

CONFLITO SÓCIO-COGNITIVO E ESTILOS DE APRENDIZAGEM NA FORMAÇÃO DE GRUPOS PARA O APRENDIZADO COLABORATIVO DE PROGRAMAÇÃO DE COMPUTADORES

Juan Manuel Adán-Coello

Centro de Ciências Exatas Ambientais
e de Tecnologias
PUC - Campinas
juan@puc-campinas.edu.br

Wiris Serafim de Menezes

Secretaria da Fazenda do
Estado de Goiás
wiris-sm@sefaz.go.gov.br

Eustáquio São José de Faria

Instituto de Informática
PUC - Minas - Arcos
eustaquio@pucminas.br

Carlos Miguel Tobar

Centro de Ciências Exatas Ambientais
e de Tecnologias
PUC - Campinas
tobar@puc-campinas.edu.br

Resumo: *A aprendizagem colaborativa tem sido apontada como uma abordagem eficaz para reduzir as dificuldades encontradas por aprendizes de programação de computadores. Em um processo de aprendizagem colaborativa, a formação de grupos é atividade fundamental e uma das mais complexas, pelo fato de não bastar agrupar alunos aleatoriamente para garantir a colaboração. Com base na constatação de que há falta de ferramentas que auxiliem o professor a formar grupos, no contexto do aprendizado colaborativo de programação, foram desenvolvidas as ferramentas Avaliador de Qualidade de Programas (AQP) e GroupOrganizer. O AQP agrupa alunos que desenvolvem programas com diferenças estilísticas significativas, a fim de que se manifeste o conflito sócio-cognitivo. O GroupOrganizer incorpora o AQP e leva também em consideração o estilo de aprendizagem dos alunos. O seu objetivo é que os grupos sejam formados por alunos que apresentam os mesmos estilos de aprendizagem ou estilos complementares. Experimentos usando as ferramentas, envolvendo duas turmas de alunos cursando disciplinas introdutórias de programação em cursos superiores na área de Informática, constaram que as ferramentas facilitam o trabalho dos instrutores na formação de grupos, auxiliam a motivar os alunos e mostram evidências de que as estratégias escolhidas contribuem positivamente para o aprendizado dos alunos.*

Palavras-chave: *Aprendizagem colaborativa, programação de computadores, formação de grupos, conflito sócio-cognitivo, estilos de aprendizagem.*

Abstract: *Collaborative learning has being pointed out as an effective approach to reduce apprentices' difficulties that arise during the effort to learn computer programming. In a collaborative learning process, definition of work groups is a fundamental activity and one of the most complex, because it is ineffective to group students randomly in order to really obtain collaboration. The AQP and GroupOrganizer tools were developed to address the lack of tools that support instructors in forming groups in the context of learning how to program computers through collaboration. The tools are based on the theory of social-cognitive conflict and on learning styles. In order to stimulate the social-cognitive conflict, AQP groups students with significant differences in programming style. GroupOrganizer incorporates AQP and forms groups taking into consideration also students' learning styles. It tries to group students with the same or complementary learning styles. Experiments with two classes of undergraduate students on introductory programming courses provide evidences that the adopted approaches contribute to increase students' learning and interest, and in helping the instructor on the task of forming groups.*

Keywords: *Collaborative learning, computer programming, group definition, social-cognitive conflict, learning styles.*

1. INTRODUÇÃO

O aprendizado de programação de computadores é um dos primeiros e maiores desafios encontrados pelos estudantes da área de computação e informática. As dificuldades encontradas pelos estudantes refletem-se em elevados índices de reprovação nas disciplinas relacionadas com o aprendizado das habilidades de programação e nas disciplinas que dependem diretamente dessas habilidades. Este cenário se deve em parte às dificuldades encontradas pelos docentes para guiar de forma efetiva os estudantes durante a realização das atividades práticas de programação.

A literatura indica que o trabalho colaborativo envolvendo grupos de estudantes pode contribuir para resolver esse problema, desde que sejam usados mecanismos adequados para construir e mediar o trabalho dos grupos. Quando isso é feito, os estudantes são capazes de colaborar efetivamente propiciando melhorias perceptíveis de desempenho e aumento da capacidade de elaborar pensamento crítico [13][35] [12].

O trabalho colaborativo parte do princípio de que dois ou mais indivíduos trabalhando conjuntamente podem chegar a uma situação de equilíbrio, onde as idéias podem ser compartilhadas entre os participantes do grupo, gerando novas idéias e conhecimento [19][27].

Em um ambiente de aprendizado colaborativo, estudantes trabalham juntos em grupo para alcançar um objetivo comum, sendo guiados por um professor ou por um líder do grupo [21]. Para Felder e Brent [9], o aprendizado colaborativo somente se qualifica quando existe interação e comunicação entre os integrantes do grupo. Para que isso ocorra, não basta agrupar os alunos aleatoriamente, é necessário adotar critérios que propiciem que a interação ocorra.

Por ser uma tarefa complexa, usualmente os professores permitem que os próprios alunos constituam os grupos de acordo com seus próprios critérios. Essa abordagem apresenta problemas importantes, decorrentes, principalmente, do fato de que os grupos geralmente são formados levando em consideração laços de amizade ao invés de interesses funcionais ou de aprendizado [30].

A constituição de grupos sem os devidos cuidados pode também levar à formação de grupos compostos somente por alunos fortes ou fracos¹. A desvantagem de grupos que contêm somente alunos fracos é óbvia, mas manter somente alunos fortes em um grupo é igualmente indesejável. Em primeiro lugar, porque o grupo forte tem uma excessiva vantagem sobre os outros grupos. Em segundo lugar por que o grupo tende a dividir o trabalho em partes a serem executadas individualmente, de modo que seus membros se comunicam superficialmente, não ocorrendo as interações dinâmicas que levam aos benefícios do aprendizado colaborativo. Por outro lado, em grupos de habilidades

mistas, os estudantes mais fracos ganham ao ver como os estudantes mais fortes estudam e abordam os problemas, e os estudantes mais fortes ganham maior entendimento sobre o assunto que estão ensinando aos outros [9].

Diversas estratégias têm sido usadas para a constituição de grupos efetivos para o trabalho colaborativo, entre elas podem ser destacadas as baseadas nas teorias neo-piagetianas do conflito sócio-cognitivo e na utilização de estilos de aprendizagem.

1.1. A TEORIA DO CONFLITO SÓCIO-COGNITIVO

A utilização do conflito sócio-cognitivo no processo de aprendizagem é baseada na teoria do equilíbrio de Piaget [28]. O conflito sócio-cognitivo ocorre quando uma interação social estabelece um estado de divergência de opiniões ou apresenta contradição entre soluções pessoais ou coletivas, decorrentes da interação ou trazidas para a mesma. O conflito sócio-cognitivo poder ocorrer, por exemplo, a partir da interação social decorrente da confrontação de soluções computacionais divergentes para um mesmo problema, produzidas pelos sujeitos participantes.

O conflito sócio-cognitivo pode ser induzido compondo grupos de maneira que eles tenham indivíduos com níveis diferentes de desenvolvimento cognitivo. Acredita-se, também, que as interações decorrentes desse conflito levem os indivíduos participantes a um estágio superior de conhecimento [3].

O conflito sócio-cognitivo refere-se à natureza do pensamento coletivo e caracteriza um conflito entre respostas socialmente diferentes, que designam também naturezas cognitivas distintas, porque em cada sujeito a incidência da interação suscita uma resposta interna diferente, produzindo um conflito interno [5]. Esse conflito pode ser positivo, desde que proporcione reestruturação cognitiva em pelo menos um dos participantes da interação [33].

Deve-se frisar que, para o estabelecimento do conflito sócio-cognitivo, é necessário que a interação social apresente conflito de idéias ou opiniões. A interação social só será construtiva se provocar nos sujeitos uma confrontação de soluções e idéias divergentes (ou diferentes).

Uma aplicação interessante das idéias da teoria do conflito sócio-cognitivo em ambientes de apoio ao aprendizado colaborativo é o sistema COLER (Collaborative Learning environment for Entity-Relationship modeling) [4], voltado para o aprendizado colaborativo de modelagem entidade-relacionamento. Nesse ambiente, os alunos desenvolvem as suas tarefas na presença de outros colegas, ou individualmente, se assim o desejarem, e depois as remetem para discussão em pequenos grupos. COLER possui um módulo verificador de diferenças significativas que procura encontrar, entre os modelos construídos individualmente e aquele do grupo, diferenças importantes que definirão se uma solução individual deve ser contrastada com a solução em grupo.

¹ Os termos forte e fraco são usados de forma abrangente para referir-se a alunos que têm mais ou menos conhecimento ou facilidade de aprendizado na área de estudo considerada.

1.2. ESTILOS DE APRENDIZAGEM

Um estilo de aprendizagem corresponde a um conjunto de fatores cognitivos, afetivos e fisiológicos que serve como indicador relativamente estável da forma como um aprendiz percebe, interage e reage ao ambiente de aprendizagem [2].

Desde a década de 60, tem-se procurado descobrir como o ser humano identifica e processa informações. O tema foi abordado originalmente na psicologia, com o propósito de compreender como a informação é processada e a sua influência nos relacionamentos inter-pessoais [25]. Recentemente, esses estudos vêm sendo progressivamente considerados por pesquisadores envolvidos em práticas educacionais.

Para Felder, os estilos de aprendizagem são preferências características e dominantes na forma como as pessoas recebem e processam informações, passíveis de desenvolver [8].

Diversos métodos foram propostos para determinar o estilo de aprendizagem de um indivíduo, entre eles destacam-se o *Myers Briggs Type Indicator* [26], o *Index of Learning Styles* [10] e o *Learning Style Inventory* [17]. Apesar das diferenças existentes entre eles, todos buscam determinar o estilo de aprendizagem predominante em um indivíduo, para entender como ele assimila melhor as informações recebidas.

Em um ambiente de ensino-aprendizagem, conhecendo os estilos de aprendizagem dos alunos e agrupando-os adequadamente, pode-se melhor explorar as habilidades do grupo e minimizar os atritos que ocorrem devido às diferenças individuais. O conhecimento dos estilos de aprendizagem dos alunos permite também que o professor direcione suas aulas, de forma a que os alunos tenham melhor aproveitamento.

A literatura apresenta diversos sistemas para apoiar a formação de grupos, por exemplo [4][15][30]. A maioria deles utiliza critérios que têm o objetivo de permitir a formação de grupos efetivos, evitando que os grupos sejam criados de maneira aleatória ou pelos próprios alunos. Entretanto, quando se trata da formação de grupos no contexto do aprendizado colaborativo de programação de computadores, a literatura é extremamente pobre.

Em vista disso, foram desenvolvidas duas ferramentas para a constituição de grupos voltados ao aprendizado colaborativo de programação de computadores. A primeira, denominada de AQP (Avaliador de Qualidade de Progrmas), apresentada na seção 2, visa a criar grupos de modo a que se manifeste o conflito sócio-cognitivo. A segunda, chamada de GroupOrganizer, discutida na seção 3, acrescenta a possibilidade de considerar também os estilos de aprendizagem dos alunos como critério de formação de grupos. Ambas seções apresentam e discutem experimentos conduzidos para validar as ferramentas e buscar indícios de que as estratégias empregadas são efetivas em termos do aprendizado das

habilidades de programação e de trabalho em grupo desejadas. Após a apresentação das ferramentas e dos experimentos realizados, a seção 4 encerra o artigo fazendo algumas considerações finais e indicando possíveis direções para o aprimoramento das ferramentas desenvolvidas.

2. FORMAÇÃO DE GRUPOS A PARTIR DA TEORIA DO CONFLITO SÓCIO-COGNITIVO

De forma semelhante ao sistema COLER, a aplicação da teoria do conflito sócio-cognitivo, no contexto do aprendizado colaborativo de programação, pode partir da identificação de diferenças significativas entre os programas desenvolvidos pelos alunos. Por exemplo, usando critérios que permitam quantificar a qualidade dos programas produzidos pelos alunos. As diferenças entre os programas devem ser importantes o suficiente para que os alunos se sintam instigados a discuti-las. Ou seja, devem ter um significado que permita que os alunos envolvidos na discussão tenham algum ganho de conhecimento advindo do eventual conflito resultante de comparar e contrastar diferentes soluções propostas para um mesmo problema.

A qualidade de um programa de computador está diretamente ligada ao uso adequado de técnicas de organização estrutural e lógica em sua construção. Não se costuma compreender como de boa qualidade um programa que simplesmente retorna um resultado correto. Além desta característica essencial, ele deve possuir outros atributos que reflitam a sua qualidade [34].

Vários métodos e sistemas para avaliar programas são encontrados na literatura, a maioria dá ênfase à determinação de atributos estilísticos [1] [14] [31] [24] [38] [16] [29]. Esses atributos estão relacionados, principalmente, com a legibilidade do código fonte do programa. Eles procuram indicar se o código é de fácil entendimento para um leitor. Atributos de qualidade são empiricamente conhecidos por programadores experientes, mas difíceis de ensinar a iniciantes, já que exigem muita abstração e prática, além de geralmente não estarem formalmente fundamentados [37].

A propósito da importância da legibilidade de um programa, convém lembrar que o código fonte de um programa é lido, revisado e alterado por diferentes pessoas em diferentes épocas, para realizar diferentes tarefas. Poucas vezes o código fonte será processado por compiladores. Contrariamente à intuição geral, o código fonte é redigido primordialmente para ser lido por pessoas e não para ser processado por compiladores.

A fim de determinar a qualidade de um programa, foi construído o sistema Avaliador de Qualidade de Programas (AQP). A linguagem alvo foi C, largamente usada em cursos introdutórios de programação. O AQP avalia as sete métricas de estilo mais encontradas na literatura: tamanho de identificadores; percentual de identificadores que são

constantes (*define / const*); tamanho dos módulos; quantidade de módulos; percentual de linhas endentadas; percentual de linhas de comentário e percentual de linhas em branco.

O AQP avalia um programa atribuindo uma pontuação para cada uma das sete métricas consideradas. A soma das pontuações atribuídas a cada métrica corresponderá ao Índice Global de Qualidade (IGQ) do programa. Para cada programa submetido, o AQP exibe um relatório que indica o valor do IGQ, os valores atribuídos a cada métrica e recomendações para melhorar o programa em cada uma das dimensões consideradas.

O cálculo da pontuação atribuída a cada métrica considerada é feito usando o método adotado por Berry e Meekings [1] e Hung, Kwork e Chan [14]. Para cada atividade a ser avaliada, o instrutor define um conjunto mestre de valores máximos e mínimos para cada métrica. Dentro do intervalo definido pelos valores máximo e mínimo aceitáveis, define-se uma faixa de valores ideais. Ao avaliar um programa, valores de atributos dentro da faixa ideal correspondem à pontuação máxima para a métrica (100 pontos). Atributos com valores entre o máximo e o mínimo, mas fora do intervalo ideal, recebem uma pontuação proporcional. Valores menores que o limite mínimo e maiores que o máximo são pontuados com zero. Detalhes de como é feito o cálculo do valor a atribuir a cada métrica podem ser encontrados em [6].

O AQP oferece uma interface que permite ao professor organizar e formar grupos a partir dos programas submetidos pelos estudantes, e aos estudantes submeter programas e examinar os relatórios de avaliação dos programas. Detalhes sobre a interface e arquitetura do AQP podem ser encontrados em [7].

2.1. EXPERIMENTO 1: FORMAÇÃO DE GRUPOS A PARTIR DA COMPARAÇÃO DOS PROGRAMAS DOS ESTUDANTES

Nesta seção são apresentados e discutidos os principais aspectos associados à montagem, condução e resultados de um experimento realizado para avaliar e validar o AQP. São também buscadas evidências de ganhos de aprendizado dos alunos decorrentes da sua utilização, focando o aprimoramento do estilo de programação dos estudantes. Maiores detalhes sobre o experimento podem ser obtidos em [6] e [7].

O experimento envolveu uma turma do 2º semestre de Sistemas de Informação da PUC Minas – Campus Arcos, cursando a disciplina Algoritmos e Estruturas de Dados. Foram montados seis grupos de quatro alunos, enquanto oito alunos trabalharam individualmente, perfazendo um total de 32 alunos.

Os alunos realizaram três tarefas de programação com complexidade crescente. A primeira tarefa foi feita individualmente. Os programas produzidos foram avaliados pelo AQP e a partir dessa avaliação foram montados grupos para refazer a mesma tarefa. A segunda e terceira tarefas seguiram o mesmo ciclo da primeira, mas respeitando os grupos montados durante a primeira tarefa.

Dos 32 alunos que realizaram a primeira tarefa, oito obtiveram IGQ alto, (maior ou igual a 420 pontos, ou 60% da pontuação máxima), oito obtiveram IGQ médio (entre 210 e 420) e 16 IGQ baixo (inferior a 210, ou 30% da pontuação máxima).

Foram montados dois grupos heterogêneos (grupos 1 e 2 – formados por dois alunos com IGQ alto e dois alunos com IGQ baixo), um grupo homogêneo com IGQ alto (grupo 3), um grupo homogêneo com IGQ baixo (grupo 4) e dois grupos homogêneos com IGQ médio (grupos 5 e 6). Além dos grupos, oito alunos com IGQ baixo trabalharam individualmente em todas as tarefas. Não houve alunos com IGQ alto e médio trabalhando individualmente, devido ao número relativamente reduzido de alunos com essas pontuações e ao particular interesse em focar o estudo nos alunos com maiores dificuldades.

O processo de colaboração aconteceu no laboratório de programação onde se realizam as aulas práticas da disciplina. Após a avaliação dos programas produzidos individualmente, solicitou-se aos alunos de cada grupo que trocassem os relatórios e os programas entre si e procurassem identificar as deficiências do programa de um dos colegas. Feito isso, cada aluno expôs ao grupo as possíveis deficiências encontradas no programa que analisou. Em seguida, o líder de grupo, escolhido por sorteio e instruído pelo professor para intermediar a colaboração entre os colegas, conduziu a construção de um novo programa. Esse programa foi também avaliado pelo AQP e um novo relatório foi gerado para que os alunos e o professor pudessem comparar o programa produzido pelo grupo com os programas escritos por seus membros. Solicitou-se aos alunos que comentassem as diferenças entre o seu programa e o do grupo naqueles aspectos em que havia diferenças importantes. Após analisar o relatório produzido pelo AQP, os alunos refizeram a tarefa individualmente e a submeteram ao AQP. Este ciclo repetiu-se para as tarefas 2 e 3.

A partir das avaliações armazenadas no AQP, comparou-se a evolução dos IGQ para os programas produzidos individualmente e em grupo entre as tarefas 1 e 3, conforme discutido na seção 2.3.

2.2. REQUISITOS DOS PROGRAMAS CONSTRUÍDOS

A tarefa 1 consistiu na construção de um programa para ler os dados de uma lista telefônica (10 telefones) e, após a leitura da lista, oferecer ao usuário um menu com as seguintes opções: (1) procurar um telefone por nome de assinante; (2) procurar um assinante por número de telefone; (3) sair.

A tarefa 2 consistiu na construção de um programa para ler, ordenar e mostrar os dados de N alunos (constante com valor 100) cursando uma disciplina, usando o método bolha de ordenação em memória primária. O usuário deveria ser guiado por um menu com as seguintes opções: (1) inserir um aluno, (2) ordenar o vetor de alunos por ordem de nome, (3) ordenar o vetor de alunos por ordem de nota final, (4) mostrar todos os alunos aprovados, (5) mostrar todos os alunos e (6) sair.

A tarefa 3 requeria a construção de um programa para controlar uma lista dinâmica de convidados para uma festa. O usuário deveria dispor de um menu com as seguintes opções: (1) inserir um convidado em ordem de nome na lista, (2) retirar um convidado da lista (digitar o nome do convidado e procurá-lo na lista), (3) mostrar a quantidade de convidados, (4) mostrar todos os convidados, (5) mostrar todos os convidados da cidade de Arcos e (6) sair.

2.3. RESULTADOS DOS EXPERIMENTOS

A tabela 1 sintetiza os resultados dos experimentos. As colunas AI1, AI2 e AI3 mostram os IGQ médios dos programas produzidos individualmente pelos membros dos grupos para as atividades A1, A2 e A3, respectivamente. De forma análoga, as colunas AG1, AG2 e AG3 mostram os IGQ dos programas produzidos pelos grupos para cada uma das três atividades. A tabela 1 não mostra a evolução média dos alunos com IGQ inicial baixo que trabalharam individualmente, que foi de 27%.

Embora não mostrados na tabela 1, cabe ressaltar que analisando os resultados detalhados dos grupos

heterogêneos, chamou a atenção a evolução média de 518% do IGQ dos alunos com pontuação inicial baixa que fizeram parte desses grupos. Sensivelmente maior que a média dos elementos do grupo, que foi de 66% (primeira linha na tabela 1). Maior também que a dos alunos do grupo homogêneo-baixo (de 210%, conforme a quarta linha da tabela 1) e que a dos alunos que trabalharam individualmente (de 27%, não apresentada na tabela 1). A pontuação dos membros dos grupos heterogêneos que tiveram IGQ inicial alto teve uma evolução média de 20%, superior aos 8% verificados no grupo homogêneo-alto (terceira linha da tabela 1).

Os bons resultados observados no grupo heterogêneo são consistentes com a literatura, que indica que grupos com membros heterogêneos costumam ser mais efetivos em termos de aprendizado individual que grupos homogêneos, porque a heterogeneidade naturalmente gera controvérsias mais frequentes [20]. Todavia, notou-se que o grupo homogêneo com IGQ baixo também apresentou bons índices de crescimento individual (210%, conforme a quarta linha da tabela 1).

Tabela 1: Evolução do IGQ nas atividades individuais.

Grupo	IGQ (0 .. 700)						Evolução Percentual	
	AI1	AG1	AI2	AG2	AI3	AG3	AI1→AI3	AG1→AG3
1 - Heterogêneo	266	532	330	570	443	600	66	13
2 - Heterogêneo	264	513	414	567	472	625	79	22
3- Homogêneo-Alto	436	642	418	533	473	667	8	4
4- Homogêneo-Baixo	112	440	281	425	348	433	210	-2
5- Homogêneo-Médio	356	614	389	533	437	550	23	-10
6- Homogêneo-Médio	328	617	356	525	383	433	17	-29

Comparando a pontuação dos programas escritos pelos componentes dos grupos com a pontuação dos programas construídos pelos grupos, nota-se que as notas dos grupos, salvo raras exceções, foram expressivamente superiores às notas individuais. Isto era esperado, especialmente porque os grupos reimplementaram programas que já haviam sido escritos e discutidos.

Os programas dos grupos heterogêneos apresentaram crescimentos de IGQ superiores aos dos demais grupos, embora a diferença não seja tão expressiva como a verificada para a evolução dos IGQ dos seus membros. Um dado que chama a atenção, e que merece ser melhor estudado, é a redução do IGQ médio verificado nos programas dos grupos homogêneo-baixo e homogêneo-médio (respectivamente de -2%, -10% e -29%).

2.4. OBSERVAÇÕES SOBRE A DINÂMICA DO TRABALHO EM GRUPO

Os experimentos realizados evidenciaram que o trabalho colaborativo resultou em programas mais compreensíveis. Além

de notar que os IGQ dos alunos melhoravam de tarefa para tarefa, o professor verificou que os programas estavam cada vez mais próximos dos padrões de legibilidade esperados. A colaboração fez com que, na maioria dos experimentos, a nota obtida pelo grupo fosse superior à maior nota individual dos componentes do correspondente grupo.

Embora os componentes com IGQ inicialmente baixos dos grupos heterogêneos tenham apresentado os maiores índices de crescimento individual, percebeu-se uma grande resistência nesses grupos para o trabalho colaborativo. Outro problema neles detectado foi a submissão por parte dos alunos que possuíam IGQ baixo. Esta submissão foi quebrada somente quando o professor se juntou ao grupo e estimulou a participação de todos os componentes.

Acompanhando os grupos de perto, percebeu-se que os alunos colaboraram mais para a construção do programa nos grupos homogêneos. Foi observado que nesses grupos o conflito aconteceu de maneira mais equilibrada na maioria das tarefas.

Embora novos experimentos sejam necessários para confirmar os resultados aqui relatados, os experimentos indicam que a colaboração teve efeitos positivos no aprendizado de programação. Entretanto, também se observou que a figura do mediador foi fundamental. Quando ele não estava presente, a colaboração tendia a não ocorrer ou ocorrer precariamente. Em grupos heterogêneos, na falta do mediador, os alunos mais fortes tendiam a querer impor as suas posições aos mais fracos. Nos grupos homogêneos médios, a colaboração fluía normalmente na ausência do mediador. Já no grupo homogêneo alto, a ausência do mediador transformava o ambiente em caos generalizado, pois cada membro do grupo acreditava que sua idéia deveria prevalecer. Por fim, no grupo homogêneo baixo, na ausência do mediador, os alunos se tornavam apáticos, sem saber por onde iniciar a discussão.

3. FORMAÇÃO DE GRUPOS A PARTIR DA TEORIA DO CONFLITO SÓCIO-COGNITIVO E DO USO DE ESTILOS DE APRENDIZAGEM

O GroupOrganizer é um sistema acessível através da Web, que permite formar grupos combinando elementos da teoria do conflito sócio cognitivo e estilos de aprendizagem. Detalhes da interface oferecida aos usuários (professor e alunos) bem como da sua arquitetura e implementação podem ser encontrados em [22] e [23].

A fim de que o conflito sócio-cognitivo se manifeste, o GroupOrganizer agrupa alunos que desenvolveram programas com diferenças significativas, utilizando os IGP dos programas produzidos pelos alunos, computados pela ferramenta AQP. Com isso, ao discutir sobre as diversas versões de programas apresentados, os alunos poderão perceber outras soluções para um mesmo problema e aprender a partir da controvérsia e da troca de conhecimento.

Para identificar os estilos de aprendizagem dos alunos, o sistema utiliza o LSI de Kolb e Kolb [17]. O objetivo é agrupar alunos com os mesmos estilos ou estilos complementares, na tentativa de evitar conflitos indesejáveis, geralmente ocorridos devido à existência de estilos de aprendizagem incompatíveis dentro do grupo.

Parece contraditório querer provocar o conflito sócio-cognitivo ao mesmo tempo em que se agrupam alunos com estilos de aprendizagem iguais ou complementares. A idéia, no entanto, é agrupar alunos que apresentaram diferentes soluções para os programas, mas que possam discutir essas diferenças e chegar a uma solução de consenso de uma forma proveitosa. Agrupar alunos com estilos de aprendizagem considerados incompatíveis pode gerar brigas e insatisfações, prejudicando a busca de uma solução de consenso. Sharp [32] constatou que pensamentos diferentes podem provocar uma sinergia especial complementando-se, entretanto, um membro de um grupo com um determinado estilo de aprendizagem pode muitas vezes reprimir ou frustrar outros membros do grupo com estilos diferentes.

3.1. DETERMINAÇÃO DOS ESTILOS DE APRENDIZAGEM DOS ALUNOS

Apesar de todos os modelos de estilos de aprendizagem mencionados na seção introdutória já terem sido empregados com sucesso, o Learning Style Inventory (LSI) foi adotado por já ter sido empregado com sucesso na área da Ciência da Computação – por exemplo, no estudo da usabilidade de softwares e no treinamento de usuários – e em áreas afins [36] [11] [32] [18].

O LSI baseia-se na Teoria da Aprendizagem Experimental, que define a aprendizagem como “o processo pelo qual o conhecimento é criado através da transformação da experiência”. Para Kolb e Kolb, o ser humano se defronta frequentemente com novas experiências e situações de aprendizagem em diferentes esferas de sua vida. Com o passar do tempo, o indivíduo tende a se tornar melhor em alguns dos passos do processo de aprendizagem do que em outros e como resultado desenvolve um estilo próprio de aprendizagem.

O LSI toma como base o modo pelo qual as pessoas aprendem ou preferem aprender. Neste contexto, aprender é o processo através do qual o conhecimento está em constante transformação, que depende de um contínuo “captar” e “transformar”, sendo continuamente criado e recriado, e não uma entidade independente a ser adquirida ou transmitida. O captar pode ser visto sob dois ângulos: o contato direto com o mundo exterior através dos sentidos, mediante o envolvimento com novas experiências, denominado por Kolb e Kolb de Experiência Concreta (EC), ou através de uma representação figurativa, criando os conceitos que integram as observações, denominado Conceituação Abstrata (CA). O transformar é visto sob dois ângulos: a introversão, que compreende refletir e observar as experiências a partir de várias perspectivas, que Kolb e Kolb denominam Observação Reflexiva (OR), e a extroversão, que envolve usar as teorias para tomar decisões e resolver problemas, denominada Experimentação Ativa (EA).

Da combinação entre EC e OR, é obtido o estilo DIVERGENTE, caracterizado por estudantes criativos, geradores de alternativas, capazes de analisar as situações sob muitas perspectivas e relacioná-las como um todo organizado, mas que apresentam um pouco de dificuldade na tomada de decisões.

Da combinação entre CA e OR, é obtido o estilo ASSIMILADOR, caracterizado por estudantes com um excelente raciocínio indutivo e habilidade para criar modelos abstratos e teóricos, mas que necessitam de tempo para refletir. Interessam-se mais pelo aspecto lógico de uma idéia do que pelo seu uso prático.

Da combinação entre CA e EA, é obtido o estilo CONVERGENTE, caracterizado por estudantes que têm como ponto forte a aplicação prática das idéias, com capacidade para definir bem os problemas e tomar decisão com rapidez, que aprendem por tentativa e erro.

Da combinação entre EA e EC, é obtido o estilo ACOMODADOR, caracterizado por estudantes que se

adaptam bem às circunstâncias imediatas e que gostam de riscos, aprendem fazendo e aceitando desafios e apóiam-se nos outros para a busca de informações.

Estudantes com os estilos CONVERGENTE e DIVERGENTE tendem a ser complementares, apesar de haver oposição de idéias entre eles. De modo análogo, os estilos ACOMODADOR e ASSIMILADOR também são considerados complementares [32].

O LSI consiste de um questionário com 12 questões, cada uma com quatro alternativas. Cada alternativa é classificada pelo aprendiz de um a quatro, sendo que quatro corresponde à forma como ele melhor aprende e um corresponde à forma que ele considera pior para aprender. Baseado nas respostas, o LSI utiliza seis variáveis para calcular a pontuação que identifica o estilo predominante da pessoa. Os detalhes associados à determinação do estilo de aprendizagem podem ser encontrados em [17].

3.2. EXPERIMENTO 2: FORMAÇÃO DE GRUPOS COMBINANDO

CONFLITO SÓCIO-COGNITIVO COM ESTILOS DE APRENDIZAGEM

A fim de validar a ferramenta GroupOrganizer, foi realizado um novo experimento durante as aulas práticas da disciplina Programação I, oferecida a alunos do primeiro ano de Licenciatura Plena em Informática no campus da Universidade Federal do Mato Grosso (UFMT), situado na cidade de Rondonópolis. Os experimentos contaram com a participação de 27 alunos, orientados pelo professor responsável pela disciplina, tendo sido desenvolvidos ao longo de 7 semanas. O experimento teve dois objetivos: (1) verificar em que medida a ferramenta auxilia o professor na tarefa de formação de grupos e (2) buscar evidências de que o trabalho em grupos formados usando as abordagens adotadas leva a ganhos de aprendizado.

Ao apresentar o experimento, a professora disse aos alunos que a ferramenta seria adotada com o intuito de auxiliar a aprendizagem e, conseqüentemente, diminuir o índice de reprovação na disciplina e não para avaliar programas ou alunos.

Os alunos responderam ao questionário LSI e observaram o resultado e as descrições do seu estilo de aprendizagem e do estilo complementar apresentadas pelo sistema. No debate que se seguiu, foi possível perceber que os alunos se identificaram com o estilo de aprendizagem encontrado, sendo que alguns afirmaram possuir também um pouco do estilo de aprendizagem complementar. A turma mostrou-se bastante diversificada com nove alunos do tipo ACOMODADOR, seis do tipo ASSIMILADOR, oito do tipo CONVERGENTE e sete do tipo DIVERGENTE.

3.2.1. ATIVIDADES REALIZADAS

Após obter os estilos de aprendizagem dos alunos, foram realizadas duas atividades preliminares com o objetivo de familiarizá-los com a ferramenta. A primeira atividade, denominada de AI1, foi realizada individualmente e a segunda, denominada de AG1, foi desenvolvida em grupo.

Após a fase preliminar, iniciou-se a fase de avaliação, composta por cinco atividades, três individuais: AI2, AI3 e AI4; e duas em grupo: AG2 e AG3. As atividades foram realizadas na seguinte ordem: AI2, AG2, AI3, AG3 e AI4. A intercalação de uma atividade individual com uma em grupo teve por objetivo verificar se houve evolução no aprendizado dos alunos após cada atividade em grupo.

O IGQ obtido na atividade AI2 foi usado para a formação dos grupos para as próximas atividades. Essa atividade solicitava a construção de um programa para efetuar a soma e a subtração de dois valores utilizando funções. Os IGQ dos programas ficaram entre 200 a 500, atingindo uma média geral de 378, com desvio padrão de 75,7.

A partir dos IGQ obtidos na atividade AI2 e dos estilos de aprendizagem dos alunos, foram criados grupos com três tipos de formação: tipo 1, grupos com alunos de estilos de aprendizagem homogêneos e IGQ heterogêneos; tipo 2, grupos com alunos com IGQ heterogêneos, sem considerar os seus estilos de aprendizagem; e tipo 3, considerando somente os estilos de aprendizagem, consistindo de grupos com alunos de estilos iguais ou complementares.

Após a formação dos grupos, foi realizada a atividade AG2, que consistiu em escrever um programa para ler um vetor A com 30 números inteiros e gerar um vetor B de acordo com as seguintes regras: (a) se o elemento de A for par, armazená-lo em B após multiplicá-lo por 5; (b) se o elemento de A for ímpar, armazená-lo em B após somar-lhe 5. Além disso, pediu-se o uso da diretiva `#define` e a utilização de funções para ler os valores, gerar o vetor B e exibir os dois vetores.

Na atividade AI3, solicitou-se aos alunos que elaborassem um programa para calcular a potência de um número inteiro, seguindo as seguintes diretrizes: (a) solicitar ao usuário um valor inteiro para a base e outro para o expoente, (b) calcular a potência do número usando uma função e (c) criar um laço de repetição que permita ao usuário solicitar o cálculo da potência de quantos números desejar.

O objetivo dessa atividade foi verificar se após o trabalho em grupo houve melhora no desempenho individual. Conforme pode ser visto na tabela 2, para a maioria dos grupos, o IGQ médio na AI3 foi menor que o obtido na AI2. Essa redução de desempenho talvez possa ser atribuída à maior dificuldade da atividade, já que essa foi a primeira atividade com o emprego do conceito de função em programação.

Na atividade AG3, solicitou-se aos grupos que elaborassem um programa para calcular o valor total de uma compra. O programa deveria ler o código de cada produto escolhido pelo cliente e a quantidade desejada, calculando o total a ser pago, até o usuário pressionar uma tecla para cancelar ou para finalizar a compra.

A atividade AI4 finalizou os experimentos. Solicitou-se aos alunos para elaborar uma função que recebesse como parâmetros vetores contendo o sexo e o estado civil de um grupo de 10 pessoas e imprimisse a quantidade e a porcentagem de homens solteiros.

Tabela 2: Evolução do IGQ nas atividades individuais. Formação: Tipo 1 = estilos homogêneos e igq heterogêneos; Tipo 2 = igq heterogêneos; Tipo 3; estilos homogêneos ou complementares.

Tipo de Formação	Grupo	IGQ Médio (0 .. 700)			Evolução Percentual do IGQ Médio			
		AI2	AI3	AI4	AI2→AI4	AI3→AI4	AI2→AI4	AI3→AI4
1	1	300	250	516,7	72,2	106,7	36,2	45,4
	2	379,6	379,7	506,6	33,4	33,4		
	3	387,3	370	430	11	16,2		
2	4	380	346,7	370	-2,6	7,6	-7	9
	5	445,3	383,3	446,6	0,3	16,5		
	6	383,3	300	306,6	-20	2,2		
3	7	338	285	475	40,5	66,6	24,4	33,6
	8	393	300	430	9,4	43,3		
	9	379	333,3	440	16,1	34,6		
	10	333	425	450	35,1	5,9		

Na tabela 3 observa-se que a evolução do IGQ médio dos grupos com formação do tipo 1 (estilo de aprendizagem homogêneo) teve uma evolução média de 20,1%, enquanto os grupos com os demais tipos de formação apresentaram uma pequena redução de IGQ.

Tabela 3: Evolução do IGQ nas atividades em grupo. Formação: Tipo 1 = estilos homogêneos e IGQ heterogêneos; Tipo 2 = IGQ heterogêneos; Tipo 3; estilos homogêneos ou complementares.

Tipo de Formação	Grupo	IGQ (0 .. 700)		Evolução Percentual do IGQ	
		AG2	AG3	AG→AG3	
1	1	470	450	-4	20,17
	2	520	500	-4	
	3	200	480	140	
2	5	250	360	44	-7,69
	6	530	380	-28	
	7	550	460	-16	
3	8	490	400	-18	-2,3
	9	400	400	0	
	10	400	400	0	
	11	450	500	11%	

As tabelas 2 e 3 mostram que entre as atividades AI2 e AI4 houve uma maior evolução no desempenho dos grupos da formação do tipo 1, seguida pelos grupos da formação do tipo 3 e por fim pelos grupos da formação do tipo 2. Constatou-se também que entre os dez alunos que tiveram maior evolução

de AI2 para AI4 há seis alunos pertencentes aos grupos com formação do tipo 1, quatro alunos pertencentes aos grupos com formação do tipo 3 e nenhum aluno pertencente aos grupos com formação do tipo 2. Ou seja, 66,6% dos alunos que obtiveram essa evolução pertencem aos grupos de formação do tipo 1. Resultados semelhantes se repetem ao comparar as atividades AI4 e AI3.

Na classificação dos dez alunos que tiveram menor evolução na atividade AI4 em relação à atividade AI2, há dois alunos pertencentes aos grupos da formação do tipo 1, dois alunos pertencentes aos grupos da formação do tipo 3 e seis alunos dos grupos da formação do tipo 2. Ou seja, 22,2% dos alunos integrantes dos grupos de formação do tipo 1 estão dentro dessa classificação. Resultados semelhantes se repetem ao comparar as atividades AI4 e AI3.

A análise dos resultados apresentados sugere que a formação de grupos combinando as abordagens da teoria do conflito sócio-cognitivo e dos estilos de aprendizagem apresenta melhores resultados do que o uso isolado de cada uma dessas abordagens.

3.2.2. OBSERVAÇÕES SOBRE A DINÂMICA DE TRABALHO

Nesta seção são apresentadas algumas observações consideradas relevantes sobre a dinâmica do trabalho individual e em grupo feitas pela professora que conduziu o experimento.

Na medida em que os alunos habituavam-se à interface da ferramenta, observou-se um aumento do interesse pelas atividades. Percebeu-se que os alunos se esforçavam para aperfeiçoar o código com o intuito de obter melhores IGQ. Quando o índice não correspondia à expectativa, os alunos questionavam a professora e pediam que o código fosse analisado por ela, de modo a verificar de que forma o índice poderia ser melhorado na próxima atividade.

Examinando os programas produzidos pelos alunos, pôde-se constatar que o processo de elaborar o programa, submetê-lo ao GroupOrganizer, analisar de forma crítica o resultado e aplicar o conteúdo aprendido no próximo programa proporcionou uma melhoria perceptível na estrutura do código, principalmente no que tange à endentação e ao tamanho dos identificadores.

Mesmo quando não foram obtidos IGQ elevados, foi perceptível a evolução dos alunos. À medida que os programas eram analisados pela ferramenta e os alunos discutiam o que poderia ser melhorado, eles compreendiam melhor algumas boas práticas de programação. Isto tornou mais fácil para o professor avaliar os programas e requerer que fossem utilizadas as práticas de programação recomendadas.

Em relação à comunicação interpessoal, os grupos 3, 4, 5 e 6 foram os que apresentaram maiores dificuldades. Durante as atividades individuais, os alunos que apresentaram maiores dificuldades para realizar as tarefas foram os pertencentes aos grupos onde houve pouca interação. Em contrapartida, nos grupos onde as atividades foram desenvolvidas de forma mais dinâmica e houve uma boa comunicação entre os membros, as atividades individuais foram resolvidas com maior facilidade.

Em síntese, ao desenvolver os programas em conjunto, testá-los, enviá-los para a GroupOrganizer e em seguida analisar os resultados da avaliação produzida, os alunos puderam compartilhar idéias e conhecimento, trabalhando efetivamente de maneira colaborativa.

4. CONCLUSÃO

O trabalho apresentado neste artigo tem o objetivo de preencher uma lacuna existente na oferta de ferramenta para auxiliar a constituição de grupos para trabalho colaborativo no contexto do aprendizado de programação. Para isso, foram desenvolvidas os sistemas AQP e o GroupOrganizer, baseados em elementos da teoria do conflito sócio-cognitivo e de estilos de aprendizagem.

Os experimentos relatados indicam que o AQP e o GroupOrganizer contribuíram efetivamente com o trabalho do professor na definição de grupos e auxiliaram os alunos a evoluir tanto em termos do aprendizado de programação como de competência para trabalho em grupo. Os experimentos, em primeira análise, confirmam que a formação de grupos utilizando a combinação das teorias do conflito sócio-cognitivo e estilos de aprendizagem é benéfica para o aprendizado. Entretanto, mais experimentos têm que ser realizados para confirmar e refinar os resultados verificados, melhor entender os efeitos do uso dessas estratégias e buscar novos elementos para a melhoria do sistema desenvolvido.

Por exemplo, ao comparar os resultados dos dois experimentos realizados, chama a atenção que no primeiro, onde a montagem de grupos foi feita usando apenas a comparação dos programas produzidos, foram observados ganhos de aprendizado sensivelmente maiores que os verificados no segundo experimento, onde se utilizou

também o estilo de aprendizagem dos alunos. Explicar esses resultados não é trivial, porque, apesar de se tentar conduzir o segundo experimento nos mesmos moldes do primeiro, isso não foi de todo possível. Os experimentos foram realizados com alunos de instituições diferentes, acompanhados por professores diferentes e em estágios distintos de sua formação. No caso do primeiro experimento, os estudantes já tinham conhecimento básico de algoritmos e programação de computadores, enquanto que no segundo, os alunos estavam cursando a disciplina de introdução à programação de computadores.

Diversas possibilidades são vislumbradas para estender o Avaliador de Qualidade de Programas (AQP) e o GroupOrganizer. Uma primeira extensão para o AQP seria poder avaliar programas escritos em outras linguagens de programação, além de C, por exemplo Java, atualmente também muito utilizada em cursos introdutórios de programação.

As métricas empregadas para avaliar e comparar programas também podem ser estendidas e melhoradas. Por exemplo, na versão atual, a semântica dos identificadores não é levada em conta, apenas o seu tamanho. Embora identificadores longos sejam usualmente mais expressivos que identificadores curtos, a rigor não há uma relação direta entre o tamanho de um identificador e o seu significado semântico. Ontologias poderiam ser usadas para verificar se um determinado identificador é significativo no domínio do problema sendo tratado. Por outro lado, na versão atual, todas as métricas têm o mesmo peso para o cálculo do Índice Global de Qualidade de um programa (IGQ). Entretanto, programadores experientes sabem que nem todas contribuem igualmente para produzir programas legíveis, embora não seja evidente qual a importância relativa de cada uma.

Além da informação sobre estilo de programação e estilo de aprendizagem, o perfil do estudante poderia incluir outros elementos que ofereceriam mais elementos para constituir grupos com habilidades mistas e complementares, por exemplo, horários disponíveis para estudo fora de sala de aula, experiência prévia com computadores e conhecimento de lógica de programação.

Os experimentos, especialmente o primeiro, indicam que a efetividade do trabalho em grupo, em muitos casos, depende de uma boa mediação das atividades do grupo. Dadas as dificuldades que um instrutor encontra para acompanhar adequadamente o trabalho de turmas com muitos grupos, seria de grande utilidade a criação de um sistema que integrasse as ferramentas já desenvolvidas a outras que permitam a comunicação entre os membros do grupo, a construção colaborativa de programas e a mediação das interações entre os membros do grupo.

AGRADECIMENTOS

À professora Neiva Coelho, da Universidade Federal do Mato Grosso do Sul, Rondonópolis, pelo empenho na condução do experimento 2.

REFERÊNCIAS

- [1] R. E. Berry, B. A. E. Meekings, A Style Analysis of C Programs. *Communications of the ACM*, 28(1):80–88, 1985.
- [2] B. L. Brown, Learning Styles and Vocational Education Practice. 1998. <http://www.calpro-online.org/ERIC/docgen.asp?tbl=pab&ID=74>, Jan. 2008.
- [3] C. Buchs, F. Butera; G. Mugny; C. Darnon, Conflict elaboration and cognitive outcomes. *Theory Into Practice*, 43(1):23-30, 2004. http://www.accessmylibrary.com/coms2/summary_0286-4658287_ITM, Nov. 2007.
- [4] M. A. Constantino-González, D. D. Suthers. A Coached Collaborative Learning Environment for Entity-Relationship Modeling. In *Proceedings of the Intelligent Tutoring Systems International Conference (ITS 2000)*, páginas 325-333, 2000.
- [5] C. A. Cunha, Padrões de condutas de aprendizagem por conflito sócio-cognitivo em conteúdo operatório. Tese de Doutorado, Faculdade de Educação – UNICAMP, 1999.
- [6] E. S. J. Faria, J. M. Adán Coello, K. Yamanaka, Forming Groups for Collaborative Learning in Introductory Computer Programming Courses Based on Students' Programming Styles: An Empirical Study. In *ASEE/IEEE Frontiers in Education Conference (FIE2006)*. Piscataway, NJ, 2006.
- [7] E. S. J. Faria, Constituição de grupos para o aprendizado colaborativo de programas a partir da comparação de programas de estudantes. Dissertação de Mestrado, Pontifícia Universidade Católica de Campinas, 2005.
- [8] R. M. Felder, Reaching the second tier: learning and teaching styles in college science education. *Journal of College Science Teaching*, 23(5):286-290, 1993. <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/Secondtier.html>, Abr. 2008.
- [9] R. M. Felder; R. Brent, Cooperative Learning in Technical Courses: Procedures, Pitfalls, and Payoffs, 1994. <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/Coopreport.html>, Abr. 2008.
- [10] R. M. Felder; B. A. Soloman. Index of Learning Styles. <http://www.ncsu.edu/felder-public/ILSpa.html>, Jun. 2008.
- [11] L. J. Feldmann; R. J. Hofinger, Active Participation by Sophomore Students in the Design of Experiments. In *ASEE/IEEE Frontiers in Education Conference (FIE1997)*, 1526-1527, 1997.
- [12] A.A. Gokhale, Collaborative Learning Enhances Critical Thinking, *Journal of Technology on Education*, 7(1), 1995.
- [13] M. Guzdial, K. Carroll, Exploring the Lack of Dialogue in Computer- Supported Collaborative Learning. In *Computer Supported Collaborative Learning Conference (CSCL)*, Boulder, Colorado, 2002, <http://books.google.com/books?hl=en&lr=&id=KVnABS1ADEYC&oi=fnd&pg=PA418&dq=Explaining+the+Lack+of+Dialogue+in+Computer+Supported+Collaborative+Learning&ots=4goOUoveH4&sig=wEhJ3wIYdASbk60XBU5qyNJGRf4#PPA421,M1>, Abr. 2008.
- [14] S. Hung, L. Kwork, R. Chan, Automatic Program Assessment. *Computers and Education*, 20(2):183-190, 1993.
- [15] A. Inaba; T. Supnithi; M. Ikeda; J. Toyoda; R. Mizoguchi, Learning Goal Ontology Supported by Learning Theories for Opportunistic Group Formation. In *Proceedings of AI-ED'99*, Le Mans, France, páginas 67-64, 1999. <http://www.ei.sanken.osaka-u.ac.jp/pub/thep/thep-aied99.pdf>. Abr. 2008.
- [16] D. Jackson, A Semi-Automated Approach to Online Assessment. In *iTiCSE 2000*, Helsinki, Finland, 164-167, 2000.
- [17] A. Y. Kolb, D. A. Kolb, The Kolb learning style inventory - version 3.1: Technical Specifications. Boston, MA: Hay Resources Direct, 2005.
- [18] N.P. Kuri, Kolb's Learning Cycle: an Alternative Strategy for Engineering Education. In *International Conference on Engineering Education (ICEE)*, Czech Republic, 1999.
- [19] P. Lévy, Collective Intelligence: Mankind's Emerging World in Cyberspace, Perseus Books; 2000.
- [20] N. Maier, Problem Solving and Creativity in Individuals and Groups. Belmont, CA: Brooks/Cole. 1970.
- [21] M. M. McManus, R. M. Aiken, Teaching collaborative skills with a group leader computer tutor, *Education and Information Technology*, 1(1): 75-96, 1996. www.springerlink.com/index/k71072175T323042, Abr. 2008.
- [22] W. S. Menezes, J. M. Adán Coello, Formação de Grupos em Ambientes de Aprendizado Colaborativo de Programação Usando Estilos de Aprendizagem e Elementos da Teoria do Conflito Socio-cognitivo. In *XXVI Congresso da Sociedade Brasileira de Computação*, páginas.128 – 137, 2006.
- [23] W. S. Menezes, Formação de Grupos em Ambientes de Aprendizado Colaborativo de Programação usando Estilos de Aprendizagem e Elementos do Conflito Sócio-cognitivo. Dissertação de Mestrado, Pontifícia Universidade Católica de Campinas, 2007.
- [24] S. A. Mengel, V. Yerramilli, A Case Study Of The Static Analysis Of The Quality Of Novice Student Programs. In *SIGCSE'99*, New Orleans. LA, USA, páginas 78-82, 1999.
- [25] M. C M. Milome, Uma Nova Abordagem ao Processo de Ensino e Aprendizagem. *IV SEMEAD – Seminários em Administração*, 1999. http://www.ead.fea.usp.br/Semead/4semead/Artigos/adm_geral/Milone.PDF, Abr. 2008.



- [26] T. P. O'Brien; L. E. Bernold, D. Akroyd, Myers-Briggs Type Indicator and Academic Achievement in Engineering Education, Int. *Journal of Engineering Education*. 14(5):311-315, 1998.
- [27] J. Piaget, *The Development of Thought: Equilibration of Cognitive Structures*. New York: Viking Penguin, 1977.
- [28] J. Piaget, *A equilibração das estruturas cognitivas*. Rio de Janeiro: Zahar, 1976.
- [29] S. Puroo, V. Vaishnavi, Product Metrics for Object-Oriented Systems. *ACM Computing Surveys*, 35(2):191-221, 2003.
- [30] M. Redmond, A Computer Program to Aid Assignment of Student Project, In *SIGCSE 2001*, Charlotte, NC, USA, 2001.
- [31] T. Schorsch, Cap: An Automated Self-Assessment Tool To Check Pascal Programs For Syntax, Logic And Style Errors.). In *SIGCSE'95*, Nashville, TN, USA, páginas 168-172, 1995.
- [32] J. E. Sharp. Learning Styles and Technical Communication: Improving Communication and Teamwork skills. In *ASEE/IEEE Frontiers in Education Conference (FIE1998)*, 1998. <http://fie.engrng.pitt.edu/fie98/papers/1358.pdf>, Abr. 2008.
- [33] F. H. S. Silva. O papel do conflito sociocognitivo na elaboração de noções de conservação por interações de pares constantes e múltiplos. Tese de Doutorado, Faculdade de Educação – UNICAMP, 1999.
- [34] I. Sommerville, *Engenharia de Software*. 6ª ed. São Paulo: Addison Wesley, 2003.
- [35] D.D. Suthers, *Computer Aided Education and Training Initiative, Relatório Técnico*, University of Pittsburgh, 1998.
- [36] R. I. Sutliff; V. Baldwin, Learning Styles: Teaching Technology Subjects Can Be More Effective. *The Journal of Technology Studies*, 2001.
- [37] L. V. Toscani, P. A. S. Veloso, *Complexidade de Algoritmos – Análise, Projeto e Métodos*. Porto Alegre: Sagra Luzzatto, 2001.
- [38] M. Xenos, D. Stavrinoudis, K. Zikouli, D. Christodoulakis, Object-oriented metrics – a survey. In *Proc. of the FESMA - Federation of European Software Measurement Association*, Madrid, Spain, páginas 1-10, 2000.

