

Processo de Aquisição de Conhecimento em Programação de Computadores Apoiado por Ambientes Inteligentes

Eleandro Maschio^{1,2}, Alexandre Ibrahim Direne²

¹Departamento de Informática – Universidade Federal do Paraná (UFPR)
Caixa Postal 19.081 – 81.531-990 – Centro Politécnico
Curitiba – PR – Brasil

²Coordenação do Curso de Tecnologia em Sistemas para Internet
Universidade Tecnológica Federal do Paraná (UTFPR)
Guarapuava – PR – Brasil

eleandro@hotmail.com, alexd@inf.ufpr.br

Abstract. *This paper describes how high-level knowledge about human expertise features can be modeled, represented and further interpreted to support learning and tutoring interactions. We focused up, specifically, on the problem of providing an epistemology for describing knowledge about exercise statements and students' states of belief, in Computer Programming, treating it through (1) a method based on genetic graphs for managing the complexity of courseware authoring of problem statements and (2) a general process for dynamic modeling a learner's knowledge by overlaying it against the domain expertise features. The method and the model are both supported here by implemented prototype software tools.*

Resumo. *O presente artigo concentra-se em descrever como o conhecimento de alto-nível sobre aspectos de experiência humana pode ser modelado, representado e, então, interpretado a fim de suportar interações de ensino e aprendizagem. Incide-se, especificamente, no problema de prover uma epistemologia para a descrição do conhecimento sobre enunciados e estágios de conhecimento de aprendizes, em Programação de Computadores, tratando-o por meio de (1) um método baseado em grafos genéticos para gerenciar a complexidade na autoria de enunciados e (2) um processo geral para a modelagem dinâmica do conhecimento de aprendizes através da sobreposição das capacidades do domínio previamente descritas. Ambos são suportados por ferramentas implementadas.*

1. Introdução

As disciplinas introdutórias de Programação de Computadores culminam grande responsabilidade em um curso de Ciência da Computação. Cabem-lhes a introdução dos princípios e consequente iniciação ao desenvolvimento de perícias na área de programação. Geralmente isso se faz em tempo restrito, caso considerada a exigência cognitiva da aprendizagem, mediante a intercalação de abordagens conceituais e práticas. O legado do trabalho nas citadas disciplinas (tanto competências quanto deficiências) desencadeia-se por entre todo o currículo do curso e tende a desembocar na prática profissional do egresso.

Contudo, conforme revisão feita sobre as pesquisas da área, pouco se investigou acerca do desenvolvimento de habilidades específicas do estereótipo de um programador perito. Nesse sentido, [Norvig 2001] manifesta-se contra tal desatenção ressaltando a íngreme curva de aprendizado enfrentada por um iniciante que objetiva proficiência na área. Corroborar tal fato a vasta literatura que anuncia ensinar programação em poucos dias, se não em horas.

O citado estudo expõe maiores agravantes quando se trata não da suficiência, mas da excelência em determinado domínio de conhecimento. A exemplo disso, [Gladwell 2008] documenta uma pesquisa, realizada com estudantes da Academia de Música de Berlim, que compara o tempo destinado à prática com o nível atingido pelos indivíduos. Constatou-se que todos iniciaram a tocar por volta dos cinco anos de idade. Nos primeiros anos, praticavam de duas a três horas semanais. Em torno dos oito anos de idade, diferenças reais apareceram. Aqueles que terminaram como melhores de suas turmas começavam a praticar mais do que o restante: 6 horas semanais aos 9 anos, 8 horas aos 12, 16 horas aos 14, aumentando para mais de 30 horas semanais de prática aos 20 anos. Nesta idade, os instrumentistas de elite totalizavam uma média de 10.000 horas de prática musical. Em contraste, aqueles que eram somente bons acumularam 8.000 horas e os que não se destacaram, pouco mais de 4.000 horas.

Assim, [Norvig 2001] aponta a prática deliberada como elemento decisivo no desenvolvimento de perícias em uma grande variedade de áreas. Sugere o desafio de tarefas que vão um pouco além da habilidade corrente do aprendiz, a fim de que ele tente desempenhá-las, analisar a própria performance durante e depois de realizá-las para, então, corrigir possíveis erros. Outro estudo [Lave 1988] reforça que um aprendizado mais efetivo requer tarefas bem definidas, com nível de dificuldade apropriado ao aprendiz em particular, bem como *feedback* informativo, oportunidade para repetição e correção de erros.

Dessa forma, consta-se o sequenciamento dos enunciados¹ a serem oferecidos ao aprendiz como um dos elementos pedagógicos cruciais ao desenvolvimento de perícias. Assim, prover uma modelagem precisa, tanto do conhecimento do domínio quanto do aprendiz, para subsidiar decisões nesse âmbito, perfaz uma contribuição relevante na área de Inteligência Artificial (IA). Ainda convém destacar que a modelagem de aprendizagens não se trata de um conceito recente no campo de Sistemas Tutores Inteligentes (STIs). Contudo, pesquisas anteriores tendiam a se concentrar em questões pedagógicas em detrimento das psicológicas. Como consequência, os esforços se destinaram à proposição de arcabouços que estabelecessem relações de arquitetura entre os modelos pedagógico e do aprendiz.

A partir disso, em contexto mais amplo, pode ser visto como desafio encontrar meios efetivos que propiciem o aprendizado, principalmente em Programação de Computadores. Na atualidade, os aprendizes têm o cotidiano imerso em tecnologias das mais variadas e sessões tradicionais de ensino mostram-se insuficientes para atraí-los e motivá-los ao estudo. Entretanto, pela característica imersiva, o ensino baseado em computadores tende a sobrepor muitas das abordagens obsoletas. Como consequência, a superação do platô constatado na modelagem do aprendiz tende a se mostrar naturalmente como o passo

¹Termo usado neste documento para referir-se, de maneira, genérica a exemplos e exercícios de um determinado domínio de conhecimento.

seguinte das pesquisas em Inteligência Artificial e Educação.

2. Definições

A problemática discutida embasa-se nos conceitos concernentes à aquisição de princípios e desenvolvimento de perícia, bem como ao sequenciamento de enunciados. Ambos são detalhados na sequência.

2.1. Aquisição de Princípios e Desenvolvimento de Perícia

O processo de aprendizagem em domínios de natureza prática e complexa fundamenta-se, segundo [Lesgold et al. 1988], na aquisição de conhecimentos sobre **princípios** e consequente desenvolvimento destes até consolidarem **perícia** na área. A **aquisição de princípios** corresponde à assimilação de conhecimento formal (científico, em alguns casos) repassado inicialmente na formação do aprendiz. Princípios são os fundamentos do domínio ao qual se referem. O **desenvolvimento de perícia**, por sua vez, diz respeito à construção do conhecimento experiencial, ou seja, a conquista de habilidade pela prática. Abrange a integração dos princípios aprendidos com a experiência obtida até o momento e, ainda possivelmente, campos distintos de conhecimento.

Embora o conhecimento de princípios seja semelhante tanto em peritos (experitos) quanto em aprendizes, o conhecimento experiencial apresenta várias e contundentes diferenças. Para tanto, o desenvolvimento de perícia ocorre através da exposição do aprendiz a casos exemplares e não exemplares [Lesgold et al. 1988, Direne and Scott 2001]. Cada caso ao qual se expõe o aprendiz é compreendido como uma oportunidade para o desenvolvimento de perícia. A aprendizagem, portanto, acontece pela indução a partir desses casos.

Adicionalmente, [Lesgold 1988] propôs uma teoria de currículo que privilegia aspectos de perícia. O currículo, em STIs, diz respeito à ordem pedagógica de apresentação na qual o conhecimento do domínio é oferecido ao aprendiz. Convém evidenciar que modelos para descrição de perícias não devem ser confundidos com os formalismos empregados por arquiteturas tutoriais para a definição de currículos, como taxonomias e partonomias [Howard 1990]. Tais modelos consistem em formalismos ainda mais gerais, sendo as taxonomias e as partonomias um mero subconjunto do universo de abstração que pode ser obtido por meio de dimensões de perícia apropriadamente definidas em um STI.

Conforme a abordagem descrita, o processo de aprendizagem em Programação de Computadores pode ser visto como a aquisição de conhecimentos sobre os princípios de lógica de programação e consequente desenvolvimento destes até consolidarem perícias na área. Distinguem-se, os princípios das perícias [Maschio and Direne 2007], porquanto os primeiros tangem à compreensão de instruções primitivas isoladas, atendo-se à rigidez léxica, sintática e semântica determinada pela linguagem que se usa. As perícias, por sua vez, abrangem o encadeamento e o aninhamento de duas ou mais daquelas instruções a fim de compor um algoritmo. Assim, a análise e a resolução de problemas através de algoritmos são alicerçados pela soma dos respectivos princípios e perícias. No entanto, a tarefa de programar impõe uma alta carga cognitiva sobre os iniciantes, pois exige a aplicação de ambas as categorias de conhecimento na externalização do código fonte. Como consequência, os aprendizes incorrem em erros frequentes que antagonizam a evolução desses indivíduos enquanto programadores.

Dentro disso, [Pimentel and Direne 1998] identificou um conjunto de características do estereótipo de um programador perito, a saber: (a) precisão sintática; (b) precisão semântica; (c) identificação de estruturas principais no programa fonte (busca por palavra-chave); (d) simulação mental dos estados do computador durante a execução; (e) catálogo de erros; (f) mapeamento mental das estruturas do programa; (g) checagem de pré-condições; (h) análise do problema; (i) integração dos subproblemas; (j) generalização da solução; (k) reutilização de soluções já conhecidas; e (l) catálogo de soluções.

Contudo, desde então, inexistem pesquisas que enfoquem o desenvolvimento de perícia em Programação de Computadores. Frente a isso, a corrente pesquisa retoma os citados estudos a fim de incorporar técnicas de autoria que permitam a definição formal de perícias em domínios de natureza prática e complexa. Tais esforços podem levar a modelagem dinâmica do aprendiz um passo adiante nas pesquisas da área.

2.2. Sequenciamento de Enunciados

Diante da perspectiva do desenvolvimento de perícias, [Pimentel and Direne 1998] descreve **medidas cognitivas** no ensino de Programação de Computadores. São uma forma de se obter meta-conhecimento acerca dos enunciados de uma base de conhecimento de um STI. Funcionam como um forte elo entre o Modelo de Domínio e o Modelo do Aprendiz nestes sistemas.

As citadas medidas têm o papel de quantificar cognitivamente um enunciado de Programação de Computadores. Ocupam-se de avaliar o quanto determinado enunciado exige de um aprendiz em termos de conhecimentos adquiridos e capacidades desenvolvidas na progressão do aprendizado. Nesse sentido, o mesmo estudo define a **carga cognitiva** de um enunciado como a própria capacidade em exercitar o aprendiz na construção de um programa de computador. Portanto, a carga cognitiva pode ser dividida em alguns componentes que individualmente se responsabilizam por medir um tipo de contribuição/exigência proporcionado pelo enunciado ao aprendiz.

A complexidade do software é um dos componentes sugeridos pelo citado estudo. Todavia, como os programas destinados a iniciantes são majoritariamente pequenos, a complexidade pode ser melhor determinada usando linhas de código (LOC) em conjunto com complexidade estrutural (CE) e respectivos detalhes de implementação (DI). A complexidade estrutural avalia o número de estruturas de decisão e de repetição, bem como o grau de dependência entre as mesmas estruturas (aninhamentos). Por sua vez, os detalhes de implementação consideram a quantidade de condições específicas que dificultam o cumprimento do enunciado.

Além dos componentes que concernem à complexidade, cada enunciado contribui para que o aprendiz desenvolva características do estereótipo de um programador perito (conforme Seção 2.1). Por consequência, os outros componentes da carga cognitiva de um enunciado baseiam-se e têm medidas definidas a partir desse grupo de características. Com isso, podem quantificar a contribuição/exigência de um enunciado frente a cada um dos atributos do estereótipo.

Portanto, a carga cognitiva de um enunciado constitui-se da ponderação das medidas cognitivas que o compõem. Embora cada medida contribua com uma parcela desta carga cognitiva, isso não ocorre de maneira uniforme. Diferentes medidas possuem pe-

sos distintos na composição da carga cognitiva de uma instância. Os pesos são valorados através de informações obtidas de especialistas no ensino de Programação de Computadores.

Evidencia-se então que estas medidas cognitivas fornecem subsídios adequados para a escolha do próximo enunciado a ser trabalhado em uma sessão de ensino, colaborando para que a progressão do aprendizado seja mais concisa e coesa. Assim, há utilidade prática na classificação de um enunciado, por grau de exigência/contribuição, diante de um elenco preciso de medidas cognitivas. Uma galeria de enunciados poderá ser catalogada e ordenada formalmente frente às dimensões que denotam o (sub)conjunto de medidas cognitivas consideradas.

Parte da importância desse procedimento pode ser constatada quando um problema de programação significativamente mais complexo do que o anterior é proposto a um aprendiz. O fato pode levar à ocorrência de erros múltiplos, impedindo o sucesso do indivíduo por longos períodos de tempo.

Partindo desse pressuposto, um fator adverso em ambientes de ensino destinados à Programação de Computadores reside na extrema rigidez na ordenação das respectivas galerias de enunciados. Mesmo que, mediante decisões fundamentadas, os projetistas e construtores de tais ambientes tenham alcançado determinada sequência de ensino, não há garantias de que a sequência privilegie toda a heterogeneidade de perfis e competências de diferentes aprendizes. Portanto, pode-se deduzir que a sucessão dos enunciados atende a um grupo específico de aprendizes, geralmente de desempenho entre baixo e mediano, e negligencia aqueles que não se enquadram no estereótipo idealizado.

Uma contribuição anterior [Soldado and du Boulay 1995] antecipou a complexidade dos enunciados como um dos principais componentes de motivação do indivíduo no aprendizado. Em consonância, [Pimentel and Direne 1998] explica que “a repetição sistemática de enunciados completamente diferentes, porém, de graus de complexidade aproximadamente iguais pode elevar consideravelmente a auto-confiança do aprendiz, mantendo-o motivado e produtivo”.

Logo, seria não somente desejável, como também necessária, a flexibilização dos ambientes de ensino para que se adaptassem às competências de cada aprendiz. Nessa direção, foi desenvolvida a ferramenta de autoria **Sequence** [Pimentel and Direne 1998], inicialmente concebida para a área médica e depois adaptada ao ensino de Programação de Computadores. Baseia-se em medidas cognitivas e avalia um enunciado mediante a complexidade (LOC, CE e DI) aliada às perícias de precisão sintática e semântica, análise do problema, reutilização de soluções e simulação mental. Com isso, consegue sugerir uma sequência inicial de enunciados e então adaptá-la à situação corrente do usuário mediante informações oriundas do Modelo do Aprendiz no STI.

Entretanto, consta-se que ainda permanece um vasto espaço para pesquisa, pois faltam estudos que se estendam na focalização das características cognitivas que compõem a complexidade de um enunciado de Programação de Computadores. Ademais, seria interessante que diferentes pesquisas reavivassem o estado da arte desta área e, como consequência, guinassem os ambientes de ensino destinados à Programação de Computadores a uma nova e promissora geração.

3. Arcabouço Conceitual e Ferramentas de Software

Discorre-se, nesta seção, sobre métodos e ferramentas propostos como contribuição no auxílio ao desenvolvimento de perícias em Programação de Computadores. Ambos incidem sobre a definição de perícias do domínio, bem como a consequente elicitación e catalogação de enunciados a serem fornecidos ao aprendiz.

3.1. Modelagem do Aprendiz e Sequenciamento de Enunciados

Conforme anteriormente abordado, consta-se o sequenciamento de enunciados fornecidos ao aprendiz como um dos elementos cruciais ao desenvolvimento de perícias. Para tanto, uma contribuição relevante consiste na modelagem apurada do conhecimento do domínio e do aprendiz, a fim de subsidiar tal processo de decisão. Nesse sentido, a critério de exemplo, os enunciados seguintes estão ordenados conforme a carga cognitiva exigida na resolução, da menos complexa para a mais complexa.

Enunciado 1

Fornecida a idade do indivíduo, responda se a maioridade já foi atingida.

Enunciado 2

Fornecida a idade do indivíduo, responda se o voto é obrigatório. No Brasil, a obrigatoriedade da votação incide sobre os cidadãos entre 18 e 70 anos.

Enunciado 3

Fornecida a idade do indivíduo, responda se o voto é obrigatório, facultativo ou não permitido. No Brasil, a obrigatoriedade da votação incide sobre os cidadãos entre 18 e 70 anos. O voto é facultativo para aqueles que têm entre 16 e 18 anos, ou mais de 70 anos.

Supondo a resolução do segundo enunciado pelo aprendiz, seria interessante dispor de mecanismos em um STI que, conforme a performance na resolução, pudessem retroceder ao primeiro enunciado ou de avançar ao terceiro enunciado. Tais mecanismos somente são possíveis mediante uma modelagem precisa e convencionada que situe o perfil do aprendiz frente ao conhecimento do domínio. Ademais, é necessário que se classifique cada enunciado conforme a fração exercitada deste conhecimento.

Considerando isto, decidiu-se, nesta pesquisa, descrever o conhecimento do aprendiz através do **Modelo de Sobreposição** (*Overlay*) [Goldstein 1979]. A abordagem consiste na representação do aprendiz como um subconjunto do conhecimento do domínio. Através da sobreposição destes conjuntos, fica evidente quais objetivos devem ser atingidos pelo aprendiz. Todavia, implica que a representação de conhecimento de ambos os modelos sejam comparáveis, ou a mesma.

Assim, buscou-se uma representação que pudesse descrever perícias no domínio de Programação de Computadores e que, frente ao Modelo de Sobreposição, denotasse a incidência do conhecimento do aprendiz sobre o conhecimento do domínio. A correlação entre as diversas perícias e sub-perícias também deveria ficar evidente. Frente a isso, o **Grafo Genético** [Goldstein 1979], técnica de representação escolhida, responde às necessidades expostas e também privilegia o Modelo de Sobreposição. Trata-se, resumidamente, de um arcabouço para representação do conhecimento. Nesse tipo específico de grafo, o conhecimento do domínio (ou de um perito) é constituído por um conjunto de fatos ou regras (vértices) interconectados por relações (arestas) que evoluem:

da simplificação à elaboração, do desvio à correção, da abstração ao refinamento, da especialização à generalização e do pré-requisito ao pós-requisito.

A Figura 1 ilustra um exemplo simplificado de representação, em um grafo genético, das perícias que constituem um domínio de conhecimento. O conhecimento do aprendiz é modelado como um subconjunto (ou subgrafo) das perícias mapeadas no domínio. Todas as perícias pendentes ao aprendiz são evidenciadas pela sobreposição dos dois grafos. Assim, através dos mesmos aspectos de representação externa, tem-se realçado o estágio atual do aprendiz frente ao conhecimento do domínio, ou seja, a sobreposição propriamente dita.

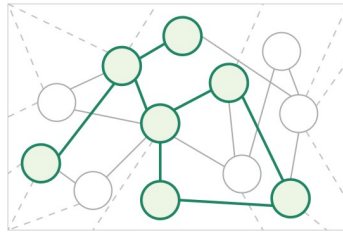


Figura 1. Sobreposição do Modelo do Aprendiz frente ao conhecimento do domínio

A partir dos dois modelos, do aprendiz e do domínio, pode-se prover um processo de busca heurística que oriente regiões do grafo a serem exploradas pelo aprendiz. Havendo, portanto, a indicação de perícias a serem prioritariamente cumpridas pelo aprendiz, conforme o estágio atual de conhecimento (Figura 2).

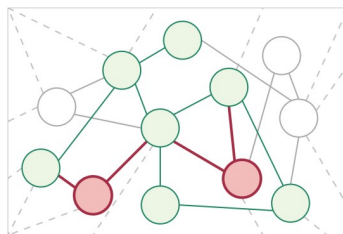


Figura 2. Indicação de regiões do grafo a serem exploradas

Em adição, empregou-se o mesmo mapeamento de perícias como um gabarito para os enunciados a serem propostos pelo aprendiz. Cada enunciado contribui para o desenvolvimento de perícias específicas do aprendiz, podendo também ser visto como um subgrafo do conhecimento do domínio, conforme Figura 3. Dessa maneira, os enunciados podem ser catalogados de acordo com o gabarito. Trata-se de uma extensão original para o trabalho de [Goldstein 1979].

Isto posto, o Modelo do Aprendiz adquire caráter dinâmico por ser alimentado com a avaliação do cumprimento dos enunciados propostos. Supõe-se a avaliação por um especialista externo (humano), diante do gabarito de perícias fornecido pelo enunciado. O procedimento documenta o progresso do aprendiz e conseqüente evolução da estrutura do grafo genético. O especialista externo pode dar lugar a uma ferramenta de diagnóstico.

Outra possibilidade, alcançada pela corrente pesquisa e garantida pela representação em grafo, consiste em sintetizar o Modelo do Aprendiz de um grupo (sala

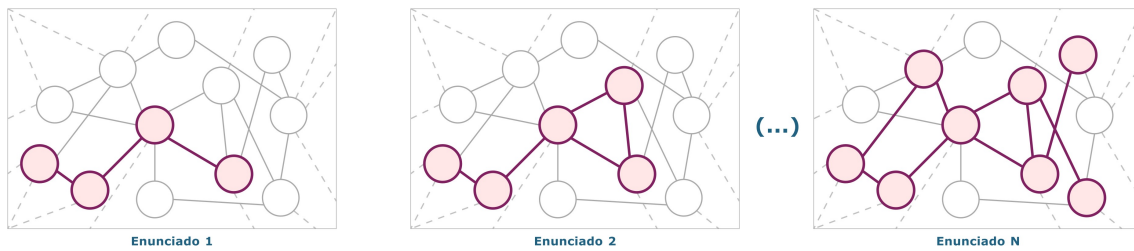


Figura 3. Catálogo de enunciados

de aula) e assim ter um espelho da ênfase com que as perícias são ensinadas pelo instrutor. Com isso, pode-se notar que um instrutor se sobressai no ensino de determinadas perícias, ou que um segundo negligencia algumas outras.

Convém salientar que, nos exemplos dados, as perícias são sinalizadas como desenvolvidas ou não-desenvolvidas. Entretanto, estuda-se a implementação do desenvolvimento gradativo das perícias pelo aprendiz por meio de uma escala entre tais extremos. Dessa forma, a progressão seria gradual e mais fiel ao denotar o percentual aprendido/cumprido de cada perícia.

Também podem ser fornecidos detalhes da dimensão temporal do progresso do aprendiz, ou seja, apresentar um histórico do desenvolvimento das perícias em um período específico. Remetendo-se à escala de progressão gradual, recém citada, há âmbito para uma representações externas bastante informativas. Por exemplo, exibir os vértices sendo (mais) coloridos à proporção do tempo, em uma animação. Adicionalmente, o mesmo recurso é também aplicável à sintetização do Modelo do Aprendiz de um grupo. Logo, é possível visualizar a progressão do aprendizado de uma sala inteira, ou de um curso inteiro.

Por fim, enfatiza-se que a modelagem do aprendiz e do conhecimento do domínio através da sobreposição abre uma vasta gama de possibilidades de aplicação em um STI. Nenhuma das ideias apresentadas nesta seção é complexa em si, entretanto se mostram bastante úteis para guiar o Modelo Pedagógico ou orientar instrutores (humanos) em sessões de ensino.

3.2. Ferramentas de Autoria

O processo de autoria descrito é suportado por duas ferramentas implementadas. A primeira concentra-se na **descrição do conhecimento do domínio** mediante o detalhamento das perícias componentes. Conforme Figura 4, trata-se de uma ferramenta de autoria que utiliza visualmente a representação de grafo genético. As opções de edição são acessadas por *menus* de contexto. Permite-se basicamente: (a) a inserção, edição e remoção de perícias; (b) o estabelecimento das relações evolucionárias entre perícias; (c) a definição de uma perícia inicial; e (d) a validação do modelo descrito quanto à alcançabilidade das demais perícias a partir daquela inicial. O arquivo com as informações do modelo é salvo em formato XML.

A segunda ferramenta, usando a mesma representação gráfica, destina-se à **catalogação e elicitación de enunciados** descritos como subgrafos do conhecimento de domínio formalizado pela primeira (Figura 5). Isto é, possibilita que sejam demarcadas as

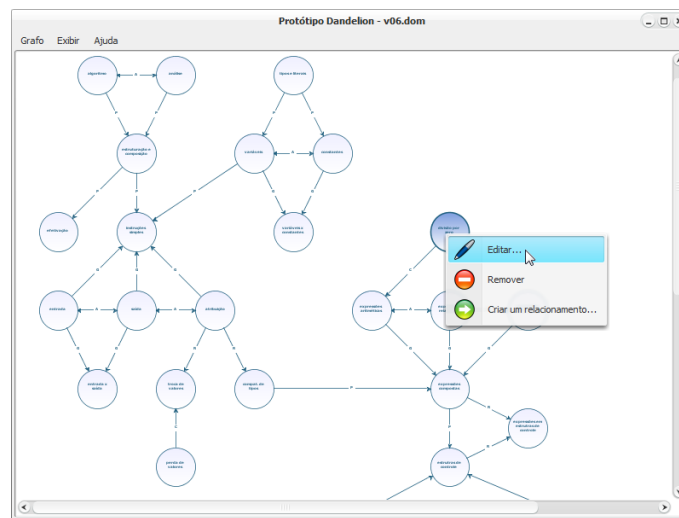


Figura 4. Ferramenta para a descrição do conhecimento do domínio

perícias específicas para as quais cada enunciado contribui. A sobreposição do enunciado enfatiza a fração do conhecimento exercitada. Em adição, pode-se inspecionar a quantidade de enunciados que exercitam determinada perícia, como também se existem perícias não abordadas pelo conjunto (catálogo) de enunciados corrente. No mesmo sentido, foi implementada uma verificação mais profunda sobre a coesão do catálogo e consequente requisição da autoria de enunciados adicionais.

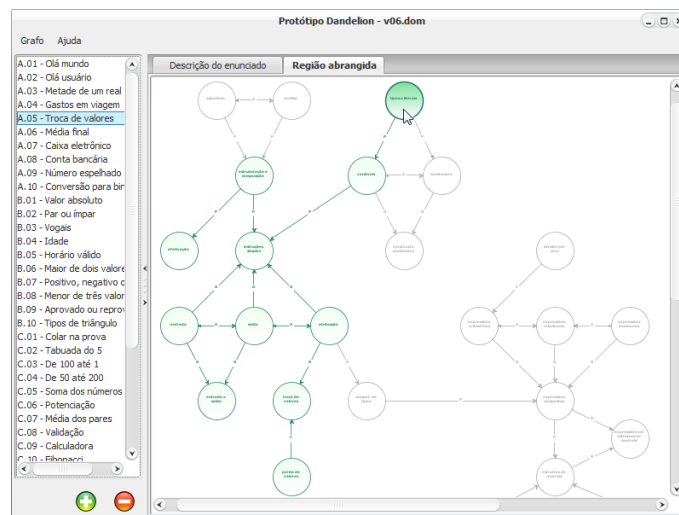


Figura 5. Ferramenta para a elicitación e catalogación de enunciados

4. Considerações Finais e Trabalhos Futuros

O método delineado propõe uma abordagem formal para a estruturação do conhecimento do domínio de Programação de Computadores, como também mecanismos para suportar a descrição simbólica dos aspectos de perícia na modelagem dinâmica de aprendizes. Procurou-se prover uma modelagem detalhada que evidenciasse a evolução do aprendiz frente ao conhecimento do domínio, como também a contribuição individual de cada

enunciado para o referido progresso. Para isso, foram revisitados a técnica de Grafo Genético e o Modelo de Sobreposição [Goldstein 1979]. De imediato, foram implementadas duas ferramentas de autoria, sendo que a primeira permite a descrição do conhecimento do domínio por meio de perícias componentes e a outra atua na consequente catalogação e elicitação de enunciados perante o domínio formalizado.

Esforços subsequentes se concentrarão em avaliar formalmente o processo de autoria implementado pela pesquisa e depois em aprimorar as potencialidades do arcabouço proposto. Destaca-se a validação de duas hipóteses, a saber: (a) o grafo genético tem caráter geral suficiente, como linguagem, para o desenvolvimento de uma ampla gama de ambientes destinados ao processo de ensino e aprendizagem; e (b) as ferramentas interativas que suportam o processo de autoria mostram-se apropriadas no sentido de proporcionar maior rapidez e de também prover um arcabouço unificado para domínios de natureza prática e complexa.

Referências

- Direne, A. I. and Scott, D. (2001). Identifying the component features of expertise in domains of complex visual recognition. Information technology research institute technical report series, University of Brighton, Lewes Road, Brighton, UK.
- Gladwell, M. (2008). *Outliers: The Story of Success*. Little, Brown and Company, New York, 1 edition.
- Goldstein, I. P. (1979). The genetic graph: a representation for the evolution of procedural knowledge. *International Journal of Man-Machine Studies*, 11(1):51–77.
- Howard, R. W., editor (1990). *Concepts and Schemata*. Cassell Educational.
- Lave, J. (1988). *Cognition in Practice: Mind, Mathematics and Culture in Everyday Life*. Cambridge University Press, New Cambridge.
- Lesgold, A. (1988). *Toward a theory of curriculum for use in designing intelligent instructional systems*, pages 114–137. Springer-Verlag New York, Inc., New York, NY, USA.
- Lesgold, A., Rubinson, H., Feltovitch, P., Glaser, R., Klopfer, D., and Wang, Y. (1988). *The nature of expertise*, chapter Expertise in a Complex Skill: Diagnosing X-ray Pictures, pages 311–342. Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, England.
- Maschio, E. and Direne, A. I. (2007). Múltiplas representações externas para o ensino de Programação de Computadores. In *Workshop de Ambientes de Apoio à Aprendizagem de Algoritmos e Programação*, volume 18 of *Anais do XVIII Simpósio Brasileiro de Informática na Educação. SBIE 2007*, São Paulo, SP, Brasil. Sociedade Brasileira de Computação.
- Norvig, P. (2001). Teach yourself programming in ten years. <http://norvig.com/21-days.html>.
- Pimentel, A. R. and Direne, A. I. (1998). Medidas cognitivas no ensino de Programação de Computadores com Sistemas Tutores Inteligentes. *Revista Brasileira de Informática na Educação (IE)*, 3:17–24.
- Soldado, T. D. and du Boulay, B. (1995). Implementation of motivational tactics in tutoring systems. *Journal of Artificial Intelligence in Education*, 6(4):337–378.