
Múltiplas Representações Externas para o Ensino de Programação de Computadores

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

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

²Departamento de Ciência da Computação
Universidade Estadual do Centro-Oeste (UNICENTRO) – Guarapuava – PR – Brazil
eleandro@hotmail.com, alexd@inf.ufpr.br

***Abstract.** This article reports on languages and tools to support the learning process that covers the phase between the acquisition of programming principles and skills. A metacognitive approach is proposed here to accomplish a solution based on the concepts of multiple external representations and interactive learning environments (ILE).*

***Resumo.** Este artigo aborda linguagens e ferramentas que ajudam na cobertura da aquisição de conhecimento na transição entre o aprendizado de princípios e o desenvolvimento de perícias em programação de computadores. Propõe-se uma abordagem metacognitiva, baseada em múltiplas representações externas e apoiada na arquitetura de um ambiente interativo de aprendizagem (AIA).*

1. Introdução

A aprendizagem de programação de computadores se divide na aquisição de conhecimento sobre *princípios* e sobre *perícias* (tradução do termo *expertise*, da língua inglesa). Os princípios tangem à compreensão de instruções isoladas (primitivas) da linguagem de programação, atendo-se aos aspectos de rigidez léxica, sintática e semântica das citadas instruções. As perícias, por sua vez, abrangem o encadeamento e o aninhamento de duas ou mais destas instruções a fim de compor um algoritmo. Frente a isso, a tarefa de programar impõe uma alta carga cognitiva sobre os iniciantes, pois exige a aplicação de ambas as categorias de conhecimentos na externalização do código fonte. Como consequência, os aprendizes cometem erros freqüentes que os impedem de cumprir seus objetivos práticos.

Conforme revisão literária realizada sobre as pesquisas da área, pouco se fez acerca da aquisição de conhecimento no entremeio de princípios e perícias. Algumas tentativas, no entanto, foram desempenhadas a fim de construir depuradores que se encaixassem no ensino de programação e permitissem ao aprendiz conduzir experimentos de execução simbólica do código fonte. Porém, ainda resta um enorme vazio de pesquisa original a ser coberto para oferecer aos aprendizes o monitoramento adequado desses experimentos por meio de (múltiplas) representações externas [Ainsworth 2006], visuais e textuais, compatíveis com suas habilidades em desenvolvimento. Além de ressaltar tal compatibilidade, influenciando assim uma

recente linha de projeto dos ambientes interativos de aprendizagem, a utilização de um conjunto multirrepresentacional no ensino tende a desembocar numa abordagem metacognitiva por parte do instrutor e do aprendiz, implicando numa divisão mais equitativa de responsabilidade e aprimorando o desempenho naquilo que se ensina.

Diante do exposto, este trabalho tem o objetivo de tratar o problema apresentado por meio de linguagens e ferramentas de visualização dos aspectos de execução de programas fonte que ajudam na cobertura da aquisição de conhecimento na transição entre o aprendizado de princípios e o desenvolvimento de perícias de programação de computadores. Para isso, propõe-se aqui uma abordagem metacognitiva baseada em múltiplas representações externas e apoiada na arquitetura de um ambiente interativo de aprendizagem.

2. Resenha Literária

O termo Representação Externa (RE) [Cox e Brna 1995], no escopo do presente documento, refere-se ao uso de técnicas para representar, organizar e apresentar conhecimento. Remete, portanto, a uma ampla variedade de representações que compreende desde modelos proposicionais/sentenciais até modelos gráficos/diagramáticos. Preenchendo a lacuna entre esses extremos, há representações intermediárias que associam elementos gráficos e textuais. Logo, são exemplos de RE: sentenças em linguagem natural ou em linguagens formais, tabelas, listas, grafos, mapas, projetos, diagramas, animações e até mesmo as realidades aumentada e virtual.

Segundo definição [Palmer 1978], qualquer RE deve ser descrita em termos de: (a) do mundo representado; (b) do mundo representante; (c) de quais aspectos do mundo representado estão sendo expressos; (d) de quais aspectos do mundo representante compõem a modelagem; (e) da correspondência entre os dois mundos. Como consequência do bom emprego de REs nas tarefas do dia-a-dia, estas tornaram-se um suporte valioso à resolução de uma diversidade de problemas de caráter mais formal, bem como um poderoso auxílio para diminuir a carga cognitiva, sendo empregadas em várias áreas do conhecimento humano.

Além do supracitado, há evidências [Cox e Brna 1995] expondo vantagens da utilização de REs como suporte tanto ao ensino quanto à resolução de problemas. Neste sentido, ultimamente as pesquisas se concentram na combinação de diferentes REs para intermediar o aprendizado. Múltiplas Representações Externas (MREs) [Ainsworth 2006], denominação dada ao conjunto, tem se mostrado um nicho próspero no atendimento às necessidades cognitivas de aprendizes diversos.

Houve pesquisas abordando exclusivamente a potencialidade e os possíveis problemas associados à aprendizagem com, simultaneamente, mais de uma RE [Ainsworth 2006]. Estas sustentam que MREs relacionadas resultam num aprendizado mais flexível e intuitivo, dando espaço a diferentes idéias e estratégias. Contudo, também se constatou que aprendizes podem não ter êxito em se beneficiarem das vantagens anteditas, pois estas nem sempre são facilmente atingidas.

Constatou-se também a incidência de MREs como recurso de depuração e/ou visualização em ambientes de programação [Romero et al. 2003]. Evidência disso são os termos “visualização de algoritmos”, “visualização de programas” e “visualização de

software”, todos análogos e referentes à re-representação gráfica de aspectos do código textual.

3. Conjunto Multirrepresentacional Envolvido

Cabe a esta seção introduzir o conjunto de MREs definido como parte da solução deste trabalho (que remete às prescrições do *framework* DeFT, descrito em [Ainsworth 2006]). Além disso, apresenta-se a correspondência entre alguns dos recursos semânticos de relacionamento entre as REs e as funções propostas na taxonomia funcionalista de Ainsworth. Para detalhes da arquitetura funcionalista do protótipo, ver [Maschio 2007].

3.1. Categorização de REs

Relaciona-se nesta seção o elenco das categorias de REs empregadas no protótipo desenvolvido, associadas ao devido valor metacognitivo.

Fluxogramas: a escolha da notação de fluxograma, como correspondente ao algoritmo textual, se baseou na notável necessidade do aprendiz desenvolver conceitos de mais baixo nível que precedem a noção de embutimento de escopos das linguagens de alto nível de abstração. Entre esses conceitos, se sobressai a compreensão do funcionamento de desvios progressivos e regressivos, que ficam explícitos no fluxograma;

Algoritmos em Pascal: a linguagem Pascal, amplamente utilizada no ambiente acadêmico, particulariza a idéia de programação estruturada ao redor do conceito de embutimento de escopos (ou contextos). Dessa forma, ressalta-se que a seqüência de processamento das instruções de um bloco ocorre sem saltos e o final lógico da execução coincide com o final sintático do texto;

Cliques de correspondência: mediante um clique em determinado elemento do fluxograma, destaca-se automaticamente a linha de código Pascal correlata, atuando na correspondência entre as duas notações. A situação inversa também ocorre;

Realce de elementos: trata do destaque automático de elementos conforme a progressão do fluxo de controle durante a execução do algoritmo;

Recolhimento e expansão de blocos de elementos: permite encapsular visualmente o que está contido dentro de uma estrutura (condicional ou de repetição) escolhida, transfigurando-a numa espécie de “caixa-preta;

As demais categorias de REs são: cores, comentários textuais, pontos de interrupção e destaque de variáveis com valores alterados; mais detalhes em [Maschio 2007].

3.2. Correspondência com as Funções de MREs

Compete a esta seção remeter à taxonomia funcionalista proposta por Ainsworth [Ainsworth 2006] alguns dos recursos semânticos das REs que compõem o conjunto recém categorizado aqui. Convém esclarecer que uma mesma RE pode desempenhar mais de um papel (ou função) em uma mesma situação de aprendizado.

Construção de compreensão aprofundada: a função que mais notoriamente embasa este projeto classifica-se como **abstração** [Ainsworth 2006], visto que a RE de fluxogramas apresenta mais baixo nível abstrato do que o código em Pascal. Com isso, ao passo que o aprendiz desenvolva habilidades, mesmo que parciais, de construção de

algoritmos em fluxogramas, o micromundo (protótipo) o subsidiará na aquisição de conhecimento sobre a tarefa correlata numa linguagem de alto nível (aqui, Pascal). Para isto, o aprendiz contará com uma série de instrumentos, como a geração automática do algoritmo textual equivalente e a visualização de aspectos dinâmicos (e também estáticos) de correspondência entre estas duas REs. As outras duas funções que tangem à construção de compreensão aprofundada (denominadas **relação** e **extensão**) também ocorrem dentro do conjunto de MREs estabelecido, porém não desempenham papel tão importante quanto a abstração.

Restrição de interpretação: nesta classe, a função que mais se sobressai é chamada de restrição por **propriedades herdadas** [Ainsworth 2006]. Nesse sentido, mesmo na ocorrência do pior caso, ou seja, quando o aprendiz não tem familiaridade com nenhuma das duas formas de representação de um algoritmo, a mais específica destas (fluxograma) deverá ajudar a restringir as ambigüidades daquela mais genérica (texto em Pascal). Além disso, ambas se beneficiarão do esclarecimento dado pelas outras REs categorizadas.

Papéis complementares: a abordagem adotada por este trabalho enquadra-se nesta classe como **informações complementares diferentes** [Ainsworth 2006]. Isso acontece porque, conforme dito, os processos intrínsecos a qualquer uma das duas formas de representação adotadas é exatamente o mesmo, mas as informações contidas em cada uma das descrições de algoritmo são fundamentalmente diferentes.

4. Conclusão e Trabalhos Futuros

Apresentou-se, neste documento, uma abordagem metacognitiva utilizando MREs para tratar da aquisição de conhecimento na fronteira de princípios e perícias de programação de computadores. Dentre os trabalhos futuros demarca-se primeiramente a coleta de dados a partir do uso inicial, que indicará tanto elementos do arcabouço proposto que possam estar incompletos quanto a demanda de alterações no protótipo. Um outro seguimento promissor consiste na extensão das funcionalidades do protótipo baseada na coleta de dados citada. Em paralelo, também está prevista a avaliação formal de aprendizes no intuito de verificar os benefícios de utilização do protótipo desenvolvido.

Referências

- Ainsworth, S. (2006). DeFT: A conceptual framework for considering learning with multiple representations. *Learning and Instruction*, 16(3):183–198.
- Cox, R. e Brna, P. (1995). Supporting the use of external representations in problem solving: the need for flexible learning environments. *Journal of Artificial Intelligence in Education*, 6(2/3):239–302.
- Maschio, E. (2007). Abordagem metacognitiva através de múltiplas representações externas para o ensino de programação de computadores. Dissertação de Mestrado, Universidade Federal do Paraná. <http://www.inf.ufpr.br/alex/dissertacaoEleandro.pdf>
- Palmer, S. E. (1978). *Cognition and categorization*, capítulo Fundamental aspects of cognitive representation, p. 259–303. LEA.

Romero, P., Cox, R., du Boulay, B., and Lutz, R. (2003). A survey of external representations employed in object-oriented programming environments. *Journal of Visual Languages and Computing*, 14(5):387–419.