

4. **Triangulação com trabalhos relacionados**

Ao investigar as dificuldades no ensino dos métodos de avaliação propostos pela Engenharia Semiótica identificamos dificuldades que ultrapassam os limites dos métodos e da teoria. Os resultados indicam que a dificuldade em compreender e aplicar o MIS e o MAC é, principalmente, reflexo da deficiência no desenvolvimento de três capacidades: de interpretação sistemática, de abstração, e de articulação de uma visão global.

Em geral o aluno já vem do ensino médio com deficiências nestas capacidades, as quais são agravadas pela estrutura curricular dos cursos de Ciência da Computação que não oferece condições destas capacidades serem desenvolvidas. Mesmo quando há iniciativas de incluir estratégias de ensino-aprendizagem os professores se deparam, por vezes, com turmas muito grandes e alunos pouco motivados.

Fizemos a triangulação exógena com os trabalhos relacionados, a partir das categorias acima (os resultados da nossa pesquisa), para verificar se essas mesmas categorias eram ou não consistentes com os resultados dos trabalhos relacionados em IHC e em CC. Em outras palavras, a triangulação, em nosso contexto, diz respeito ao processo de análise do conteúdo desses trabalhos à luz das categorias de nossa pesquisa – as capacidades de interpretação sistemática, de abstração, e de articulação de uma visão global. Cada trabalho foi examinado com o objetivo de identificar a existência ou não de significados que fossem relacionados às capacidades discutidas em nossa pesquisa. A presença dessas capacidades poderia ser explicitamente definida no discurso científico desses trabalhos (ou seja, o próprio trabalho enuncia como dificuldade no ensino, por exemplo, a capacidade de abstração) ou desvelada a partir do discurso implícito (por exemplo, quando, sem nomear como tal, uma dificuldade concreta exemplificada no trabalho dizia respeito ao que definimos na pesquisa como dificuldade de abstração). Ao final desse processo, o resultado da triangulação revelou a convergência entre nossas

categorias e os trabalhos relacionados, validando cientificamente, da perspectiva qualitativa, os resultados de nossa pesquisa.

A área de IHC em geral também enfrenta problemas semelhantes. Segundo, Carroll (2003) a área de IHC tem muitas teorias e muitos métodos derivados das diversas áreas das ciências cognitivas e da Engenharia de Software. A consequência disto é uma imensa fragmentação. Estas teorias e métodos não são articulados. Na perspectiva de Carroll, embora muitos dos livros didáticos de IHC abordem os fundamentos das ciências e teorias da área, a tendência é enfatizarem métodos e técnicas individualmente. Consequentemente, conforme mostram os resultados de nossa pesquisa, os professores têm dificuldade em fornecer uma visão global ao aluno. Deste modo, a própria configuração da área reforça a dificuldade que o aluno já possui em ter visão global de um método, de uma teoria e de IHC.

Ainda no contexto de IHC, mesmo quando se trata do ensino de assuntos específicos, os professores também encontram dificuldades, como relatam os trabalhos de Berkun (2002), Vat (2001), Rosson *et al.* (2004) e Leventhal *et al.* (2004).

O trabalho de Berkun discute as dificuldades presentes no treinamento de profissionais da área de design de interfaces. Na opinião deste autor, resolver um problema apenas no nível tático não é suficiente, é necessário desenvolver a habilidade de atacar o problema também no nível estratégico. Para isto é preciso melhorar a habilidade dos profissionais em ter uma visão global, o que frequentemente é pouco desenvolvida, como argumenta o autor. Ele sugere que o ensino do processo de design seja feito pela prática ou “*at least by watching good people do what they do.*” (Berkun, 2002, p. 124)⁴⁷

A sugestão de Berkun vai ao encontro da proposta de Schön (1983) da prática reflexiva. De acordo com o segundo autor um ambiente de aprendizagem apropriado para o ensino de métodos qualitativos é aquele que privilegia a prática guiada por um especialista. O especialista pode ajudar o aprendiz a desenvolver a capacidade de analisar uma situação do ponto de vista global. É também esta estratégia de propor atividades práticas com o acompanhamento personalizado do

⁴⁷ Tradução da autora: “pelo menos pela observação do trabalho de bons profissionais.”

professor que os participantes de nossa pesquisa sugerem. A prática também é sugestão dos dois autores seguintes.

Vat (2001) fala sobre a necessidade da prática para melhorar a experiência dos estudantes e a compreensão sobre os métodos de design e avaliação de IHC. Também aponta o desafio de proporcionar uma visão holística de IHC em uma única disciplina de IHC, compartilhando as mesmas considerações de Carroll (2003).

Por sua vez, Rosson *et al.* (2004) sugere a prática com projetos realistas que estimulem o interesse dos alunos. Mas adverte que estes projetos devem ser administráveis dentro de um semestre. Em sintonia com Rosson *et al.*, Fernandez (2004) também sugere a inserção de projetos práticos de IHC com o envolvimento de usuários reais. Segundo Fernandez (2004) os sistemas baseados na web são bons candidatos a projetos para serem realizados no transcorrer de um semestre letivo.

A prática apontada por Berkun (2002), Vat (2001) e Rosson *et al.* (2004) converge com as iniciativas didáticas identificadas pela nossa pesquisa. Os professores participantes também acreditam que a prática pode melhorar a compreensão sobre os métodos.

O enfoque do trabalho de Leventhal *et al.* (2004) por sua vez é sobre os desafios enfrentados pelos professores de Engenharia de Usabilidade. A experiência docente dos autores mostrou que a fase de especificação é um desafio pedagógico para os professores. Ao longo do artigo os autores mostram que os alunos têm dificuldade durante todo o processo: eles não sabem por onde começar, que tarefas incluir, como separá-las e quando parar o processo de especificação. Alguns alunos reconhecem que a tarefa de análise não é fácil e compreendem que isto ocorre porque tudo é muito abstrato e eles geralmente não estão acostumados a pensar desta maneira, do mesmo modo que apontam os resultados sobre o MIS e o MAC.

Por exemplo, eles compreendem o conceito de cenários, entretanto não sabem como abstraí-los em tarefas dos usuários. Eles também assumem que tem dificuldade em diferenciar tarefas de detalhes de implementação. O grande desafio da especificação é o movimento cíclico de ir do nível abstrato para o concreto e para o abstrato novamente para extrair as tarefas (concreto) dos cenários (abstrato), assim como ocorre no MIS e no MAC.

A proposta dos autores foi melhorar a abordagem do ensino da especificação da Engenharia de Usabilidade incluindo conceitos de casos de uso do Processo Unificado de Desenvolvimento de Software, conseguindo assim melhores resultados. Embora estes autores mencionem que apresentar os requisitos de projeto e ensinar o processo de especificação são desafios para os professores de Engenharia de Usabilidade, o relato não explora isto explicitamente, apenas comenta sobre as dificuldades dos alunos.

A dificuldade em lidar com conceitos abstratos também é relatada no contexto do ensino do conceito de não-determinismo (Armoni *et al.*, 2008). A grande dificuldade dos alunos está em compreender o caráter imprevisível do não-determinismo. Segundo os autores, a experiência prévia dos alunos com máquinas das quais sempre se espera uma resposta correta faz com que eles desenvolvam uma visão equivocada deste conceito.

O desconforto em lidar com situações que admitem várias soluções e não apenas uma solução correta também é enfrentado pelos alunos e professores do MIS. Na fase de identificação dos signos como metalinguísticos, estáticos e dinâmicos não há certo e errado. A definição destes tipos de signos deve ser usada como um *framework* pelo avaliador que vai instanciá-las conforme sua experiência prévia, mantendo sempre a coerência na classificação. A dificuldade em lidar com a possibilidade de várias respostas corretas para um mesmo problema está relacionada à capacidade de interpretação sistemática. Alunos e professores têm dificuldade em compreender que a atribuição de significados (guiada pelas categorias propostas pela teoria) é um processo único e válido.

No caso das dificuldades relatadas por Armoni *et al.* (2008) estes autores sugerem que uma proposta para minimizar estas dificuldades é expor o conceito de não-determinismo em vários contextos de CC, como por exemplo, sistemas operacionais.

O ensino de orientação a objeto (OO) também enfrenta dificuldades por lidar com conceitos abstratos. Or-Bach & Lavy (2004) realizaram um estudo para verificar a compreensão de conceitos fundamentais de OO tais como classes, herança e polimorfismo. Tal como ocorrido em nossa pesquisa, os resultados mostraram que a maioria dos estudantes não alcançou o nível de abstração desejado. Alguns novatos não conseguiram decompor um grande procedimento em unidades funcionais menores e optaram por associar o procedimento como um

todo em uma única classe. Os autores, em sintonia com os trabalhos de Schön (1983) e Berkun (2002), acreditam que se fosse dada uma oportunidade de orientação para revisar os designs dos participantes da pesquisa, o resultado seria diferente. Além disto, os autores supõem que a apresentação de exemplos e contra-exemplos do uso de um determinado conceito pode aprofundar a compreensão sobre ele.

A área de IHC parece realmente não ser a única a enfrentar desafios no processo de ensino-aprendizagem. Yuen (2007) realizou um estudo qualitativo para investigar como os alunos conheciam e aplicavam o que aprendem em um curso de CS1⁴⁸. Foi solicitado a oito estudantes que resolvessem um problema que envolvia funções, recursão e *arrays* – três tópicos gerais bastante conhecidos em programação, e considerados difíceis. Os resultados revelaram que em geral os alunos parecem memorizar os conceitos de funções e recursão, mas têm dificuldades em explicá-los ou usá-los em seus códigos.

Adicionalmente, os alunos tendem a codificar as soluções do problema sem planejamento ou mesmo um rascunho prévio. O propósito do estudo era principalmente exploratório, mas o autor sugere que uma solução para combater tais problemas é tentar compreender o processo de conhecimento dos estudantes logo após eles terem sido ensinados e não apenas no final do semestre.

Os resultados identificados por Yuen parecem ser semelhantes aos resultados sobre o MIS e o MAC no sentido de que os alunos têm dificuldade de interpretação e abstração: memorizam os conceitos mas não sabem interpretá-los a ponto de explicá-los ou usá-los. Outro ponto é a ansiedade por uma resposta rápida, deixando de lado a reflexão sobre uma proposta de codificação.

Polycarpou *et al.*, (2008) também apresentam considerações sobre as dificuldades no ensino de indução para Ciência da Computação. Embora o conceito de indução seja central para várias áreas de CC e haja várias iniciativas para melhorar o ensino deste conceito, os alunos ainda têm dificuldades que vão desde compreender os passos envolvidos em uma prova por indução até a dificuldade em realizar a prova por indução em problemas que não similares aos que já tenha tido contato anteriormente. O que se percebe é um conhecimento

⁴⁸ Também conhecida como Introdução à Computação, uma disciplina que trabalha com programação.

superficial e mecânico. O aluno não é capaz de compreender o conceito e aplicá-lo, como ocorre com os problemas de programação relatados por Yuen (2007).

Algo semelhante aos trabalhos de Polycarpou *et al.*, (2008) e Yuen (2007) ocorre no ensino do MAC com relação as etiquetas. As definições das etiquetas são conhecidas por profissionais, alunos e professores, entretanto todos revelam dificuldades em relacionar estas etiquetas às rupturas de comunicação. No caso do MIS uma das dificuldades é abstrair os problemas em classes de problemas.

Investigando as possíveis razões para o alto índice de desistência de alunos de Ciência da Computação os autores Beaubouef & Mason (2005) discutem vários assuntos sobre o ensino de CS1 e CS2, pois a maioria das desistências ocorre no primeiro e no segundo ano do curso. Eles discutem assuntos relacionados a estudantes, professores e até a infra-estrutura do curso, mas não apresentam soluções para os problemas identificados. Entre as razões das desistências dos alunos os autores mencionam que a habilidade para solucionar problemas é muito fraca. Por exemplo, os alunos não cometem erros de sintaxe quando escrevem seus códigos. A dificuldade é elaborar o próprio algoritmo. Apesar de identificarem estes problemas os autores reconhecem que o ensino sobre solução de problemas está fora do escopo dos cursos de Ciência da Computação. Esta lacuna no currículo de CC aumenta o vão entre os requisitos necessários para a formação do profissional da área de Informática e o conteúdo trabalhado em sala de aula, como já apontam os resultados da nossa pesquisa com relação à abstração, interpretação e visão global.

Para solucionar um problema é preciso interpretar, abstrair e ter uma visão global do problema. Ao ter dificuldade em elaborar o próprio algoritmo os alunos demonstram que embora consigam atingir um nível bom de interpretação (não cometendo erros de sintaxe, por exemplo) não conseguem a abstração e a visão global necessárias para a elaboração de um bom algoritmo.

Ainda no contexto de programação o trabalho de Beaubouef & Mason (2005) mostra que não há tempo suficiente para corrigir e discutir com cada aluno os seus programas. Conseqüentemente, os professores não têm condições de identificar com precisão os pontos onde os estudantes precisam de mais prática ou instrução. A questão do tempo também foi identificada em nossa pesquisa e estes resultados se coadunam com a visão de Norvig (2001) sobre o tempo necessário para alguém ser um bom programador. Norvig apresenta uma lista de argumentos mostrando

que, assim como aprender a tocar piano, aprender a ser um bom programador também exige um longo período de tempo.

O trabalho de Beaubouef & Mason (2005) também aborda a questão do papel do professor como um dos elementos que contribui para o problema educacional. Embora a maioria seja capaz de ensinar disciplinas como CS1 e CS2 eles não foram preparados para isto. Logo a qualidade da formação é comprometida.

Nesta mesma linha, McCauley (2004) também chama a atenção para o fato de que os professores de CC geralmente não são preparados para ensinar. De fato, a maioria deles tenta fazer o seu melhor, mas sem qualquer tipo de conhecimento pedagógico. Com frequência a estratégia é usar o modelo das aulas onde ele se formou. Entretanto, o autor sugere que os professores devem ensinar do mesmo modo como fazem pesquisa: com planejamento e rigor. Outra sugestão é dialogar com a comunidade da área da educação. Para ilustrar a sugestão do autor ele descreve uma experiência bem sucedida com a taxonomia de Bloom's⁴⁹ para investigar o desempenho dos alunos em um curso de CS1.

A *Grounded Theory* (Glaser & Strauss, 1967) é um método de pesquisa que faz uma análise sistemática dos dados sem nenhuma hipótese prévia. Esta análise é um processo iterativo que consiste em identificar conceitos através das evidências coletadas em entrevistas procurando por códigos, conceitos e finalmente categorias. Ao relacionar as categorias e investigar as conexões entre os conceitos uma nova teoria emerge. Allan (2003), ao utilizar esta teoria dentro do contexto de CC, faz alguns questionamentos sobre esta teoria que tem uma forte semelhança com as dúvidas que os participantes da nossa pesquisa relatam.

O processo iterativo de análise requer diferentes e crescentes níveis de abstração. Em suas experiências pessoais Allan relata sua dificuldade em identificar quando parar a análise, assim como os participantes da nossa pesquisa tem dificuldades em saber qual deve ser o nível de profundidade de uma inspeção semiótica. Allan também sente falta de uma definição clara sobre o processo de codificação das entrevistas. Esta dificuldade está relacionada ao apego aos métodos quantitativos. Em uma perspectiva qualitativa apenas os conceitos da

⁴⁹ Esta taxonomia consiste em seis níveis de objetivos educacionais para o domínio cognitivo: 1) lembrar, 2) compreender, 3) aplicar, 4) analisar, 5) avaliar, 6) criar.

codificação são suficientes. A interpretação destes conceitos oferece a flexibilidade necessária para os métodos qualitativos, tornando cada análise única, mas sempre sistemática por seguir os conceitos sugeridos pelo método.

Outras semelhanças entre as dúvidas sobre o uso da *Grounded Theory* no contexto de CC e a nossa pesquisa são: (i) dúvida sobre o que é um código *versus* dúvida sobre o que é um signo metalinguístico, estático ou dinâmico; (ii) perda de foco ao fazer uma micro-análise (palavra por palavra) dos códigos *versus* empobrecimento dos resultados da fase de interpretação do MAC quando as etiquetas são analisadas isoladamente; (iii) aumento da confiança no processo de codificação e diminuição da incerteza com a prática *versus* melhor compreensão do processo de etiquetagem com a prática do MAC.

Em resumo, várias das dificuldades identificadas em nossa pesquisa são experimentadas em outros contextos externos à Engenharia Semiótica. Encontramos semelhanças tanto no próprio contexto de IHC quanto fora dele, em disciplinas de programação, orientação a objetos, e no ensino de conceitos-chaves de CC como não-determinismo e indução.

Observando a convergência dos resultados de nossa pesquisa com os resultados dos outros trabalhos apresentados neste capítulo, verificamos que o problema vai muito além do ensino dos métodos da Engenharia Semiótica e apontam para dificuldades muito mais abrangentes. A presença dessas dificuldades como obstáculos em várias áreas apontam para a necessidade de repensar não apenas o processo de ensino dos métodos examinados em nossa pesquisa, mas também, de forma integrada, o processo de ensino-aprendizagem da área de Ciência da Computação.

Segundo os resultados de nossa pesquisa, o desenvolvimento das capacidades de interpretação sistemática, abstração e visão global são necessárias para uma melhor compreensão de vários conteúdos da área de CC. Certamente não são as únicas e tampouco têm o mesmo peso em cada uma das sub-áreas da CC. São, contudo, como a pesquisa revela, capacidades que podem ser úteis termos em mente para apoiar nossas discussões e ações relativas ao processo de ensino-aprendizagem da área de CC.

Em comum com o relato autocrítico de nossos entrevistados, nos trabalhos relacionados destacamos que muitos professores dos cursos de CC são ex-alunos de cursos de CC e carregam e repassam todas as deficiências do currículo.

Reconhecemos que há um grande esforço por parte da maioria dos professores em melhorar o processo de ensino-aprendizagem de suas disciplinas, entretanto isto se faz através de uma abordagem de “tentativa e erro”.

Embora haja eventos e trabalhos que discutem o ensino de CC, ainda não há soluções para vários dos problemas já identificados. Por exemplo, ainda não se percebe nenhuma mudança curricular que possa contribuir para a formação docente na área. A grade curricular não contempla disciplinas de metodologia de ensino, por exemplo, mesmo que em caráter optativo, embora vários dos alunos de CC se tornarão professores da área. A iniciativa dos cursos de Licenciatura em Ciência da Computação vem tentar mitigar apenas o problema de formação docente para atuar no ensino médio.

Outro ponto de convergência entre os relatos de nossos entrevistados e os trabalhos relacionados é o tempo necessário para a formação de bons profissionais. A formação de profissionais competentes requer o desenvolvimento de capacidades que exigem um longo período de tempo. Entretanto, é inviável tanto para a área acadêmica quanto para o mercado de trabalho que o tempo de formação de um profissional seja maior do que ocorre hoje (com cursos de 4 ou 5 anos). Sendo assim, é urgente que haja então um processo de inovação no processo de ensino-aprendizagem de CC para que seja possível mitigar as deficiências que os alunos apresentam logo ao ingressar na universidade e prover meios para que as capacidades exigidas de um bom profissional sejam melhor desenvolvidas no período de formação universitária.

Muitos dos problemas identificados no contexto dos métodos de avaliação da Engenharia Semiótica são vividos na área de IHC e são também compartilhados por outras áreas de CC. A triangulação dos resultados de nossa pesquisa com os trabalhos aqui analisados indica que a própria área de CC parece estar sofrendo de um dos resultados alcançados: dificuldade em construir uma visão global do seu conjunto de disciplinas e dos problemas enfrentados pela área. Uma vez que há dificuldades compartilhadas, as soluções aplicadas em uma determinada disciplina pode beneficiar outras, ainda que esta não seja a intenção. Isto porque, se um aluno suplanta uma dificuldade em uma disciplina, nas próximas disciplinas que exijam capacidades similares às desenvolvidas anteriormente, um importante desenvolvimento já foi obtido. Isso não parece, contudo, ser o suficiente. Parece ser necessário, portanto, que a relação entre as disciplinas do currículo de CC

sejam amplamente discutida pelo corpo docente e amplamente divulgada e trabalhada com o corpo discente, a fim de potencializar as ações positivas desenvolvidas no interior de cada uma das disciplinas do currículo.