
ProgrameFácil: Ambiente de Programação Visual para o Kit de Robótica Educacional RoboFácil

Leonardo Cunha de Miranda¹, Fábio Ferrentini Sampaio²,
José Antonio dos Santos Borges²

¹Departamento de Sistemas de Informação
Instituto de Computação – Caixa Postal 6.176
Universidade Estadual de Campinas – UNICAMP
13.083-970, Campinas, SP, Brasil

²Instituto de Matemática/Núcleo de Computação Eletrônica – Caixa Postal 2.324
Universidade Federal do Rio de Janeiro – UFRJ
20.001-970, Rio de Janeiro, RJ, Brasil

professor@leonardocunha.com.br, {ffs,antonio2}@nce.ufrj.br

Abstract. *The educational robotics is a challenging activity and playful that puts an effort on education during the creation of hardware and software based solutions. This work presents a computer programming environment based on visual elements to construct programs called ProgrameFácil. This visual computational language allows to program and to control electronic components such as LEDs, motors, displays, sensors of luminosity and temperature, connected to a hardware (educational robotics kit), such as RoboFácil. One of the main distinguishing factors of this environment is the possibility to visually simulate the implemented logic on a screen before transferring it to the hardware.*

Resumo. *A robótica educacional é uma atividade desafiadora e lúdica, que utiliza o esforço do educando na criação de soluções, sejam essas compostas por hardware e/ou software, visando a resolução de problemas. O presente trabalho apresenta o ambiente visual de programação denominado ProgrameFácil. Essa linguagem computacional icônica permite programar e controlar componentes eletrônicos como leds, motores, displays, sensores de luminosidade e temperatura, acoplados ao hardware de kits de robótica educacional, como por exemplo, o hardware do Kit RoboFácil. Um dos diferenciais desse ambiente em relação a outras soluções do gênero reside na possibilidade de simulação em tela da lógica implementada – programa – antes da sua efetiva transferência para o hardware do kit.*

Palavras-chave: Robótica Educacional, Hardware e Software, Informática na Educação.

1. Introdução

Há algumas décadas, diferentes teóricos discutem a inserção das Tecnologias da Informação e da Comunicação (TICs) no ambiente educacional. Busca-se estabelecer,

com essas novas tecnologias, uma melhor qualidade do ensino e ambientes de aprendizagem mais ricos e motivadores para os discentes.

Dentre o amplo espectro de idéias e propostas, no que se refere aos artefatos computacionais, é notório observar que a maioria das soluções apresentadas exploram “apenas” a vertente do *software*. No entanto, a demanda por novos aparatos de *hardware* na educação vem crescendo, sendo evidenciada, sobretudo, pelos esforços da comunidade acadêmica em propor a inserção da robótica com fins pedagógicos apoiados em resultados favoráveis à sua aplicação (Alves 2005; Santos e Menezes 2005; Schons et al. 2004; Zilli 2004; Steffen 2002; d’Abreu et al. 2002). Vale destacar que os trabalhos anteriormente citados em sua maior parte fazem uso de *kits* de robótica educacional comercializados no Brasil (na maioria importados). De fato, poucos são os trabalhos publicados que apresentam alternativas de *kits* para o contexto nacional (Sasahara e Cruz 2007; Miranda 2006; Chella 2002; d’Abreu 1999).

A robótica educacional é uma atividade desafiadora e lúdica, que utiliza o esforço do educando na criação de soluções, sejam essas compostas por *hardware* e/ou *software*, visando a resolução de um problema proposto – podendo o mesmo ser real. A maioria dos projetos pedagógicos que utilizam a robótica educativa na sala de aula faz uso da teoria construtivista de Jean Piaget para auxiliar o processo de ensino, permitindo ao aluno, nesse processo de construção do conhecimento, a oportunidade de participar de uma aprendizagem mais efetiva e desenvolver uma percepção mais acurada dos fenômenos científicos. Para Schons et al. (2004), a robótica pedagógica “[...] constitui nova ferramenta que se encontra à disposição do professor, por meio da qual é possível demonstrar na prática muitos dos conceitos teóricos, às vezes de difícil compreensão, motivando tanto o professor como principalmente o aluno.”.

Segundo Zilli (2004), a robótica educacional pode desenvolver as seguintes competências: raciocínio lógico; habilidades manuais e estéticas; relações interpessoais e intrapessoais; integração de conceitos aprendidos em diversas áreas do conhecimento para o desenvolvimento de projetos; investigação e compreensão; representação e comunicação; trabalho com pesquisa; resolução de problemas por meio de erros e acertos; aplicação das teorias formuladas a atividades concretas; utilização da criatividade em diferentes situações; e capacidade crítica.

Podem-se citar algumas vantagens com a adoção de *kits* de robótica educacional existentes no mercado em geral: 1) *hardware* e *software* construído direcionado a atender propósitos educacionais específicos; 2) flexibilidade na forma de utilização em distintas aplicações; 3) existência de documentação técnica e manual do usuário dos produtos, incluindo, em alguns casos, material pedagógico de apoio às atividades do professor em sala de aula; e 4) maior facilidade para aquisição e operação por usuários não familiarizados/especializados com as tecnologias da eletrônica e/ou informática.

Para Miranda (2006), poucos são os produtos que, disponíveis para comercialização no Brasil, possuem uma razoável relação “recursos X custo”. O uso efetivo desses *kits* no Brasil é dificultado pelo fato dos modelos importados terem, normalmente, um elevado custo para os padrões nacionais e os fabricados no Brasil possuírem limitações de *hardware* e/ou *software* para algumas aplicações, tais como a impossibilidade de permanecerem desconectados do computador durante sua execução e de não permitirem a reprogramação do seu comportamento. Nesse contexto, a geração

de soluções alternativas para resolver os problemas citados pode levar a um maior emprego dessa ferramenta pedagógica nas salas de aula das escolas com poucos recursos econômicos, principalmente por estudantes adolescentes do Ensino Médio.

O foco deste artigo é apresentar um dos artefatos de *software* (a linguagem computacional ProgrameFácil) projetado e implementado durante uma pesquisa de Mestrado. A escolha pelo desenvolvimento do *software* ProgrameFácil foi impulsionada pela necessidade de um ambiente visual – como o RoboLab (2004) – que tivesse uma interface de fácil utilização, possibilitando aos usuários da ferramenta programar o comportamento do *hardware* do Kit RoboFácil (Miranda 2006) de forma mais intuitiva e sem a necessidade de se conhecer a arquitetura eletrônica do referido artefato digital.

Apresenta-se na Seção 2 o ambiente ProgrameFácil. Na Seção 3 é realizada uma discussão sobre as tecnologias propostas neste artigo, incluindo uma comparação das soluções aqui descritas com produtos efetivamente em uso pela comunidade escolar no Brasil. Na última seção serão tecidas as considerações finais e indicados alguns trabalhos futuros relevantes à continuidade da pesquisa na área de robótica pedagógica.

2. Ambiente ProgrameFácil

O ProgrameFácil é uma linguagem computacional icônica (baseada na manipulação de ícones gráficos) que permite programar dispositivos eletrônicos e/ou eletromecânicos, tais como lâmpadas, *displays*, *leds*, motores de passo, e sensores de luminosidade e temperatura, fazendo uso das tradicionais construções de programação estruturada, tais como condicional e repetição. Inicialmente a linguagem foi desenvolvida para controlar o *hardware* do Kit RoboFácil, uma vez que o mesmo, na sua versão inicial só poderia ser (re)programado pelas vias “tradicionais”, fazendo uso das linguagens Assembly e C.

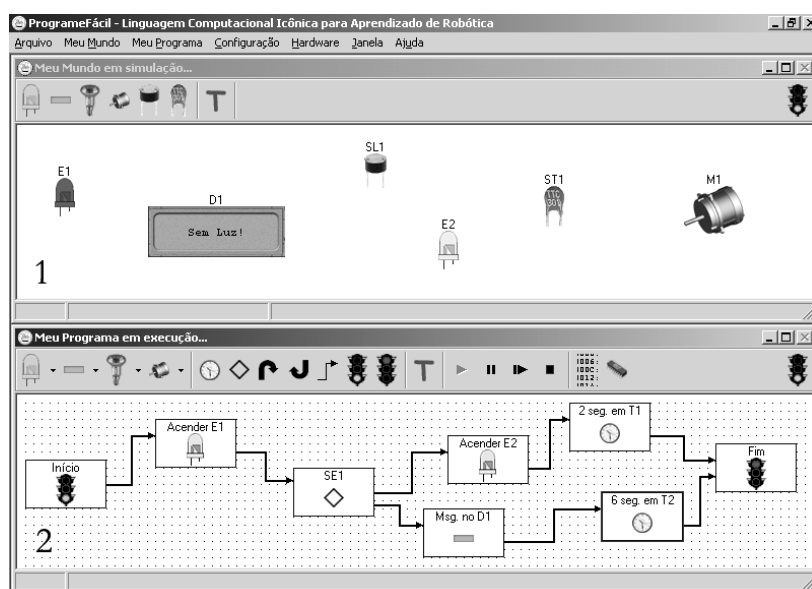


Figura 1. O ambiente ProgrameFácil apresentando as janelas Meu Mundo (1) e Meu Programa (2)

As fases de concepção, projeto e implementação da linguagem ProgrameFácil levaram sempre em conta a necessidade de criação de um ambiente intuitivo, de forma a torná-lo agradável de usar e eficiente para o controle dos recursos eletrônicos do

hardware do Kit RoboFácil. Neste sentido concebeu-se o ambiente de interação com o usuário como sendo constituído de dois mundos hipotéticos: o primeiro, chamado de “Meu Mundo”, onde se especifica a configuração desejada do *hardware* (*leds*, motores, sensores etc.) e visualiza-se seu funcionamento com a simulação; o segundo, chamado de “Meu Programa”, onde é permitido explicitar a programação do *hardware* detalhado no “Meu Mundo”. Tais mundos são implementados no ambiente ProgrameFácil através de duas janelas, conforme indicado na Figura 1. A adoção de etapas distintas e explícitas para “desenhar” e executar/simular modelos – mapeados no ambiente em diferentes janelas – tem como objetivo facilitar a averiguação da lógica empregada em cada programa criado pelo usuário.

A fim de garantir maior flexibilidade e qualidade na forma de interação entre a tríade Usuário – ProgrameFácil – *hardware* RoboFácil, incorporou-se um compilador a esta linguagem e um interpretador ao *firmware* do *hardware* RoboFácil. O compilador tem como função converter os objetos icônicos¹ que compõem o programa criado pelo usuário em macrocódigos *assembly*². O interpretador, presente no *firmware* do RoboFácil – e discutido em detalhes em outra publicação (Miranda 2006) – permite, por sua vez, converter programas escritos nos macrocódigos *assembly*, gerados ou não automaticamente pelo ProgrameFácil, em instruções que podem ser executadas no *hardware*, tais como as atividades de acender ou apagar *led*, escrever mensagem no *display*, movimentar o motor, entre outras.

Uma outra importante característica do ProgrameFácil é a existência de um simulador integrado ao ambiente. Esse simulador possibilita a visualização de um modelo de comportamento sendo executado/simulado. Essa característica faz com que, entre outras vantagens, não seja necessária a aquisição igualitária entre *software* e *hardware* quando do uso da robótica em ambientes reais de ensino.

2.1. Modelos de Comportamento e Objetos

Para facilitar o entendimento do funcionamento da linguagem ProgrameFácil faz-se necessário apresentar a definição do que entendemos por modelo de comportamento (ou modelo) e modelo de objetos, utilizados no contexto do ProgrameFácil.

Definiu-se no ProgrameFácil o termo modelo de comportamento, ou simplesmente modelo, como a configuração de um *hardware* de robótica hipotético ou real, com o seu respectivo programa de controle. A materialização dessas idéias na ferramenta ProgrameFácil foi alcançada através do conjunto de peças/componentes eletrônicos (Objetos de Hardware) do “Meu Mundo” (janela onde se especifica a configuração do *hardware* de um *kit* de robótica – área 1 da Figura 1) e dos “Objetos de Programação” do “Meu Programa” (janela onde se define o programa que irá controlar o funcionamento do *hardware* de robótica – área 2 da Figura 1).

¹ Objetos icônicos, no contexto do *software* ProgrameFácil, pode ser definido como símbolos gráficos que representam dispositivos eletrônicos ou estruturas existentes em linguagens de programação.

² Emprega-se, nessa pesquisa, o termo “macrocódigos *assembly*” para referenciar os códigos hexadecimais definidos (*bytecodes*), que representam a linguagem *assembly* virtual do *hardware* RoboFácil.

O modelo criado pelo usuário na ferramenta ProgrameFácil poderá retratar ou planejar arquiteturas, casos, atividades, comportamentos, falhas existentes ou não, contribuindo assim para a estruturação do pensamento e posterior simulação acerca dessa modelagem que permitirá infinitas combinações, passíveis de averiguação no que tange à lógica definida.

Na prática, para se criar um modelo na linguagem ProgrameFácil, é necessário realizar três etapas distintas: 1) especificar os “Objetos de Hardware” que serão utilizados na janela “Meu Mundo”; 2) incluir na janela “Meu Programa” os “Objetos de Hardware” e os “Objetos de Programação” que serão utilizados; e 3) estabelecer na janela “Meu Programa” o fluxo de execução através do objeto “Linha de Programação”. As formas de manipulação da interface a fim de realizar os passos descritos acima estão detalhados na ajuda “on-line” disponibilizada no *software*.

Um objeto no ProgrameFácil é uma representação gráfica – semelhante a um ícone – que pode ser manipulado tanto na janela do “Meu Mundo”, como do “Meu Programa”. Os objetos no ProgrameFácil foram divididos em três categorias para melhor identificar seu propósito: 1) Objetos de Hardware; 2) Objetos de Programação; e 3) Objetos de Apoio.

Os “Objetos de Hardware” representam dispositivos eletrônicos e foram divididos em duas sub-categorias: 1) Objetos de Hardware de Entrada; e 2) Objetos de Hardware de Saída. Os “Objetos de Programação” fazem referência às estruturas utilizadas em linguagens de programação. Por sua vez, os “Objetos de Apoio” visam unicamente proporcionar facilidades e recursos operacionais aos usuários da ferramenta, tal como, o recurso para incluir comentários textuais nos modelos.

Os “Objetos de Hardware de Saída” são nomeados igualmente em ambas as janelas, todavia possuem características distintas, por exemplo: o objeto Led do “Meu Mundo” possui a particularidade de “Cor” e do “Meu Programa” a de “Acender”. Após a inclusão de um “Objeto de Hardware” na Janela “Meu Mundo” é possível alterar as propriedades do mesmo utilizando, por exemplo, o “Objeto de Programação Estrutura de Controle Se”. Desta forma pode-se mandar acender ou apagar um *led*, escrever uma determinada mensagem no *display*, acionar um determinado motor por um certo tempo etc. baseado em decisões lógicas anteriores.

2.2. Janelas Meu Mundo e Meu Programa

O “Meu Mundo” é a janela onde o usuário poderá especificar o mundo hipotético que representa uma configuração de *hardware* de um *kit* de robótica. Os objetos concretos que representam os elementos eletrônicos disponíveis para escolha pelo usuário são apresentados em uma barra de ferramentas dessa janela, com exceção do objeto “Comentário”, que visa unicamente possibilitar a inserção de texto no modelo.



Figura 2. Barra de ferramentas da janela Meu Mundo

Na Figura 2 é apresentado a barra de ferramentas da janela “Meu Mundo” com seus Objetos de Hardware: 1) Led; 2) Display; 3) Lâmpada; 4) Motor; 5) Sensor de

Luminosidade; 6) Sensor de Temperatura; e 7) o recurso de Comentário. Os objetos de *hardware* passíveis de apresentação na janela “Meu Mundo” foram abstraídos da vida real. Portanto, para associá-los a um *hardware* físico, faz-se necessário conhecer suas características físicas e ações permitidas na realidade.

A janela “Meu Programa” pode ser definida como o local onde o usuário constrói o programa que irá controlar o funcionamento dos objetos existentes no “Meu Mundo”. Tal processo se dá através da definição das ações e do encadeamento de estruturas, tais como condicional e repetição, utilizando elementos icônicos para representá-las (presentes na barra de ferramentas da janela).



Figura 3. Barra de ferramentas da janela Meu Programa

Na Figura 3 é apresentada a barra de ferramentas da janela “Meu Programa” com seus Objetos de Hardware: 1) Led; 2) Display; 3) Lâmpada; e 4) Motor. Também estão disponíveis os Objetos de Programação: 5) Temporizador; 6) Estrutura de Controle Se; 7) Início de Looping; 8) Fim de Looping; 9) Linha de Programação; 10) Início de Programa; 11) Fim de Programa; e 12) o recurso de Comentário.

O objetivo é tornar possível a construção de uma lógica de programação entre os elementos presentes, formando assim o que é definido no contexto do ambiente ProgrameFácil como o “Programa do Modelo”. Para alcançar esse propósito a linguagem foi construída em conformação com cinco regras de formação:

Primeira regra: Cada objeto tem um ou nenhum sucessor na estrutura lógica de programação;

Segunda regra: O objeto Semáforo Vermelho (Objeto 11 da Figura 3) – que representa fim de programa – não poderá ter sucessores;

Terceira regra: O objeto SE (Objeto 6 da Figura 3) – que representa a estrutura condicional Se (If) – terá até dois sucessores;

Quarta regra: Cada objeto pode ter um ou mais predecessores na estrutura lógica de programação; e

Quinta regra: O objeto Semáforo Verde (Objeto 10 da Figura 3) – que representa início de programa – não poderá ter predecessores.

A inclusão de “Objetos de Hardware” na janela “Meu Programa” está atrelada à sua existência no “Meu Mundo”. Nessa janela, tanto “Objetos de Hardware” como “Objetos de Programação” podem ser utilizados.

O “Programa do Modelo”, ou simplesmente “Programa”, estabelece a forma de comportamento de um *hardware* de robótica e pode ser compreendido como o conjunto de objetos do “Meu Programa”, com as suas respectivas ações definidas, interligados por um fluxo de execução atemporal – fluxo constituído sem nenhuma referência temporal para execução, estabelecido pelo objeto “Linha de Programação”. O Programa inicia com o Objeto de Programação “Início de Programa” (semáforo verde) e termina no Objeto de Programação “Fim de Programa” (semáforo vermelho), quando da sua simulação no ambiente ProgrameFácil, ou execução no *hardware* RoboFácil.

Demonstra-se, na Figura 1, um exemplo de modelo de comportamento construído no ProgrameFácil. Esse modelo possui dois *leds* (E1 e E2), um motor (M1), um sensor de temperatura (ST1), um sensor de luminosidade (SL1) e um *display* alfanumérico (D1). Na área 1 é apresentado o “Meu Mundo” com esses “Objetos de Hardware” e na área 2 o “Meu Programa” com os respectivos “Objetos de Hardware” e “Objetos de Programação” que compõem o “Programa do Modelo”. O modelo exemplificado tem como objetivo acender o *led* E2, caso o sensor de luminosidade (SL1) esteja sob incidência de luz. Quando da execução desse modelo no *hardware* RoboFácil ou simulação no *software* ProgrameFácil, se o referido sensor não estiver sob incidência de luz, a mensagem “Sem Luz” deverá ser apresentada no *display* (D1).

2.3. Simulador e Compilador

O simulador foi incorporado ao *software* a partir da necessidade de tornar executável – virtualmente – o modelo anteriormente criado na tela do monitor, antes de realizar a transferência do programa criado para o *hardware* RoboFácil. Essa funcionalidade, entre outras vantagens, visa proporcionar um momento intermediário de depuração acerca do programa criado. As características de implementação do simulador permitem seu funcionamento de forma automática, após solicitação de início pelo usuário, ou interativamente – de forma pausada: passo-a-passo.

Assim, há dois semáforos agregados ao ambiente – um na janela “Meu Mundo” e outro na janela “Meu Programa”, sendo esses posicionados ao lado direito, na barra de ferramenta de cada janela, que funcionam em conjunto, e representam sempre o mesmo estado. Sua função é indicar o *status* do modelo de comportamento: em desenvolvimento (sinal vermelho) em ambas as janelas, em pausa (sinal amarelo) em ambas as janelas, em simulação (sinal verde) na janela “Meu Mundo” e em execução (sinal verde) na janela “Meu Programa”.

Na janela “Meu Mundo” os objetos do *hardware* hipotético ou real, quando em simulação, vão alterando suas propriedades baseadas no programa do modelo desenvolvido. Conforme a simulação vai prosseguindo, na janela “Meu Programa”, um “Ponteiro de Execução” – retângulo vermelho que contorna os “Objetos de Hardware” e “Objetos de Programação” da janela “Meu Programa” (se configurado sua visualização pelo usuário) – identifica o ponto atual de execução do programa.

Para se converter o programa de um modelo desenvolvido na linguagem computacional visual ProgrameFácil nos macrocódigos *assembly* concebidos para o *hardware* do kit em questão (*assembly* virtual do *hardware* RoboFácil), integrou-se um compilador a essa linguagem. A tradução é engendrada através da correspondência de macrocódigos *assembly* para cada “Objeto de Hardware” e/ou “Objeto de Programação” presentes no programa de um modelo, que em seguida será entendida pelo interpretador, componente do *firmware* do RoboFácil. Por exemplo, possibilitando especificar qual motor deve ser ligado, qual *led* deve ser apagado, entre outros.

Evidencia-se a preocupação, na interação com o usuário, em fornecer qualidade e conforto na utilização da ferramenta, minimizando pré-requisitos como conhecimentos avançados de programação. Assim, o processo de compilação do programa de um modelo é realizado apenas com um único clique do *mouse* sobre o botão correspondente a essa funcionalidade. Depois de concluído esse processo o

compilador pode apresentar uma janela informando o resultado obtido: compilação realizada com sucesso ou compilação realizada com erro(s).

3. Discussão

O tema (robótica educacional) apesar de não ser novo, ainda apresenta relativa escassez de situações do gênero no Brasil se comparado com outras áreas da informática na educação (fato este corroborado pelo reduzido número de publicações acadêmicas).

No contexto das inovações tecnológicas, no que se refere aos ambientes de *software* para programação de *hardwares* de *kits* de robótica educacional, cabe nesse momento uma análise comparativa das tecnologias apresentadas nesse artigo com outras soluções do gênero que acompanham os quatro *kits* de robótica educacional mais utilizados nas escolas do Brasil.

O *software* que acompanha o Kit Robótica Fácil (2003), diferentemente do ProgameFácil, é baseado numa interface de linhas de comando, apesar de só funcionar no ambiente Windows. O programa que acompanha esse *kit* apresenta uma quantidade menor de recursos, apenas possibilitando ligar ou desligar algum dispositivo eletrônico, tais como motor ou lâmpada, e esperar alguma quantidade de tempo. Essa limitação da linguagem pode ser atribuída à arquitetura eletrônica simples desse produto.

O ambiente de programação nativo do Kit Super Robby (2004), apesar de ser baseado numa interface visual, possui limitações no desenvolvimento de modelos. Seu ambiente não permite que sejam utilizadas estruturas condicionais para tomada de decisão, por exemplo, para ligar um determinado motor, caso um sensor identifique a presença de luz. Essa restrição de funcionalidades desse *software* parece ser reflexo dos poucos recursos de *hardware* disponibilizados neste *kit*.

O Kit GoGo Board (2004) não apresenta um ambiente de programação com foco educacional nativo para o seu *hardware*. Os idealizadores desse *hardware* disponibilizam pela Internet bibliotecas prontas para controlar esse *kit* através de alguns ambientes comerciais de programação, tais como Visual C++ e Visual Basic.

O Lego Mindstorms (2005) é acompanhado do *software* RoboLab (2004) para controlar seu *hardware*. Esse é um dos ambientes mais completos para programação visual do *kit* de robótica da Lego. Entretanto, a versão *standard* do RoboLab não apresenta algumas funcionalidades que podem ser almejadas por algumas escolas, por exemplo, a simulação.

Uma importante característica tecnológica do *software* ProgameFácil, em relação às soluções apresentadas acima é o recurso de simulação incorporado à ferramenta. Entre outras vantagens, tal característica, minimiza a necessidade de se possuir um *hardware* para cada grupo de alunos, contribuindo assim para a redução de custos na montagem de laboratórios de robótica. Entretanto, é importante ressaltar que se deve ter um número de *kits* suficientes para que todos os alunos participem do processo de construção/montagem e controle do *hardware* de robótica educacional.

Assim, de forma sintética, apontamos os seguintes itens como principais diferenciais da linguagem ProgameFácil: 1) ambiente de programação visual para *kits* de robótica com foco educacional; 2) sistema aberto; 3) recursos de simulação no ambiente de programação; e 4) projeto e implementação totalmente nacionais.

Conjeturamos, num contexto sócio-técnico, que as soluções aqui apresentadas poderão ser utilizadas como base para novos desenvolvimentos, permitindo-nos também vislumbrar uma maior utilização da robótica com fins pedagógicos por segmentos da sociedade ainda não atingidos com essas tecnologias, potencializando assim o acesso à informação e ao conhecimento, e o uso efetivo dessas tecnologias por parte dos estudantes brasileiros (Baranauskas e Souza 2006; Carvalho et al. 2007).

4. Considerações Finais e Trabalhos Futuros

A relevância dessa pesquisa está na fusão de conhecimentos teóricos de distintas áreas das engenharias e ciência da computação que, integradas, proporcionaram a implementação de um *kit* de robótica com fins educacionais de baixo custo composto de um *hardware* (conhecido como RoboFácil) e o *software* ProgameFácil.

O Kit de Robótica Educacional RoboFácil e suas características o fazem diferente da maioria das soluções existentes no mercado brasileiro. O Kit visa atender a um público carente de alternativas nessa área (alunos do Ensino Médio). Não descartando, todavia, outros segmentos do ensino.

A linguagem ProgameFácil descrita neste trabalho foi originalmente projetada para utilização com o Kit RoboFácil. Entretanto, dada as suas características de implementação é possível integrá-la a diferentes projetos de *hardware* de robótica educacional, ou mesmo empregá-la em situações de sala de aula voltadas ao Ensino Técnico/Profissionalizante, Superior e de Pós-Graduação.

Conjetura-se a possibilidade de utilização deste Kit integrado às atividades curriculares de disciplinas das Engenharias, Ciência da Computação, Física e Matemática, visto que metodologias didáticas que privilegiem o uso de artefatos tecnológicos em situações de ensino, abrem oportunidades ao discente de relacionar aspectos conceituais com situações do dia-a-dia. Busca-se com isso a ampliação da capacidade de entendimento dos fenômenos estudados através de atividades que motivem a participação dos usuários no processo de produção de conhecimento.

Experimentos, inicialmente realizados com alunos de Pós-Graduação em Informática, demonstraram o potencial de aplicação das soluções aqui descritas. Esses testes permitiram ao nosso grupo de pesquisa obter uma visão mais prática da utilização desses artefatos digitais em ambientes reais de ensino, nos dando um *feedback* de algumas melhorias a serem implementadas.

Como trabalhos futuros propomos a realização de estudos-piloto para que sejam pensadas propostas pedagógicas que possam explorar as tecnologias aqui disponibilizadas, no sentido de formar cidadãos melhor preparados a lidar com tecnologias no seu cotidiano.

Referências Bibliográficas

- Alves, A.C., Blikstein, P. e Lopes, R.D., 2005. Robótica na periferia? Uso de tecnologias digitais na rede pública de São Paulo como ferramentas de expressão e inclusão. Anais do XXV Congresso da Sociedade Brasileira de Computação.
- Baranauskas, M.C.C. e Souza, C.S., 2006. Desafio 4: Acesso Participativo e Universal do Cidadão Brasileiro ao Conhecimento. Computação Brasil, ano VII, número 23.

-
- Carvalho, A.R., Mantovani, O., Dias, M.H.P. e Liesenberg, H.K.E., 2007. Apenas Acesso Participativo e Universal ao Conhecimento? Anais do XXVII Congresso da Sociedade Brasileira de Computação.
- Chella, M.T., 2002. Ambiente de Robótica para Aplicações Educacionais com SuperLogo. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Estadual de Campinas.
- d’Abreu, J.V.V., Gonçalves, L.M.G., Garcia, M.F. e Garcia, L.T.S., 2002. Uma Abordagem Prático-Pedagógica para o Ensino de Robótica em Ciência e Engenharia de Computação. Anais do XIII Simpósio Brasileiro de Informática na Educação.
- d’Abreu, J.V.V., 1999. Desenvolvimento de Ambientes de Aprendizagem Baseados no Uso de Dispositivos Robóticos. Anais do X Simpósio Brasileiro de Informática na Educação.
- Gogo Board, 2004. GoGo Board. Disponível em: <<http://www.gogoboard.org>>. Acesso em: 30 abr. 2004.
- Lego Mindstorms, 2005. Lego Mindstorms. Disponível em: <<http://mindstorms.lego.com>>. Acesso em: 06 set. 2005.
- Miranda, L.C., 2006. RoboFácil: Especificação e Implementação de Artefatos de Hardware e Software de Baixo Custo para um Kit de Robótica Educacional. Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro.
- RoboLab, 2004. RoboLab Software. Disponível em: <<http://www.lego.com/eng/education/mindstorms/home.asp?pagename=robolab>>. Acesso em: 30 abr. 2004.
- Robótica Fácil, 2003. Robótica Fácil. Manual do Kit Robótica Fácil.
- Santos, C.F. e Menezes, C.S., 2005. A Aprendizagem da Física no Ensino Fundamental em um Ambiente de Robótica Educacional. Anais do XXV Congresso da Sociedade Brasileira de Computação.
- Sasahara, L.R. e Cruz, S.M.S., 2007. Hajime – Uma nova abordagem em robótica educacional. Anais do XXVII Congresso da Sociedade Brasileira de Computação.
- Schons, C., Primaz, E. e Wirth, G.A.P., 2004. Introdução a Robótica Educativa na Instituição Escolar para alunos do Ensino Fundamental da disciplina de Língua Espanhola através das Novas Tecnologias de Aprendizagem. Anais do I Workshop de Computação da Região Sul.
- Steffen, H.H., 2002. Robótica Pedagógica na Educação: Um Recurso de Comunicação, Regulagem e Cognição. Dissertação (Mestrado em Comunicação) – Universidade de São Paulo.
- Super Robby, 2004. Super Robby. ARS Consult. Disponível em: <<http://www.arsconsult.com.br/produtos/srobby>>. Acesso em: 30 abr. 2004.
- Zilli, S.R., 2004. A Robótica Educacional no Ensino Fundamental: Perspectivas e Prática. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Catarina.