodologias que aliadas a essas tecnologias auxiliam na aprendizagem de programação em grupo. Para isso, iniciamos com relatos do estado da arte nesse tema, comparando abordagens que utilizam groupware e LMSs passíveis de utilização na aprendizagem de programação em grupo. Após essa exploração inicial, descrevemos exemplos de ferramentas desenvolvidas especialmente para apoiar programação em grupo. Continuando com a busca por diretrizes para unir tecnologias e metodologias para criar cursos introdutórios de programação em grupo, apresentamos modelos de cursos idealizados com esse propósito. Na última seção, discutimos então, em um contexto de utilização de um LMS, a aplicação de uma inspeção para se descobrir se os elementos de interface projetados visando aprendizagem de programação em grupo estavam adequados ao seu propósito, de acordo com um método de inspeção baseado em engenharia semiótica.

Há algum tempo, em computação, para facilitar a abstração do mundo complexo, adota-se uma estratégia de divisão e conquista a priori, assumindo-se que as peças divididas são independentes umas das outras (Dijkstra, 1982). Além disso, com o avanço no desenvolvimento das tecnologias, é necessário procurar entender como cada peça se encaixa no contexto (Lieberman & Selker, 2000).

Por esse motivo, é necessário se adaptar as ferramentas utilizadas para apoiar aprendizagem em geral a um contexto específico, envolvendo todos os tipos de tecnologias e metodologias necessárias e adequadas, independente de sua área de aplicação.

Acreditava-se que colaboração irrestrita e completamente natural ocorria sempre que se formavam grupos de interesses comuns, mas com o passar dos anos, tornou-se evidente que alguma forma de estrutura adicional é necessária para facilitar a aprendizagem e a interação (Dillenbourg & Fischer, 2007). Essa estrutura adicional deve estar presente nos ambientes computacionais utilizados

para aprendizagem. A partir de agora, para facilitar a leitura da tese, se o LMS se propuser a fornecer suporte também à colaboração, chamaremos de ambiente CSCL (do inglês Computer Supported Collaborative Learning).

Com o objetivo de analisar as interações em ambientes CSCL, de acordo com seu contexto de aprendizagem, conforme descrito em (Dillenbourg & Fischer, 2007), as abordagens baseadas em esquemas de codificação são, segundo o autor, “especialmente interessantes” por facilitarem a automatização do processo de análise. Essa possibilidade de automatização favorece análises de interação em tempo real, o que possibilita que o professor receba informações sobre quais grupos em sua turma precisam de sua ajuda em um determinado momento e que tipo de suporte é necessário, facilitando a intervenção.

Apesar de a intervenção ser um fator chave para a aprendizagem de programação, há relatos de experimentos com ambientes de programação colaborativa, não voltados para a aprendizagem, onde ela parece não ter tido muita importância. É o caso dos experimentos descritos em (Nosek, 1998), onde são enfatizadas algumas vantagens em se investir mais em práticas de colaborativas. Nas duas edições desses experimentos os times desenvolveram programas mais eficientes, demonstrando, em alguns casos, soluções muito criativas para problemas desafiadores e importantes em seu domínio natural.

O trabalho de Nosek afirma que a programação colaborativa deve ser incentivada por questões de eficiência. No entanto, um ambiente CSCL com suporte à programação deveria possuir mecanismos mais específicos para acompanhamento da codificação dos grupos.

Para se intervir em ambientes CSCL configurados para programação, é necessário que estes possuam ferramentas de coordenação de atividades relacionadas ao processo de desenvolvimento, de forma que as oportunidades para de intervenção sejam identificadas e direcionadas rapidamente ao professor para que assim consiga intervir e modificar a atitude do grupo assim que a dificuldade surge. A ferramenta AcKnow, detalhada no capítulo anterior, foi desenvolvida para avaliar a evolução dos códigos individuais dos alunos, mas poderia ser utilizada, em conjunto com práticas de programação em grupo, para avaliar a evolução dos códigos do grupo. Nesse caso, a análise poderia ser realizada dentro do espaço do grupo no ambiente CSCL utilizado.

Sabe-se (Davis, 1989) que todos os sistemas de ensino e aprendizagem deveriam ser construídos em duas fundações: as necessidades dos alunos e a consequência da aprendizagem do curso ou programa. Um ambiente de aprendizagem ideal tem que ser baseado em um plano que flui de um entendimento completo desses dois fundamentos (Davis, 1989). Se um plano de utilização do ambiente é necessário para aprendizagem em geral é importante também para aprendizagem de programação em grupo.

Corroborando com a importância da aprendizagem de programação em grupo com apoio de ambientes CSCL, em (Aragon, Johnson & Shaik, 2002) são definidos os requisitos para aprendizagem nesses ambientes e são feitas algumas recomendações para quem for trabalhar com esses ambientes: criar times virtuais de aprendizagem; simular a realidade utilizando estudos de caso apropriados; e requisitar projetos colaborativos das escolas, empresas ou outras organizações.

Na seção seguinte selecionamos e descrevemos algumas ferramentas desenvolvidas para apoiar a programação em grupo, considerando além da necessidade de representação de contexto em ambientes CSCL, a identificação de oportunidades de intervenção nesses ambientes e as recomendações para cursos que utilizam esses ambientes.

3.1.Ferramentas para Apoiar a Aprendizagem de Programação em Grupo

Das ferramentas pesquisadas no Google até início de Janeiro de 2011, foram selecionadas as que apresentavam uma descrição consistente e resultados claros de análise de utilização. Como cursos introdutórios utilizam diferentes paradigmas de programação, procuramos fazer uma análise abrangente, iniciando com ferramentas voltadas à produção de algoritmos. Em seguida, apresentamos uma que utiliza linguagem de programação estruturada, envolvendo um processo de solução de problemas. Outras ferramentas desenvolvidas especificamente para programação distribuída e as voltadas à percepção das atividades de times de desenvolvimento precedem um relato sobre uma análise comparativa das ferramentas que envolvem visualização.

Scratch é uma ferramenta educacional para o entendimento de construção de algoritmos (Jain, Singhal & Gupta, 2010). Ela foi projetada para facilitar a

manipulação de mídia por alunos iniciantes em programação. Segundo seus autores, ela estimula os alunos a desenvolverem programas e aprenderem sobre objetos, pois basta arrastar e organizar blocos, no estilo Lego, para construir um algoritmo. Como extensão ao Scratch, os autores propõem um framework colaborativo para o compartilhamento de idéias e códigos implementados. O aluno envia seu código diretamente a outro usuário ou compartilha-o com um grupo, que analisam e editam o código e o devolvem com anotações.

Na mesma linha voltada à construção de algoritmos está o RAPTOR (Carlisle, Wilson, Humphries & Hadfield, 2004) que utiliza símbolos gráficos, como caixas de fluxograma, em vez dos blocos Lego do Scratch. Os alunos, então, executam esses algoritmos no ambiente nos modos contínuo ou passo a passo. O ambiente visualmente mostra a localização do símbolo em execução assim como os conteúdos de todas as variáveis.

No sentido de viabilizar uma maior percepção de contexto, é descrito em (Selker, 1994) o COgnitive Adaptive Computer Help (COACH), que é um sistema de ajuda que monitora as ações do usuário. Quando um usuário inicia uma atividade não familiar, o COACH lhe apresenta conselhos proativamente. Esse sistema foi desenvolvido para proporcionar aprendizagem de programação através do uso de “helps”. Assim, COACH é um sistema de help proativo e adaptativo que explica procedimentos em termos de contexto do próprio usuário.

Em cursos de programação há muitas dificuldades a considerar, incluindo as formas de estimular as interações dos alunos na aula ou entre aulas, métodos de enriquecer as experiências de aprendizagem dos alunos e meios de ajudar os alunos no compartilhamento de conhecimento com seus colegas. Em cursos tradicionais de programação, programação baseada em solução de problemas é considerada uma abordagem promissora (Hwang, Wang, Hwang, *et al.*, 2008). A principal dificuldade encontrada por iniciantes em programação é em expressar soluções de problemas como programas. Portanto, a abrangência de compreensão de programação e como aplicá-la à geração de programas devem permanecer um foco importante. O WPAS foi desenvolvido para dar suporte às atividades de aprendizagem, elaboradas segundo a taxonomia de Bloom voltado ao desenvolvimento cognitivo. Fazendo uma correspondência com a taxonomia de Bloom, as atividades elaboradas envolvem: teste de conceitos de programação, preenchimento de lacunas no programa, depuração de código, codificação para

resolver problemas e revisão por pares, relativas aos níveis de desenvolvimento cognitivo: conhecimento e compreensão, aplicação, análise, síntese e avaliação.

Trabalho em grupo pode ser apoiado com outras ferramentas groupware como anotações de código. Programação pode ser apoiada, por exemplo, com destaque de sintaxe. A transmissão de código e visualização deve ser feita mais facilmente. Além disso, É necessário que se tenha suporte para a edição de código fonte em grupo (Moreno, Myller & Sutinen, 2004).

RECIPE (Shen & Sun, 2000) é um protótipo para dar suporte à programação colaborativa. Essa ferramenta possibilita o compartilhamento de códigos em Java, com níveis de permissão diferentes, fornecidos pelo autor. Ela integra também um compilador e os códigos compartilhados são visualizados e editados ao mesmo tempo por múltiplos usuários, desde que tenham permissão para isso.

Em (Hillegersberg & Herrera, 2007) é apresentada uma visão geral e uma proposta de requisitos de usuários para ambientes de apoio ao desenvolvimento em times de desenvolvimento geograficamente distantes. Os requisitos são: descrever e manter informação dos dados durante o ciclo de vida do desenvolvimento do produto; manter coordenação e controle durante um projeto e transversalmente a projetos; possuir acessibilidade remota ao ambiente de projeto; negociar e encontrar consenso com membros de times de outros ambientes semelhantes.

Em (Jo & Arnold, 2003) é apresentada uma experiência com projeto e implementação de uma ferramenta colaborativa para suporte à programação na Internet. Ele dá suporte comunicação síncrona entre os diferentes dispositivos e participantes. A ferramenta DPE é então um tipo de groupware que suporta programação distribuída por engenheiros de software. DPE possui mecanismos de discussões online, banco de dados de software, ambiente de programação integrado e ferramentas para engenharia de software distribuída.

JeCo (Moreno, Myller & Sutinen, 2004b) é uma ferramenta colaborativa simplificada para dar suporte à aprendizagem de programação. A proposta é fornecer uma ferramenta que promova a visualização de programas e facilite a colaboração para aumentar o compromisso dos alunos. JeCo enfatiza aprendizagem em grupo aumentando a percepção dos times de alunos. JeCo integra dois sistemas existentes, Jeliot3 e Woven Stories. O Jeliot 3 anima a execução de programas. O Woven Stories é uma ferramenta de co-autoria que

representa o histórico do documento em forma de grafo (Moreno, Myller & Sutinen, 2004).

Jazz (Cheng, Hupfer, Ross, Patterson & Clark, 2003) é baseado em uma metáfora de escritório aberto para desenvolvimento de aplicações. Seguindo a proposta da Extreme Programming (Beck, 1999), desenvolvedores organizadores em um time trabalham próximos, cada um em sua estação de trabalho, onde haverá em outro lugar um espaço para as reuniões dos times, compartilhamento de quadros e horários, etc. O ponto chave dessa organização do trabalho é percepção (*awareness*) do time. O aspecto mais visível dos melhoramentos do Jazz para o Eclipse é a “Jazz Band”, uma visualização que proporciona ao usuário uma percepção periférica de outros membros do time. Ela mostra o usuário e as listas de times às quais o usuário pertence assim como membros de outros times. Os times no Jazz são esperados serem pequenos, informais, *ad hoc* e baseados em convite: qualquer um pode criar um time e acrescentar qualquer um ao seu time. Para estimular a colaboração foram criados os chats ancorados [Hupfer, Cheng, Ross, et al, 2004]. Nesses chats um desenvolvedor tem a opção de destacar uma região de código no editor, clicar e iniciar um chat sobre ela. Assim que a discussão termina, uma transcrição é salva e aparece como um marcador na margem esquerda do código. Os membros do time podem posteriormente rever o chat, clicando no marcador.

Em (Storey, Cubranic & German, 2005) é descrito um levantamento na literatura sobre ambientes de visualização de código para embasar a proposta de se utilizar visualização de códigos e informações relacionadas como perfil do desenvolvedor no desenvolvimento de programas como forma de fornecer mecanismos de percepção para desenvolvimento de software. Os autores definem um framework, contendo os critérios utilizados para avaliar as ferramentas. Eles concluem, no entanto, que segundo aqueles critérios todas as ferramentas analisadas ainda precisam de muitas melhorias para que atendam a seus propósitos. As ferramentas de visualização de código têm um grande apelo, porém ainda não possuem as funcionalidades necessárias para apoiar a aprendizagem de programação, pois tal processo envolve a necessidade de coordenação do curso, juntamente com recursos comuns a LMS em geral, acoplados a um ambiente de desenvolvimento de software.

As ferramentas descritas nesta sessão propiciam a colaboração espontânea, considerando que os participantes possuem a necessidade de colaborar em busca de uma troca de idéias e não apenas em busca de soluções prontas para suas dificuldades. No contexto da aprendizagem de programação em cursos introdutórios, além desses mecanismos de visualização e compartilhamento de códigos são necessários aportes teóricos e tecnológicos que promovam a aprendizagem.

3.2. Aportes Metodológicos e Tecnológicos como Apoio a Disciplinas de Programação Introdutória

O projeto Studio-1.00 focou no desenvolvimento de um novo currículo e a reconfiguração de facilidades para suporte ao ensino da disciplina introdutória de programação do MIT como um curso baseado em estúdio. O projeto tinha o objetivo de melhorar as técnicas de aprendizagem ativa, programação interativa e a exploração e a descoberta de métodos e conceitos de desenvolvimento de software. Tudo isso facilitado com o uso de dispositivos móveis em uma sala de aula eletrônica (Barak, Lipson & Lerman, 2006). Nesse formato, o curso incluía 3 sessões de estúdio por semana, cada uma com 90 minutos e um tutorial para grupos pequenos (12 alunos), com duração de 60 minutos.

Como resultado da análise (Barak, Lipson & Lerman, 2006), os notebooks ligados à rede sem fio possibilitaram a integração do formato de estúdio em uma sala de aula tradicional, somente com carteiras e quadro branco, sem necessidade de laboratórios especialmente projetados. Os instrutores e alunos conseguiram aproveitar todas as vantagens do uso de computadores como ferramentas cognitivas em uma sala com configuração padrão. A tecnologia funciona melhor quando é mesclada com outros elementos do processo de aprendizagem e atende a uma necessidade educacional específica que não tinha sido contemplada de outra maneira.

No contexto de um curso introdutório de Engenharia de Software [Chen, Liu, Sun & Wang 2010] notou-se a necessidade de se desenvolver habilidades colaborativas nos alunos para que fossem mais bem preparados para o mercado de trabalho. Foi definido um modelo de aprendizagem que é uma combinação de aprendizagem baseada em problemas e aprendizagem baseada em tarefas, o

PTBL. Esse modelo é combinado com tecnologias Web3D para ampliar as habilidades de trabalho em equipes de desenvolvimento.

A abordagem de aprendizagem baseada em problemas é utilizada para ensinar e treinar habilidades de desenvolvimento de programas. Os alunos, após receberem os problemas especialmente para treinar uma habilidade são levados a se comunicar com os outros integrantes de seus times. A abordagem baseada em tarefas é utilizada para proporcionar ao aluno uma circunstância real que pode motivá-los a estudar. As tecnologias Web3D, envolvendo jogos multiusuários, são unidas à abordagem baseada em tarefas e a tarefa é definida como um jogo, que os times de alunos deverão entregar para completar a tarefa.

Assim que os alunos iniciam no curso, devem primeiro se organizar em grupos e escolher um dos jogos descritos por professores e assistentes. Os próprios alunos em seus grupos são responsáveis pela divisão de tarefas, de acordo com as tarefas envolvidas em cada jogo. Eles devem se comunicar e colaborar entre si, o que vai influenciar diretamente as notas que irão receber.

Os modelos apresentados não analisam como ocorre a colaboração entre os membros das equipes, concentrando-se somente nos resultados. No entanto, a análise dos cursos evidencia uma demanda por tecnologia que auxilie a intervenção do professor durante o processo, servindo de subsídio para uma nova proposta de organização do curso introdutório que promova a colaboração em programação.

3.3. Proposta de Adaptações de um ambiente CSCL para o Contexto da Aprendizagem de Programação

No contexto de um projeto de pesquisa da UFAM, foi desenvolvido o ColabWeb, um ambiente CSCL configurado e desenvolvido utilizando a plataforma Moodle. Este ambiente é utilizado como apoio às disciplinas presenciais ministradas por professores do Departamento de Ciência da Computação. O projeto de interface para o ColabWeb privilegiou sua funcionalidade, o que fez com que os primeiros cursos ocorressem com um aproveitamento desequilibrado de seus recursos. Em vista dessa diferença em termos de sucesso no uso daquele groupware, evidenciada neste trabalho através

do gerenciamento de cursos de computação introdutória, decidiu-se fazer uma inspeção que evidenciasse o poder de comunicação do artefato de software (ColabWeb). Para isso, é utilizado o Método de Inspeção Semiótica (MIS) (De Souza *et al.*, 2008).

Outros métodos de avaliação aplicados a ambientes CSCL já foram utilizados em contextos de curso, como os descritos em (Pimentel, Fuks & Lucena, 2004) que avalia a participação dos alunos em fóruns, (Fuks *et al.*, 2003) que avalia a participação dos alunos de um curso em todas as atividades do groupware utilizado e (Pimentel, Fuks & Lucena, 2003b) que avalia a participação em debates. No entanto, nenhum desses trabalhos considera que a interface influencia na avaliação da participação dos alunos nas atividades propostas em um ambiente CSCL. Tal aspecto só é considerado em métodos de inspeção baseados na semiótica como é o caso do MIS.

Conforme a expectativa do MIS, foram elencadas recomendações de melhorias para a comunicabilidade do ColabWeb. Este trabalho propõe e inaugura uma linha de avaliação de ambientes de aprendizagem com ênfase na comunicabilidade. Avalia-se uma instância (ColabWeb) de uma arquitetura amplamente utilizada (Moodle), observando-se unidades, que neste caso são cursos. A partir dessas unidades generalizam-se as observações para a instância.

3.3.1. A Engenharia Semiótica

A Engenharia Semiótica propõe uma abordagem para IHC (Interação Humano-Computador) centrada na comunicação, onde o designer dos sistemas computacionais são atores ativos na comunicação com o usuário. Segundo De Souza (2005), enquanto teoria de IHC, as metas da engenharia semiótica são: apresentar uma caracterização de IHC (extensa e distinta); prover uma ontologia consistente da qual podem ser derivados frameworks e modelos, sobre aspectos particulares de IHC; e detalhar restrições epistemológicas e metodológicas, bem como condições aplicáveis ao espectro da pesquisa na qual a teoria possa servir de apoio.

Ao se construir um ambiente computacional, segundo a engenharia semiótica, é necessário refletir sobre a metacomunicação da interface com o usuário. Para isso, é adotado o modelo de funções comunicativas de Jakobson

(Jakobson, 1960) *apud* (De Souza, 2005), o qual propõe funções comunicativas. Estas funções são utilizadas para a construção de artefatos de metacomunicação em IHC. O foco de cada função está em um dos seguintes elementos: o autor, o receptor, o contexto, o canal, o código e a mensagem. Isto é explorado indiretamente ao longo do trabalho de inspeção semiótica, uma vez que o próprio método analisa aspectos relacionados a essas funções comunicativas.

Pode ser que o pesquisador de IHC não queira ou não possa construir um ambiente computacional. Isto ocorre porque seu público alvo (usuários) já utiliza um ambiente ou porque sua instituição adota uma determinada plataforma computacional. Neste caso, é necessário avaliar os elementos de IHC por meio das funções comunicativas que já estão presentes no ambiente, a fim de propor modificações para melhorar a metacomunicação com o usuário ou, se o ambiente atender às funções comunicativas conforme desejado, relatar e registrar esses elementos.

A fim de avaliar o efeito da comunicabilidade dos sistemas computacionais, a engenharia semiótica propõe dois métodos, o Método de Inspeção Semiótica (MIS) (De Souza *et al.*, 2006) e o Método de Avaliação da Comunicabilidade (MAC) (Prates, De Souza & Barbosa, 2000). O MIS é utilizado para inspecionar um artefato computacional, considerando o designer como ator ativo através de sua metacomunicação com o usuário, sem a necessidade de fazer testes com usuários. Já o MAC propõe uma abordagem mais centrada na utilização do artefato pelo usuário, envolvendo testes com usuários considerando um aspecto de interface. Vale salientar que neste trabalho somente é utilizado o MIS, dada a necessidade de se efetuar uma inspeção geral no artefato, antes de se concentrar em aspectos específicos de sua interface e pelo fato de a análise ter sido realizada após a utilização do ambiente CSCL como apoio a uma disciplina ministrada no semestre anterior.

3.3.1.1. O Método de Inspeção Semiótica

Conforme definido em (De Souza *et al.*, 2008), o propósito do Método de Inspeção Semiótica (MIS) é avaliar a comunicabilidade de artefatos computacionais interativos. O MIS é, portanto um método que avalia a comunicabilidade, concentrando-se nos significados da interface com o usuário

expressos pelo designer. O método auxilia os inspetores a anteciparem os tipos de consequência que as escolhas de projeto (design) podem trazer quando usuários reais interagem com o sistema.

O MIS é aplicado em 5 etapas: (i) análise dos signos metalinguísticos, (ii) análise dos signos estáticos, (iii) análise dos signos dinâmicos, (iv) uma comparação entre a mensagem de metacomunicação do designer gerada nos passos anteriores, e (v) uma avaliação final da comunicabilidade do sistema inspecionado.

Antes de se iniciar o MIS propriamente dito, é necessária uma fase de preparação, que envolve um levantamento sobre a documentação existente e uma organização prévia do espaço a ser analisado, por exemplo, a criação de grupos ou inscrição em grupos existentes. Nesse trabalho será analisada a metamensagem do designer do ColabWeb, ambiente CSCL inspecionado, considerando a metamensagem do designer do Moodle, pois é a plataforma onde o ColabWeb foi implementado. Em (i) analisa-se os signos metalinguísticos existentes na documentação, help do sistema ou meta-mensagem na tela. Os signos estáticos analisados em (ii) são aqueles cujo significado é interpretado independentemente das relações temporais e causais, sendo o contexto da interpretação limitado aos elementos presentes na interface em um dado momento. Em (iii) são consideradas as transições na interface, como consequências das ações realizadas sobre uma tela. As três primeiras etapas se combinam para transmitir a mensagem de metacomunicação do designer (iv), que pode ser parafraseada ao seguinte esquema:

“Aqui está o meu entendimento de quem você é, o que eu aprendi que você quer ou precisa fazer, em qual jeito você prefere fazer e porquê. Este é o sistema que eu projetei pra você e este é o jeito que você pode ou deve usá-lo para satisfazer seus propósitos que casam com esta visão”.

Na última etapa do MIS (v) avalia-se a comunicabilidade do sistema, reconstruindo uma mensagem de metacomunicação unificada e julgando os custos e benefícios das estratégias comunicativas identificadas nas etapas anteriores.

O MIS pode ser aplicado a qualquer artefato de software de onde se queira inspecionar o poder de comunicação do designer com o usuário. Para groupware, há preocupações próprias, portanto é necessário se observar outras interações, além das de usuário com designer (De Souza, 2004). Pelo menos um trabalho

exemplifica o MIS para groupware, publicado em (Guimarães & De Souza, 2008). A diferença dessas inspeções anteriores para a realizada neste trabalho é quanto à unidade, conforme mencionado anteriormente. Em vez de se avaliar o ColabWeb de forma geral, o cenário de inspeção escolhida conduz à avaliação da metacomunicação em cursos, posteriormente generalizada para o software.

Na próxima será feita uma descrição sobre a utilização do ColabWeb (Santos, Castro & Castro, 2007) como ferramenta de apoio à aprendizagem de programação introdutória. O ColabWeb é um groupware baseado na arquitetura Moodle, desenvolvido incrementalmente e utilizado para gerenciar artefatos digitais produzidos pelos alunos nas disciplinas de graduação e pós-graduação de computação na Universidade Federal do Amazonas (UFAM).

3.3.2. O Contexto da Aprendizagem de Programação utilizando o ColabWeb

No primeiro semestre de cada ano acadêmico, são oferecidas quatro turmas da disciplina Introdução à Computação na UFAM para alunos cursando, preferencialmente, o primeiro semestre de seus cursos. Dessas quatro turmas, duas são dirigidas aos alunos de Ciência da Computação e duas para os alunos de Engenharia da Computação. Ao todo são aproximadamente 110 alunos por semestre.

Em 2005 começou-se a utilizar ferramentas de apoio para o ensino de Introdução à Computação, devido à grande quantidade de alunos. Como resultado de estudos de caso da utilização dessas ferramentas (Almeida, Castro & Castro, 2006), as turmas de 2008 utilizaram o ColabWeb, devido às suas funcionalidades de gerenciamento de grupo e conteúdo. Esta última funcionalidade foi necessária, pois segundo a evolução da metodologia adotada no curso, os alunos passaram a ser divididos em grupos para resolverem exercícios de programação. Consequentemente, os alunos se engajaram em outras atividades, como por exemplo, discussões em grupo, escrita cooperativa de relatórios e reflexões sobre os códigos. Para tanto, foram criados 3 cursos no ColabWeb que serão descritos a seguir.

O curso IC-Apoio serve como site de apoio para a disciplina, contendo informações comuns a todas as turmas, ementa e discussões mais gerais dos

alunos, servindo também de socialização entre os alunos dos diferentes cursos. Os outros dois cursos são cursos propriamente ditos: IC-CComputação, com as duas turmas de Ciência da Computação, e IC-EngComputação, com as duas turmas de Engenharia da Computação. Apesar da expectativa de que ambos os cursos fossem uniformemente frequentados, a utilização não foi homogênea, o que será discutido nas próximas sessões.

3.3.3. Inspeção Semiótica do ColabWeb

Dado que o ambiente ColabWeb é uma instância do ambiente Moodle,¹ foi a partir deste ambiente que foi realizada uma inspeção inicial para se descobrir quem eram seus possíveis usuários na visão do designer, representante da equipe de desenvolvimento do Moodle, e se há documentação suficiente para todos os tipos de usuário que este designer pretende atender. Constatou-se que a documentação, apesar de pouca, era consistente com os propósitos da ferramenta. O Moodle, então, pode ser visto como uma ferramenta para desenvolvedores e que, devidamente estendida, conforma-se com as necessidades específicas de cada instituição que a utiliza.

Como o projeto do Moodle tem como meta fornecer um sistema totalmente configurável para o professor gerenciar seus cursos, há muitos elementos na configuração de cada curso, o que por sua vez requer uma análise minuciosa para avaliar sua adequação a cada tipo de curso. O ColabWeb manteve esses elementos totalmente configuráveis assim como o projeto de interface original. Isso deixa o cenário de inspeção muito amplo e difícil de analisar, necessitando para isso delimitar um escopo.

A fim de se delimitar o escopo da inspeção, o curso utilizado foi a complementação virtual da disciplina Introdução à Computação, ministrado para alunos do primeiro semestre de Ciência e Engenharia da Computação. Para este trabalho foram analisadas quatro turmas do curso, três que já ocorreram durante o semestre acadêmico de 2008.1 e outro desenvolvido como exemplo. Na última etapa do MIS, isto é, na avaliação final, são propostas adaptações, considerando as limitações da arquitetura utilizada. Os cursos analisados (conforme definição do

¹ <http://moodle.org>

ColabWeb), e utilizados por alunos, professores e monitores, foram: “IC-CComputação”, “IC-EngComputação” e “IC-Apoio”. O curso “EngSemiotica” foi criado com configurações padrão, para que fosse possível observar quais aspectos de qualquer instância do Moodle podem ser modificados.

O cenário de inspeção elaborado foi o seguinte: “Joana é uma aluna recém chegada ao curso de Ciência da Computação e está matriculada na disciplina Introdução à Computação. Nesta disciplina ela vai aprender a programar utilizando Haskell, uma linguagem funcional. As atividades propostas, bem como avisos e notas serão acompanhadas no ambiente computacional ColabWeb. O problema é que ela nunca usou um ambiente gerenciador de cursos e, devido à necessidade de estudar programação, não tem muito tempo de percorrer o ambiente para se acostumar com o esquema de navegação e organização em seu curso. Por outro lado, seu professor criou o curso IC-Apoio no mesmo ambiente, que serve de apoio às atividades propostas no curso IC-CComputação, com informações gerais sobre a disciplina. Ela espera não ficar confusa em ter que utilizar esses dois cursos dentro de um ambiente que não conhece, pois precisa fazer os trabalhos.”

3.3.3.1. Etapas do MIS

As cinco etapas do MIS apresentadas anteriormente na Seção 3.3.1.1 são instanciadas abaixo para o ColabWeb, o ambiente CSCL que utilizamos na pesquisa desta tese.

Etapa 1 – análise dos signos metalingüísticos

Nesta etapa é realizada uma inspeção na documentação disponível (on-line e off-line) sobre a ferramenta em questão, buscando a meta-mensagem do designer para o usuário. No caso do ColabWeb, não há documentação disponível sobre a ferramenta, somente um artigo científico (Santos, Castro & Castro, 2007) que discorre sobre a funcionalidade de gerenciamento de grupos, uma extensão ao Moodle original, descrita como “... a percepção dos indivíduos enquanto parte de um ou vários grupos, além da possibilidade de re-organização em subgrupos”.

Dado que o ColabWeb herda muitas das restrições e funcionalidades do Moodle, é importante citar a mensagem do designer que aparece na abertura do

site do Moodle (traduzido para português): “O Moodle é um Sistema Gerenciador de Cursos (SGC), também conhecido como um Sistema Gerenciador da Aprendizagem (SGA) ou Ambiente Virtual de Aprendizagem (AVA). É uma aplicação web gratuita que educadores podem utilizar para criarem sites de aprendizagem efetivos”. Há alguns manuais disponíveis no site, mas conforme consta na descrição transcrita anteriormente o Moodle foi projetado para ser estendido por programadores ou utilizado por instrutores de cursos na web, que ora assumem o papel de administradores de cursos ora de tutores ou professores. Portanto, não há documentação para um dos usuários finais, o aluno. O designer do Moodle entende que isto, a documentação, cabe ao administrador ou professor.

Apesar de o projeto do Moodle não possuir documentação para alunos, no site há alguns exemplos de documentação, entre elas “Moodle: An introductory user guide for students”. Neste documento, o designer se refere ao Moodle como uma ferramenta bastante intuitiva e convida o usuário a experimentar o sistema “tente e veja o que acontece” (trad. do autor). Este e os outros exemplos de documentação para o usuário final foram desenvolvidas juntamente com uma instância do Moodle e a presença de uma página de exemplos indica que é isso que o designer do Moodle espera que seus usuários (designers de sites) façam.

Quanto ao ColabWeb, ao se entrar em qualquer curso não há sequer uma “dica” de como proceder, ou do que os ícones significam. Foi observada também a ausência de um help para o usuário. Com relação ao curso IC-CComputação há descrições gerais sobre as atividades a serem desenvolvidas feitas pelo professor (designer de curso), como o que consta no wiki para o exercício 4, conforme ilustrado na Figura 3.1. Os signos metalinguísticos presentes na interface do ColabWeb podem ser muitos ou poucos, dependendo do designer do curso. Nos cursos analisados esses signos quase não existem, com exceção das meta-mensagens conforme exemplificado na Figura 3.1, “Aqui deve ser feito o relatório do exercício 4”.



Figura 3.1 – Página de Exemplo no wiki

O MIS sugere que se vá construindo a metamensagem do designer à medida que se conclui cada etapa do método. Esta metamensagem deve ser baseada em um esquema de metacomunicação, descrito na seção anterior. Há dois usuários finais para o ColabWeb, um é o aluno, segundo o projeto do ColabWeb. O outro é o professor, uma vez que o ColabWeb herda parte do projeto do Moodle. Para essa inspeção consideramos somente o aluno como usuário final. Em seguida está a meta-mensagem inicial de comunicação do designer do ColabWeb para seu usuário final (aluno). O texto entre colchetes é o esquema de metacomunicação.

[Aqui está o meu entendimento sobre quem você é]. Você é um aluno de computação, que já está acostumado a utilizar ambientes de redes sociais.[O que eu aprendi que você quer ou precisa fazer, de que forma e porquê]. Você precisa de um sistema para gerenciar a disciplina que você está matriculado, podendo interagir com outros alunos, resolvendo exercícios e acompanhando as instruções do professor. Você precisa ser capaz de visualizar as atividades de seus colegas e escrever textos em ferramentas auxiliares como wikis, fóruns e chats. [Este é o sistema que eu projetei pra você e essa é a maneira pela qual você pode ou deve utilizá-lo de forma que preencha uma variedade de propósitos que coadunam com esta visão]. Você pode usar o calendário para verificar os eventos do curso, a estação de percepção para estar a par automaticamente da movimentação em seus outros cursos e ainda subdividir seus grupos em subgrupos, conforme a atividade pretendida.

Etapa 2 – análise dos signos estáticos

A Figura 3.2 apresenta a tela de abertura do ColabWeb. A mensagem “login” é substituída pela palavra “acesso”, o que não é um jargão muito comum nesses ambientes. Além disso, não é bem comunicada, pois se localiza na parte superior, entre parênteses e utilizando fonte muito pequena.

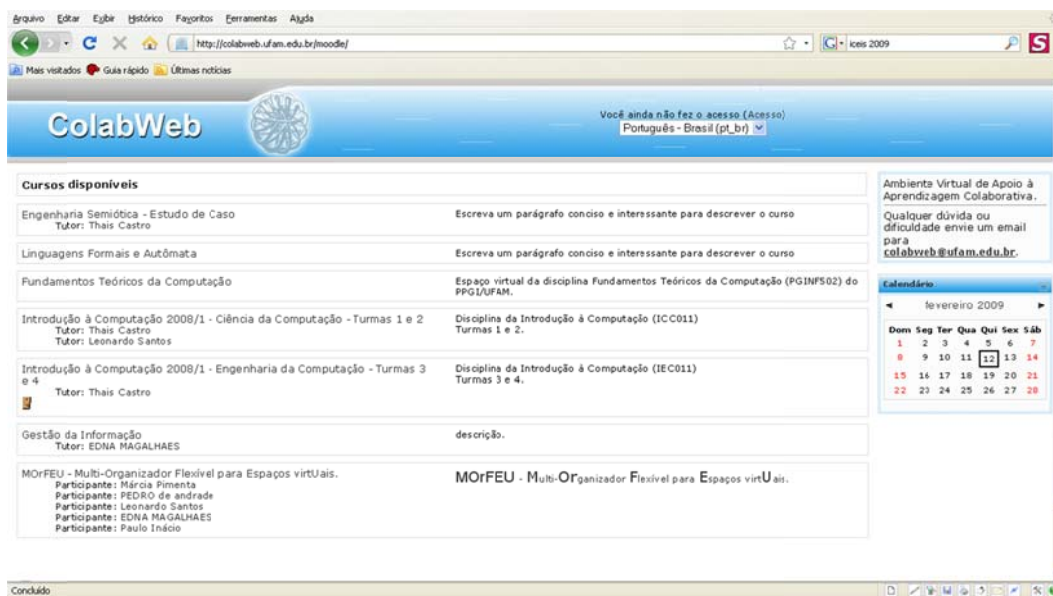


Figura 3.2 – Página de abertura do ColabWeb

Conforme ilustrado na Figura 3.3, no curso IC-CComputação, aparecem inicialmente os grupos existentes, à esquerda, e no centro, em “Grupos” o designer do curso, neste caso o professor, destaca a palavra “importante” para chamar atenção do usuário quanto ao seu grupo. Isto é um bom recurso quando não se consegue eliminar os outros grupos da visão do usuário. À direita, há um recurso chamado estação de percepção (Spósito, Castro & Castro, 2008) monitora as atividades nos cursos que o usuário pertence. As palavras padrão que aparecem, Meus Cursos e Monitorar Tudo são úteis para o seu entendimento. Caso o usuário selecione outra palavra, as caixas abaixo da Estação de Percepção são alteradas.

Da mesma forma que o link para entrar no ambiente, o link para sair está muito pequeno. O seu posicionamento é bom, pois aparece centralizado com o nome do ambiente, mas está entre parênteses e utilizando fonte pequena.

Figura 3.3 – Predominância da Linguagem Textual

A Figura 3.4 mostra o recurso “Calendário”, presente nesse curso. Apesar de útil, marcando as datas ocupadas com algum tipo de atividade, abaixo do calendário aparecem “events key”, que por estarem escritos em outro idioma, confundem o usuário, segundo declarações feitas ao professor da disciplina (designer do curso) e podem numa última instância tornar o recurso inútil.

Figura 3.4 – Recurso Calendário

A Figura 3.5 mostra a estruturação de tópicos no curso IC-EngComputação. Neste caso, o professor decidiu organizar o curso por tipos de atividade. Deste modo, ao clicar em uma atividade, o usuário visualiza também as outras atividades do mesmo tipo e perde o contexto do curso, como por exemplo, o tópico do curso abordado naquela atividade.



Figura 3.5 – Perda de Contexto

Em seguida está o refinamento da metamensagem do designer ao final desta 2a. etapa, baseada somente nos signos estáticos, onde o texto entre colchetes é o esquema de metacomunicação do designer, o texto riscado é o que foi retirado e o texto em destaque, o que foi acrescentado.

[Aqui está o meu entendimento sobre quem você é]. Você é um aluno de computação, que já está acostumado a utilizar ambientes ~~de redes sociais~~ **computacionais**. [O que eu aprendi que você quer ou precisa fazer, de que forma e porquê]. ~~Você~~ **Seu professor** precisa de um sistema para gerenciar a disciplina que você está matriculado ~~podendo interagir com outros alunos, resolvendo exercícios e acompanhando as instruções do professor~~ e **se comunicar com você**. Você precisa interagir com os outros alunos de seu grupo, resolvendo exercícios e acompanhando as instruções do professor. Você precisa ser capaz de visualizar as atividades de seus colegas ~~escrever textos em ferramentas auxiliares como~~ e **colaborar com eles** utilizando wikis, fóruns e chats. [Este é o sistema que eu projetei pra você e essa é a maneira pela qual você pode ou deve utilizá-lo de forma que preencha uma variedade de propósitos que coadunam com esta visão]. Você pode usar o calendário para verificar os eventos do curso, a estação de percepção para estar a par automaticamente da movimentação em seus outros cursos e ainda subdividir seus grupos em subgrupos, conforme a atividade pretendida.

Etapa 3 – análise dos signos dinâmicos

A partir da lista de grupos mostrada na página principal do curso é possível selecionar qualquer grupo em que o usuário esteja inscrito. Esta ação gera uma página como a mostrada na Figura 3.6. Um primeiro problema é que o usuário vê quem participa em qualquer grupo, inclusive seus logs de acesso. Mais grave é o problema quanto à navegação. O usuário fica perdido, pois precisa selecionar um dos links hierárquicos na parte superior da tela, algo que nem sempre consegue enxergar. Caso selecione grupos, o usuário receberá uma lista com links para grupos, o que é totalmente inútil e confuso visto que o usuário chegou à página atual selecionando um grupo na página principal do site do curso (Figura 3.7).

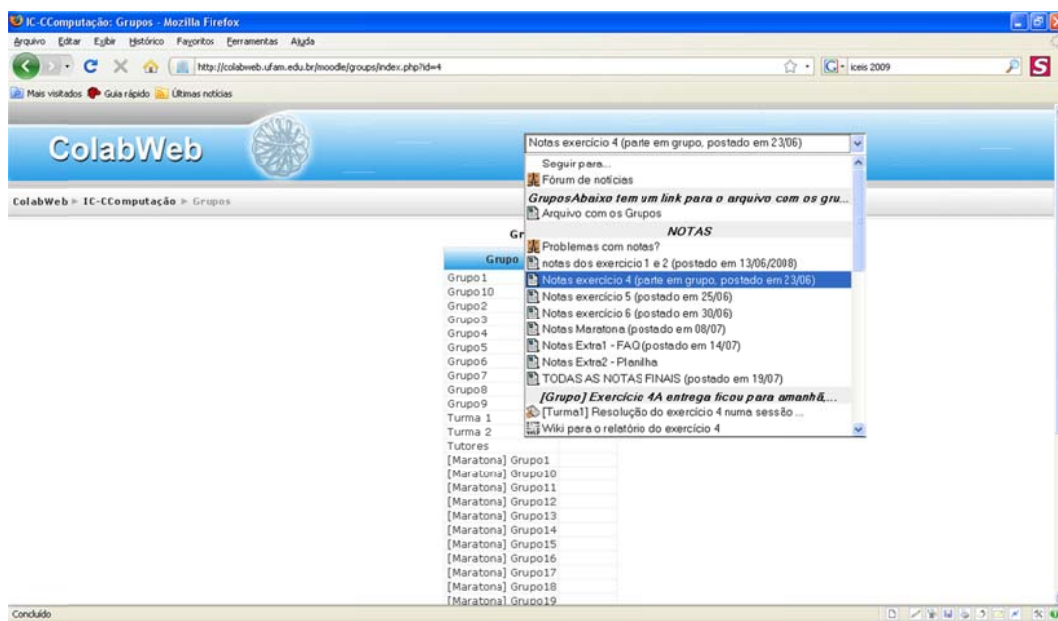


Figura 3.6 – Visualização de Informações sobre Grupo

Ainda na Figura 3.7, a caixa de diálogo “Seguir para...” é de difícil visualização. Caso o usuário consiga enxergá-la, ele poderá selecionar um dos links que lá aparecem. Isto mitiga o problema da falta de referencial.

A seguir apresentamos o refinamento da metamsagem do designer ao final desta 3a. etapa. O que está entre colchetes é o esquema, o que está riscado é o texto retirado e o que está destacado é o que foi acrescentado ao final desta etapa.

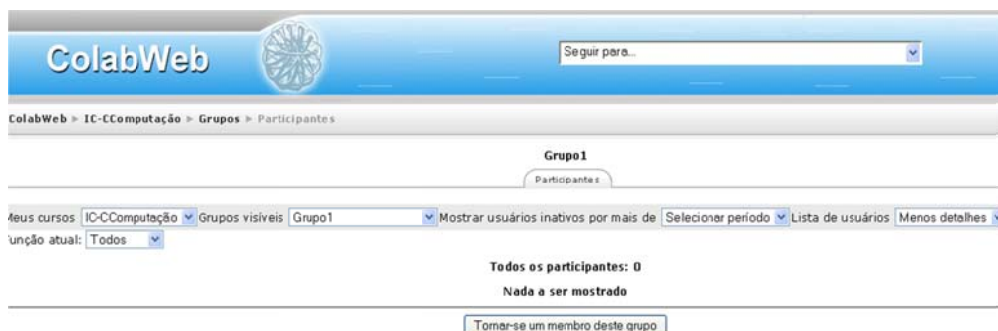


Figura 3.7 – Navegação nos Grupos

[Aqui está o meu entendimento sobre quem você é]. Você é um aluno de computação, que já está acostumado a utilizar ambientes computacionais. [O que eu aprendi que você quer ou precisa fazer, de que forma e porque]. Seu professor precisa de um sistema para gerenciar a disciplina que você está matriculado e se comunicar com você. Você precisa interagir com os outros alunos de seu grupo, resolvendo exercícios e acompanhando as instruções do professor. Além disso, você pode visualizar qualquer atividade de seus colegas, através do link de grupos dentro do curso e depois caminhar pelos links superiores para a parte do ambiente que você quiser ir. Você precisa ser capaz de visualizar as atividades de seus colegas e colaborar com eles utilizando wikis, fóruns e chats. Para isso, você também pode acessar os recursos isoladamente, visualizando todos os daquele tipo. [Este é o sistema que eu projetei pra você e essa é a maneira pela qual você pode ou deve utilizá-lo de forma que preencha uma variedade de propósitos que coadunam com esta visão]. Você pode usar o calendário para verificar os eventos do curso, clicar sobre um evento e ver sua descrição. Pode também modificar suas opções de visualização na estação de percepção, customizando que tipo de movimentação deseja estar ciente ou que cursos deseja acompanhar a movimentação para estar a par automaticamente da movimentação em seus outros cursos e ainda subdividir seus grupos em subgrupos, conforme a atividade pretendida.

Etapa 4 – conclusão do MIS

A impressão geral sobre a análise dos signos metalinguísticos, estáticos e dinâmicos é focada na navegação por um usuário final de um curso no ColabWeb. No cenário utilizado, os problemas navegacionais concentram-se nas visualizações das atividades. A especificidade da visualização dos grupos é um exemplo do que ocorre também com outros recursos. Aparentemente, o designer do ColabWeb crê que ao usuário clicar em um determinado fórum, ele estaria interessado em acessar todos os itens daquela categoria, não se dando ao trabalho de consultar antes o usuário, se ele gostaria de ver outros fóruns ou voltar para onde estava.

No caso específico de IC-CComputação o curso foi configurado considerando a ordem cronológica dos exercícios, de acordo com os conceitos trabalhados no curso, o que facilitou a organização das atividades e manteve um padrão navegacional constante e coerente.

Os problemas navegacionais encontrados se referem a qualquer curso criado no ColabWeb, constatado através da criação do curso IC-EngSemiotica. Quanto a problemas específicos nas configurações de cursos, como o que ocorre com o curso IC-EngComputação (Figura 3.5). Isto se deve ao fato do ColabWeb herdar do Moodle o amplo esquema de configurações para o professor, o que acarreta em situações muito complicadas.

Para concluir a inspeção, nota-se que a metacomunicação no ColabWeb é eficiente quando o designer de curso é um especialista no uso dessa ferramenta. Porém não é eficiente para os usuários finais (alunos), visto que os cursos podem ser organizados de maneiras diversas. Além disso, o usuário se perde com o esquema navegacional, interrompendo sua atividade para retomá-la do ponto onde havia parado. Os usuários do ColabWeb (alunos de computação) normalmente têm a característica de explorar o ambiente para entender seu funcionamento. Para este tipo de usuário o software, apesar de apresentar problemas navegacionais, adéqua-se ao propósito. Quanto às permissões de acesso a alguns logs, não é nem um pouco efetiva, pois confunde os usuários, além de violar os direitos dos grupos.

Etapa 5 – avaliação qualitativa dos resultados do MIS

A triangulação dos resultados na inspeção do ColabWeb é baseada na análise de um chat de propósito geral em IC-Apoio, em wikis de IC-CComputação, no chat do curso IC-Apoio e fóruns em IC-EngComputação. Apesar de se crer que a entrevista sobre os elementos centrais da inspeção é importante, ainda não foi possível entrevistar os alunos que participaram dos cursos em 2008.1, pois os mesmos não responderam aos emails enviados. Foram acrescentadas, então, entrevistas com os professores, sobre comentários específicos de alguns alunos ou algum item que foi detectado no MIS.

Nos registros do ColabWeb, há uma predominância sobre dificuldades relacionadas à conectividade ou demora associada ao carregamento de páginas do ColabWeb. O comentário a seguir, retirado de um fórum do curso IC-EngComputação, exemplifica este problema:

“Com 300kbs ,a tarde, ta mt complicado de usar o ColabWeb.”

A aluna se refere à velocidade de sua conexão em casa. A equipe de suporte do ColabWeb informou que o problema ocorre no acesso externo à rede metropolitana de pesquisa, uma vez que ficou sujeito a uma grande quantidade desses acessos. O suporte informou também que isto já foi resolvido com a compra de um servidor mais robusto e um pequeno aumento na largura de banda da RNP-AM, onde a UFAM está conectada.

Outro comentário diz respeito ao editor do ColabWeb. De modo geral os editores do ColabWeb não parecem ter um comportamento esperado. Algumas vezes não é possível customizar o texto, conforme reclamação de uma aluna:

“Quando se modifica as cores de um texto, ao invés de mudar somente a parte selecionada, outro pedaço do texto não selecionado aparece de cor diferente”.

A aluna faz uma sugestão:

“Minha opiniao é que trocassem esse editor de texto online. Colocar um mais simples e funcional. Não sei se tem alguém está pretendo usar tantas cores, plano de fundo, smiles(!), acho pouco provável. Um editor de texto a "la bloco de notas" estaria ótimo”.

Sobre o problema de navegação, apontado nas outras etapas do MIS, os professores afirmaram ter recebido muita reclamação por email no início da disciplina, mas depois os alunos se “acostumaram”. Ainda segundo os

professores, os alunos inscritos no curso IC-EngComputação encontraram mais problemas para utilizar o ColabWeb, tanto de localização dos exercícios quanto ao uso dos recursos associados aos exercícios.

Entende-se que os alunos de computação sejam mais acostumados a se adaptarem ao uso de um novo software. Por isso, os problemas navegacionais foram superados, em geral, no início da disciplina. No entanto, eles ocorreram e foram registrados pelos professores. Vale também ressaltar que os alunos do curso IC-EngComputação vivenciaram mais dificuldades em associar recursos a exercícios, o que pode ser constatado em alguns comentários nos fóruns.

O fato de a qualidade da metacomunicação do ColabWeb depender da experiência do designer de curso significa que o sistema é configurável. Como o designer de curso sem experiência, usuário a quem a plataforma Moodle se propõe a auxiliar, não consegue projetar um curso de forma que consiga acompanhar as atividades, utilizando um mínimo de recursos simples e úteis, a metacomunicação do Moodle e, conseqüentemente, do ColabWeb, é muito pobre.

Os designers do Moodle omitem informações importantes e diretrizes para que o usuário se sinta à vontade de fazer um bom projeto de curso e realizar as atividades da maneira desejada. Isto faz com que os designers do Moodle não consigam atingir seu propósito com a tecnologia, que é, conforme mencionado anteriormente, o de auxiliar o professor (designer de cursos) a projetar um curso semi-presencial ou a distância.

3.3.4. Proposta de Melhorias e Adaptações

Ao longo deste artigo chamou-se de problemas navegacionais a forma de navegação e estruturação de páginas web do ColabWeb, que é uma herança do Moodle. Nesta sessão, pretende-se discutir sobre os aspectos relacionados a esse esquema de navegação, procurando alternativas para amenizar seus efeitos.

O Moodle é estruturado em módulos quase independentes, de forma que seja fácil para outros desenvolvedores construírem outros módulos em forma de plugins. Por essa característica, o Moodle é considerado um sistema extensível. Seu esquema navegacional segue a estrutura de diretórios físicos, acompanhando a sequência de páginas visitadas pelo usuário ao entrar no ambiente. Sendo assim, ao entrar no ColabWeb, aparecem somente os cursos onde o usuário pode entrar.

Quando ele seleciona um curso, o designer do sistema que deixá-lo ciente de como fazer para retornar à página anterior, fornecendo um link com o título da página ou uma referência ao seu conteúdo. Começa a ficar problemático quando o usuário começa a selecionar outros links dentro do curso e não consegue retornar para onde estava utilizando o mesmo raciocínio.

Para exemplificar, suponha que um usuário do ColabWeb acesse o fórum específico de um exercício. Caso desista de fazer comentários ou simplesmente quiser retornar para a página anterior, o link imediatamente anterior não irá produzir o resultado esperado, levando-o à lista de todos os fóruns disponíveis. Neste caso, o designer espera que o usuário saiba que o lugar onde ele estava era IC-CComputação, já que era a página principal do curso.

Para alguns usuários a mensagem do designer está clara, mas para muitos pode confundir, pois a semântica da estrutura de navegação que aparece em forma de links diz respeito à forma como o sistema armazena os dados e não segundo do ponto de vista semiótico.

No contexto em que o ColabWeb foi utilizado, aprendizagem de programação, acredita-se que o sistema não necessite de muitas adaptações para o contexto, pois as atividades de grupo e entrega de códigos e relatórios podem ser desenvolvidas sem prejuízo em um ambiente CSCL, como o ColabWeb.

As sugestões de adaptações e/ou extensões que podem ser desenvolvidas para o ColabWeb, de forma que atenda melhor seus usuários precisam estar de acordo com o que é possível modificar no sistema sem modificar muito a estrutura da arquitetura onde ele está, pois modificações mais profundas trazem efeitos colaterais para o sistema, muitas vezes imprevisíveis. Tendo isso em vista, propõem-se algumas sugestões quanto ao contexto no qual o sistema foi inspecionado e outras mais gerais, sobre a estrutura navegacional.

Apesar de serem adequadas ao uso de alunos aprendendo programação, algumas extensões seriam úteis, como uma ferramenta que fizesse um rastreamento na evolução dos códigos dos grupos para que todos (professores, monitores e alunos) pudessem consultar e aprender sobre as estratégias adotadas para cada situação. No curso de IC-CComputação os alunos utilizaram uma ferramenta para submissão de códigos, que armazenava todas as versões em um banco de dados. Visto que já há uma ferramenta desenvolvida para acompanhar a

evolução dos códigos a partir de um banco de dados (Castro, Fuks & Castro, 2008b), isso poderia ser uma extensão aos recursos do ColabWeb.

Uma desvantagem levantada nessa inspeção foi a abrangência de tipos de configuração para o designer do curso. O Moodle apresenta uma lista muito grande de atividades para configurar. O designer de curso menos experiente, aquele professor que resolve utilizar o sistema como apoio à sua disciplina, sem nunca tê-lo utilizado antes, não consegue nem entender o que são todas aquelas opções. Há opções padrão, mas não há uma explicação sobre elas ou sugestões de como organizar seu curso. Normalmente os designers de curso mantêm a configuração padrão porque é aparentemente mais fácil. No entanto, o padrão do Moodle é um curso com todos os recursos possíveis, cujas atividades são organizadas em Tarefas. Cremos que esta é uma das maneiras mais complicadas de se configurar um curso.

O ColabWeb poderia apresentar uma tela anterior à configuração do curso, na qual o designer perguntasse ao usuário qual seu nível de habilidade no uso do sistema. A partir daí, as telas seguintes poderiam apresentar mais ou menos opções de configurações, deixando as outras como padrão. Para aqueles designers de curso com menos experiência seriam apresentadas somente configurações sobre conteúdo, datas e eventos. A interface do curso seria montada com os recursos mais utilizados. Vale ressaltar que isto é possível de ser realizado no ColabWeb, apenas omitindo da tela as informações fornecidas pelo Moodle e fazendo um preenchimento automático das requisições de configuração.

A estrutura de navegação do Moodle não permite ser modificada. No entanto, da mesma forma que as configurações, podem-se omitir os links indesejados da interface com o usuário, acrescentando outro, referenciando assim o desejado. Dessa forma, o problema que ocorre com a visualização de grupos poderia ser amenizado retirando-se o link de grupos, permanecendo somente a palavra que os descreve.

3.4. Conclusão do Capítulo

Neste capítulo discutimos sobre como as ferramentas utilizadas para apoiar a colaboração precisam ser sensíveis ao seu contexto de aplicação e devem possuir algum esquema de codificação das atividades para facilitar as próprias

interações e as intervenções no caso de serem voltadas à aprendizagem. Destacamos que os ambientes CSCL podem ser adaptados e mecanismos auxiliares podem ser incorporados a eles para que apoiem aprendizagem de programação em grupo.

Mostramos relatos sobre o funcionamento de ferramentas e metodologias desenvolvidas para apoiar a aprendizagem de programação em grupo. Os defensores desta modalidade de ensino de programação afirmam que a atividade é, no mínimo, mais prazerosa que o método tradicional, focado no indivíduo, servindo de incentivo para os alunos. As ferramentas apresentadas não são completas no sentido de acompanhar os processos de raciocínio e discussão nos grupos. Por isso a proposta de se utilizar ambientes CSCL, que já possuem mecanismos essenciais à colaboração como base para o desenvolvimento de outros recursos necessários à aprendizagem de programação.

A Seção 3.3 é baseada em (Castro & Fuks, 2009). Ela faz uma inspeção semiótica em um ambiente CSCL chamado ColabWeb, que por sua vez, é uma implementação a partir do Moodle. Essa inspeção foi realizada em 3 cursos que ocorriam simultaneamente no ambiente, voltados à aprendizagem de programação. Foram identificadas dificuldades de navegação e sugeridas modificações para aquele tipo de curso. Uma vez seguidas essas recomendações, o ColabWeb poderia continuar sendo utilizado para apoiar a aprendizagem de programação em grupo.

No próximo capítulo discutimos sobre colaboração na aprendizagem de programação em grupo. Iniciamos com uma revisão na literatura sobre métodos de colaboração, seguida de uma prospecção sobre as características da aprendizagem de programação em grupo e a necessidade de incorporar colaboração às metodologias utilizadas. Terminamos o capítulo com a descrição de um estudo de caso exploratório para identificar as dificuldades encontradas pelos alunos de graduação em computação ao programarem em grupo.