

Proposta de um sistema multiagente para o ensino de Programação com *Mastery Learning*

Ceres Germanna Braga Morais, Thiago Jobson Barbalho

¹Departamento de Informática – Universidade do Estado do Rio Grande do Norte.
Avenida Antônio Campos, Bairro Costa e Silva, Mossoró/RN, CEP 59610-090.

{ceresgbmorais, thiagojobson}@gmail.com

Abstract. *This paper presents a proposal for a multiagent system that uses an online judge and Mastery Learning concepts to help in the process of learning Programming. Composed of a layer between student and judge, this system could be able to validate knowledge acquired by the student and identify students presenting learning difficulties in order to help them in the process of master the knowledge via feedbacks like tips, tutorials, videos etc.*

Resumo. *Este trabalho apresenta uma proposta de sistema multiagente para auxiliar o ensino de Programação utilizando um juiz online e técnicas de Mastery Learning. Composto de uma camada entre o estudante e o juiz, tal sistema seria capaz de validar o conhecimento adquirido pelo aluno e identificar alunos com dificuldades de aprendizado e auxiliá-lo no processo de domínio do conhecimento utilizando-se de feedbacks como dicas, tutoriais, vídeos e outros tipos de conteúdo extraclasse.*

1. Introdução

O ensino de Programação é fundamental nos cursos da área tecnológica por ser o passo inicial para o desenvolvimento do raciocínio lógico, e, por consequência, para a introdução dos conceitos e prática da programação. A disciplina destaca-se ainda por exigir do docente, e seus auxiliares, uma forte demanda de interação a fim de atender, acompanhar, mediar e avaliar individualmente os alunos [Raabe et al 2015]. No geral, as disciplinas que necessitam da prática de uma grande quantidade de exercícios para a aprendizagem dos conteúdos - onde a resolução de um exercício necessita da execução de um conjunto de passos - apresentam as maiores dificuldades para parte dos alunos. Assim, muitos alunos não conseguem acompanhar o ritmo das sequências de exercícios com nível de complexidade e recursos computacionais crescentes passadas pelo professor [Iepsen, 2013].

Diante disto, surge a necessidade de se desenvolver ferramentas que apoiem tanto professores quanto estudantes no tocante do ensino e aprendizado. No entanto, em relação ao ensino de Programação propriamente dito, o contexto de sala de aula geralmente é bastante heterogêneo: alguns alunos nunca viram algoritmos, e outros já trabalham com programação. Logo, cabe ao professor verificar os diferentes níveis de conhecimento entre os alunos em uma turma muitas vezes numerosas. Nesse sentido, este artigo apresenta a proposta de um ambiente de aprendizagem que fundamenta-se na

teoria *Mastery Learning* para proporcionar o nivelamento e equilíbrio entre os alunos, ao passo que cada aluno volta-se a tarefas e conteúdos de acordo com seu nível de conhecimento.

Vale salientar que a ferramenta ainda se encontra em fase de desenvolvimento e o que será apresentado neste artigo é um protótipo ainda não validado. Para melhor entendimento e para justificar a necessidade de desenvolvê-la, este trabalho está assim organizado: a Seção 2 apresenta a contextualização; a Seção 3 traz a proposta em si; e por fim as Considerações Finais são apresentadas.

2. Contextualização

As seguintes subseções apresentam um breve conceito sobre conhecimentos aplicados nesta pesquisa.

2.1 *Mastery Learning*

Mastery Learning (Aprendizagem por Maestria) é uma estratégia educacional na qual as habilidades e conhecimentos dos alunos são rigorosamente confrontados com padrões predeterminados de aprendizagem. Assim, estudantes não podem avançar para um próximo módulo, estágio, nível de conhecimento até que alcancem um designado nível de maestria [Eppich et al 2015]. É baseado na filosofia de que qualquer estudante pode aprender qualquer conteúdo se este for dado de forma sistemática; se este estudante receber instruções quando demonstrar dificuldades de aprendizado e se for dado tempo e oportunidades suficientes para que este alcance o domínio do conteúdo [Sahare 2010].

No *Mastery Learning*, as métricas que determinam se um estudante conquistou ou não um nível de domínio são extremamente importantes para o aprendizado, uma vez que o seu objetivo é garantir que ninguém seja deixado para trás e que todos estejam bem preparados para ter sucesso em fases subsequentes da formação. Se estes padrões são muito baixos, os alunos não estarão bem preparados; se os padrões são muito altos, os alunos irão despender um esforço desnecessário que poderia ser melhor gasto em outros objetivos educacionais [Yudkowsky et al 2015].

Este tipo de estratégia pode ser utilizada para dominar quase todos os tipos de conhecimento, mas se encaixa perfeitamente com o ensino de Matemática. O conhecimento matemático é construído sobre ele mesmo, então, para que um aluno possa ter sucesso, é preciso que este possua um nível de domínio em todas as suas unidades [Sahare 2010][School of Education 2010]. Esta característica também pode ser encontrada em ensino de Programação: conjunto de passos que levam à solução de um problema em um número finito de etapas. Estes passos são uma junção de elementos (variáveis, condicionais, entrada e saída, estruturas de repetição, funções, etc.) que precisam ser perfeitamente utilizados em um conjunto lógico para que se possa chegar à solução.

O trabalho de Morais et al (2014) buscou quantificar a satisfação de estudantes que cursaram uma disciplina de introdução à Programação em uma modalidade de *Mastery Learning* com ensino personalizado e utilização de testes automatizados para

auxiliar na aprendizagem de exercícios. Como resultado, o estudo concluiu que os alunos ficaram satisfeitos com a metodologia utilizada – principalmente com o feedback automático realizado pelos testes – e são favoráveis a uma expansão para outras disciplinas. Também, elementos do *Mastery Learning* podem ser encontrados no trabalho de Iepsen et al (2010) em que propuseram uma abordagem para evitar a evasão dos alunos de cursos da área de Computação com o desenvolvimento de um sistema web que realiza a personalização dos exercícios de algoritmos. O sistema recomenda exercícios de acordo com o nível de conhecimento e habilidades adquiridas pelo aluno.

Posteriormente, Iepsen (2013) realizou uma pesquisa para detectar os alunos que evidenciam sinais de frustração em atividades de ensino e de aprendizagem na área de Algoritmos, para então, auxiliá-los com ações proativas de apoio. Tal apoio consiste na exibição de um tutorial com a resolução passo a passo do exercício no qual o aluno apresenta dificuldades e na recomendação de um novo exercício com níveis de complexidade mais lineares aos conceitos trabalhados até aquele ponto de vista.

Os resultados dos experimentos destes trabalhos mostraram que evidências como, o alto número de tentativas de compilação de um programa sem sucesso, o grande número de erros em um mesmo programa ou a quantidade de tempo gasto na tentativa de resolver, podem estar relacionados ao estado de frustração do aluno. Também Raabe et al (2015) apresentaram uma abordagem para a construção de um gerador de dicas sobre os erros cometidos pelos estudantes, usando como base uma combinação de casos de testes aplicados ao programa a ser avaliado e uma análise estrutural do código, detectando padrões de erros comuns realizados por estudantes.

O objetivo deste experimento foi avaliar a influência das dicas emitidas pelo corretor nas ações tomadas pelos estudantes para corrigir o erro diagnosticado, bem como avaliar se este feedback ajuda o estudante a resolver exercícios propostos. A avaliação obtida pelos autores do trabalho foi de que “o mecanismo de correções e geração de dicas proposto teve um impacto positivo e auxiliou os estudantes na resolução dos exercícios” e a “melhorar a legibilidade do código”, o que certamente leva a crer que utilizar este método para o ensino de Programação pode ser de grande relevância.

2.2 Informática na Educação

Informática na Educação significa a inserção de recursos digitais no processo de aprendizagem dos conteúdos curriculares de todos os níveis e modalidades de educação, enfatizando que o professor da disciplina curricular deva ter conhecimento sobre os potenciais educacionais do computador e ser capaz de alternar adequadamente atividades tradicionais de ensino/aprendizagem e atividades que o utilizem.

Segundo Martins (2012), é muito comum as escolas apresentarem seus projetos de inserção a tecnologia com a justificativa de que esse trabalho é importante por ser a realidade dos estudantes de hoje em dia. A geração atual, da qual estes estudantes fazem parte, possui facilidade em utilizar ferramentas digitais e em interagir em ambientes virtuais como smartphones, redes sociais, blogs, sites, etc. Porém, isso não significa que eles possuem facilidade em utilizar essas ferramentas no ambiente escolar.

2.3 Sistemas Multiagente

Segundo Juchem e Bastos (2001), um agente é uma entidade de software com capacidade de resolução de problemas, que exhibe um comportamento autônomo e proativo, que está situado em algum ambiente, e a partir do qual percebe alterações sendo capaz de realizar ações para alcançar os objetivos de projeto. Inserido nesta visão, definem o agente como tendo as seguintes propriedades:

- Autonomia – executam a maior parte de suas ações sem interferência direta de agentes humanos ou de outros agentes computacionais, possuindo controle total sobre suas ações e estado interno;
- Habilidade social – por impossibilidade de resolução de certos problemas ou por outro tipo de conveniência, interagem com outros agentes (humanos ou computacionais), para complementarem a resolução de seus problemas, ou ainda auxiliarem outros agentes. Disto surge a necessidade de que os agentes tenham capacidade para comunicar seus requisitos aos outros e um mecanismo decisório interno que defina quando e quais interações são apropriadas;
- Capacidade de reação – percebem e reagem às alterações no ambiente em que estiverem inseridos;
- Capacidade proativa – agentes, do tipo deliberativo, além de atuar em resposta às alterações ocorridas em seu ambiente, apresentam um comportamento orientado a objetivos, tomando iniciativas quando julgarem apropriado.

3. Proposta

O URI Online Judge é um projeto que vem sendo desenvolvido na URI – Universidade Regional Integrada - Campus de Erechim, desde 2011. É um portal que contém problemas no estilo do ICPC (International Collegiate Programming Contest) da ACM (Association for Computing Machinery) e que fornece ao usuário um juiz online¹ para testar suas soluções a estes problemas. O objetivo inicial da criação do portal foi desenvolver funcionalidades que oferecessem uma alternativa ao método tradicional de ensino de Algoritmos e Programação, conforme proposto pelos autores em [Tonin et al. 2015].

A proposta deste trabalho é desenvolver um sistema que implemente o conceito de *Mastery Learning* em disciplinas de Programação utilizando questões do URI Online Judge como forma de validar o domínio de conhecimento dos alunos, utilizando uma arquitetura multiagente. Composto de uma camada entre o estudante e o juiz online URI, tal sistema seria capaz de identificar alunos com dificuldades de aprendizado e auxiliá-lo no processo de domínio do conhecimento utilizando-se de feedbacks como nos trabalhos de Iepsen et al (2010) e Iepsen (2013) e Raabe (2015): dicas, tutoriais, vídeos e outros tipos de conteúdo extraclasse.

¹ Juiz online verifica automaticamente se um algoritmo é capaz de resolver um problema proposto.

Na Figura 1 é possível ter uma visão abstrata da arquitetura do sistema que é composta de um servidor que intermedia a ligação entre a interface, utilizada pelo aluno, e o URI Online Judge, que é o repositório de questões. Tendo quatro agentes (explicados a seguir) ativos no servidor que monitoram o estado do ambiente, podendo, assim, executar ações visando atingir o seu respectivo objetivo.

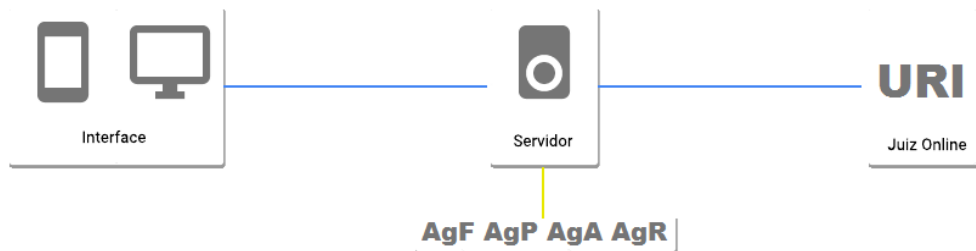


Figura 1: Arquitetura do sistema.

Fonte: Próprio autor (2017)

O sistema multiagente proposto para essa pesquisa, a priori, possui os agentes: Agente Facilitador (AgF), que terá o intuito de acompanhar o estudante, motivando-o, e percebendo suas atitudes, tais como permanência no ambiente de aprendizagem, participação; Agente Pedagógico (AgP) que irá atribuir tarefas ao discente, de acordo com o nível de domínio no qual este está inserido; Agente Avaliador (AgA), que a partir de métricas pré-definidas, irá avaliar o nível de maestria do estudante, de forma a permitir ou não, dependendo dos resultados obtidos, que este avance de nível de aprofundamento dos conteúdos vistos em sala de aula; e Agente de Retorno (AgR), que terá a função de dar feedback ao aluno e professor, quanto ao grau de excelência que o aluno está percebendo ao longo da disciplina, além de verificar se existe discrepância entre os vários estudantes envolvidos no processo de aprendizagem.

Como interface para utilização do sistema, o aplicativo de mensagens Telegram foi escolhido por ser um software popular entre usuário de smartphone, possuir clientes para os principais sistemas móveis, web e desktop e possuir uma API (Application Programming Interface) pública. Fazendo uso desta API, podemos usar toda a robustez da arquitetura do aplicativo a nosso favor para troca de mensagens texto, imagens, vídeos, etc. Dessa forma, delegando ao Telegram toda a troca de mensagens entre os alunos e o sistema, permite que nós foquemos no desenvolvimento dos outros componentes desta pesquisa.

4. Considerações Finais

Este trabalho apresentou uma proposta de um sistema multiagente que implementa a técnica de *Mastery Learning* para auxiliar no ensino/aprendizagem em disciplinas de Programação. Embora esta pesquisa ainda esteja em fase de desenvolvimento, trabalhos anteriores já demonstraram que sistemas que fazem um acompanhamento individual e são capazes de apresentar feedbacks para auxiliá-los em sua dificuldade de aprendizado

em Programação apresentaram um impacto positivo. Os próximos passos dessa pesquisa são: desenvolver a arquitetura multiagente; desenvolver o sistema proposto; validar o sistema em turmas de Programação, realizando, através de estudo de caso, experimentos pequenos e contínuos; estabelecer métricas para o nivelamento dos alunos; apresentar resultados obtidos à comunidade acadêmica.

Referências

- Dicheva, D., Dichev, C., Agre, G., Angelova, G. “Gamification in Education: A Systematic Mapping Study”, *Educational Technology & Society*, 18 (3), 2015. p. 75-88.
- Eppich, W. J., Hunt, E. A., Duval-Arnould, J. M., Siddall, V. J., Cheng, A. (2015) “Structuring Feedback and Debriefing to Achieve Mastery Learning Goals”, *Academic Medicine*, Vol. 90, No. 11, Nov 2015.
- Fardo, M. L. (2013) “Conjectura: Filos. Educ.”, v. 18, n. 1, p. 201-206, Jan/Abr.
- Iepsen, E. F M., Bercht, E., Reategui, E. (2010) “Persona-Algo: Personalização dos Exercícios de Algoritmos auxiliados por um Agente Afetivo”, In: Simpósio Brasileiro de Informática da Educação (SBIE).
- Iepsen, E. F. (2013) “Ensino de Algoritmos: Detecção do Estado Afetivo de Frustração para Apoio ao Processo de Aprendizagem”, Tese (Doutorado em Informática na Educação) – Universidade Federal do Rio Grande do Sul. 157 p.
- Juchem, M., Bastos, R. M. (2001) “Engenharia de Sistemas Multiagentes: Uma Investigação sobre o Estado da Arte”, *Tecninal Report Series*, No. 14, 2001.
- Martins, R. dos S. (2012) “O Uso das Tecnologias de Informação e Comunicação na Educação”, <http://escolabakhita.com.br/educacao/o-uso-das-tecnologias-de-informacao-e-comunicacao-na-educacao>, Mar 2017.
- Morais, L. A. de M., Figueiredo, J. C. A, Guerrero, D. D. S. (2014) “Students Satisfaction with Mastery Learning in an Introductory Programming Course”, In: Simpósio Brasileiro de Informática na Educação (SBIE).
- Raabe, A. L. A., Jesus, E. A., Hodecker, A., Pelz, F. (2015) “Avaliação do Feedback Gerado Por um Corretor Automático de Algoritmos”, In. Simpósio Brasileiro de Informática na Educação (SBIE), p. 358-366.
- Sahare, Namita S. (2010) “Mastery Learning”, <https://pt.slideshare.net/techcreation/mastery-learning-4798261>, Mar 2017.
- School of Education, (2010) “Behaviour Modification Model”, <https://pt.slideshare.net/competents2011/mastery-learning-7354062>, Mar 2017.
- Tonin, N. A., Selivon, M., Bez, J. L. (2015) “URI Online Judge Academic: Integração e Consolidação da Ferramenta no Processo de Ensino/Aprendizagem”.
- Youdkowsky, R., Park, Y. S., Lineberry, M., Knox, A., Ritter, E. Matthew (2015) “Setting Mastery Learning Standards”, *Academic Medicine*, Vol. 90, No. 11, Nov 2015.