

Desenvolvimento de um Chatterbot para Área de Medição de Software

José L. Silveira^{1,2}, Marcello Thiry^{1,2}

¹Centro de Ciências Tecnológicas da Terra e do Mar – Universidade do Vale do Itajaí (UNIVALI) – Campus São José
Rod. SC 407, Km 04 - 88122-000 – Sertão do Imaruim – São José – SC – Brasil

²Laboratório de Qualidade e Produtividade de Software (LQPS) – Universidade do Vale do Itajaí (UNIVALI) – Campus São José
Rod. SC 407, Km 04 - 88122-000 – Sertão do Imaruim – São José – SC – Brasil
{joseluisilveira2,marcello.thiry}@gmail.com

Abstract. *Currently there is a difficulty for students in learning software measurement the way it is taught, so tools in order to assist in the teaching of this subject are welcome. This paper presents the development of a chatterbot to help the learning of concepts in the software measurement area. The chatterbot was developed for the web platform using the Java programming language and AIML technology.*

Resumo. *Atualmente existe uma dificuldade dos alunos em aprenderem medição de software da forma como ela é ensinada, sendo assim ferramentas com o objetivo de auxiliar no ensino deste tema são bem vindas. Este trabalho apresenta o desenvolvimento de um chatterbot para auxiliar a aprendizagem de conceitos da área de medição de software. O chatterbot foi desenvolvido para a plataforma web utilizando a linguagem de programação JAVA e a tecnologia AIML.*

1. Introdução

A medição de software é um tema que vem ganhando destaque nos dias atuais, e se torna parte cada vez mais importante no contexto da Engenharia de Software, visto que ela tem auxiliado a melhorar a qualidade e produtividade dos processos através de um mecanismo de *feedbacks* [ROCKENBACK 2003]. Medir é fundamental em qualquer disciplina de engenharia, e a Engenharia de Software não é exceção, mas apesar da sua importância, infelizmente a medição está longe de ser comum na Engenharia de Software [PRESSMAN 2006]. Alguns dos motivos que levam as empresas a não aplicarem a medição na prática, ou que levam ao insucesso quando aplicadas, são: falta de objetivos bem definidos, incapacidade de analisar e usar os dados, falta de motivação dos envolvidos, falta de comunicação e falta de profissionais capacitados a realizarem a medição e análise [WANGENHEIM 2006].

Um dos motivos para a dificuldade das pessoas em medição de software é a forma como a medição é ensinada. Os processos tradicionais de capacitação dentre eles, cursos, treinamentos e aulas em instituições de ensino superior, não enfatizam o exercício prático de medição [WANGENHEIM *et al.* 2009]. Além disso, esses cursos geralmente não possuem tempo suficiente para que os alunos recebam uma aprendizagem sólida de toda a teoria que envolve a medição de software fazendo com

que eles aprendam apenas os conceitos básicos desta disciplina [OTT 2005]. E ainda os materiais didáticos destes cursos não apresentam um fundamento sólido da teoria da medição [OTT 2005]. Isto acaba reduzindo o interesse dos alunos dando-lhes a impressão de que é um tema difícil e complexo [HOCK & HUI 2004 *apud* WANGENHEIM *et al* 2009].

Para auxiliar no ensino de medição de software este trabalho apresenta o desenvolvimento de um *chatterbot*¹. Um *chatterbot* pode contribuir para o ensino de medição de software, pois é uma ferramenta que pode estar sempre à disposição do usuário, respondendo perguntas em uma linguagem natural, como se o usuário estivesse conversando com um professor. Ele pode guiar o usuário na aquisição do conhecimento indicando endereços que podem ser acessados para obter informações mais detalhadas sobre determinados assuntos. E pode motivar os alunos na aprendizagem por ser uma ferramenta diferente de assimilação de conteúdos, onde quem comanda e direciona o diálogo é o próprio usuário [LEONHARDT 2005a].

Existem alguns *chatterbots* para ensino disponíveis na Internet [LEONHARDT *et al* 2003; SGANDERLA, FERRARI & GEYER 2003; LEITÃO 2004; SCHOPF 2004; LEONHARDT 2005b; ROTHERMEL & DOMINGUES 2007], mas não foi encontrado nenhum que aborda exclusivamente medição de software, sendo este o principal diferencial do trabalho desenvolvido.

O *chatterbot* foi desenvolvido para a plataforma *web* utilizando a linguagem de programação Java [SUN 2009] e a tecnologia AIML [ALICE AI FOUNDATION 2009]. Antes de começar o desenvolvimento o trabalho passou por uma etapa de modelagem. Este artigo apresenta somente o desenvolvimento e avaliação do *chatterbot*, assim como os resultados e conclusões obtidos.

2. Desenvolvimento

A etapa de desenvolvimento envolveu as etapas de construção, criação da base de conhecimentos e testes. Estas três etapas foram executadas de forma iterativa, ou seja, uma etapa não precisava estar completamente finalizada para passar para a outra etapa.

Para construção foi obtido a última versão do interpretador AIML Program D² disponível no site <<http://aitools.org>>. O formato da versão obtida foi o *source*, pois foi necessária realizar algumas alterações nos arquivos fontes importados do Program D para que eles pudessem suportar as necessidades que foram surgindo durante a construção. Algumas das necessidades que surgiram foram, calcular a idade do *chatterbot* (para quando o usuário perguntar sua idade), calcular a hora (para quando o usuário perguntar a hora), e calcular o dia (para quando o usuário perguntar que dia é hoje).

Na construção da base de conhecimentos primeiro buscaram-se adicionar conhecimentos que permitisse manter uma conversa com um humano, como saudações, conhecimentos sobre ele e conversas realizadas no dia a dia. O conhecimento sobre medição de software foi construído baseado na revisão bibliográfica realizada e abordou

¹ *Chatterbots* são programas de computador que tentam simular conversações humanas de uma forma que as pessoas tenham a impressão de estar conversando com outra pessoa [TEIXEIRA 2005].

² Interpretador AIML para a linguagem de programação Java.

conhecimentos gerais sobre o assunto e conhecimentos específicos sobre o método GQM³. Como resultado da revisão bibliográfica foi elaborado uma planilha com os principais pontos, colocando mais de uma definição para cada ponto e buscando defini-los em forma de conversa. Também foi colocado um arquivo para tratar desrespeito para fornecer uma resposta mesmo quando o usuário falar alguma coisa deste tipo. E para aperfeiçoar a base de conhecimentos foi criado um arquivo que contém categorias que trocam palavras sinônimas por uma única palavra, desta forma aumenta o número de perguntas que podem ser feitas para a mesma resposta sem que se tenha que adicionar novas categorias para tratar cada uma dessas perguntas separadamente.

A etapa de teste envolveu algumas pessoas da equipe do LQPS (Laboratório de Qualidade e Produtividade de Software) e foi utilizada a interface de teste desenvolvida, onde os testadores podiam sugerir respostas diferentes das recebidas em suas perguntas. Assim, o *chatterbot* pode evoluir bastante principalmente na parte de diálogo básico. Após a etapa de teste os resultados eram analisados e retornava-se para etapa de construção e criação da base de conhecimentos, repetindo este ciclo até a etapa de avaliação.

3. Avaliação

Para verificar se o *chatterbot* pode contribuir para o ensino de medição de software, foi realizado um experimento com alunos da disciplina de Inteligência Artificial do curso de Ciência da Computação da UNIVALI (Universidade do Vale do Itajaí) campus São José.

Neste experimento uma prova composta por dez questões sobre medição de software foi aplicada a sete alunos que não possuíam experiência nem conhecimento sobre o assunto. Para responder à prova os alunos podiam utilizar o *chatterbot* para auxiliá-los. Para cada questão os alunos deveriam caracterizar quanto o *chatterbot* ajudou a responder à questão. As questões da prova foram retiradas de um questionário de Avaliação Empírica sobre Aprendizagem em Medição de Software [KOCHANSKI 2009] que contém vinte e cinco questões sobre o assunto. Este questionário visava identificar o nível de aprendizado dos alunos conforme a taxonomia de Bloom [BLOOM *et al* 1972], que divide o nível de aprendizagem em três grandes grupos (Lembrança, Compreensão e Aplicação). Para este experimento foram utilizadas somente dez questões para permitir que o experimento pudesse ser aplicado no espaço de uma aula regular. As questões foram selecionadas de forma aleatória, buscando selecionar pelo menos duas questões de cada nível da taxonomia de Bloom. O resultado da prova aplicada pode ser conferido na Tabela 1.

Tabela 1. Resultado da prova aplicada no experimento

QUESTÕES	ACERTOS	ERROS	QUANTO O CHATTERBOT AJUDOU?			
			AJUDOU MUITO	AJUDOU	AJUDOU POUCO	NÃO AJUDOU
1	6	1		7		
2	4	3	2	5		
3	2	5	1	3	2	1

³ Método GQM (Goal / Question / Metric – Objetivo, Questão e Métrica) parte do pressuposto que para uma organização medir, ela deve primeiro especificar os objetivos a serem alcançados com a medição.

4	1	6			7
5	3	4		4	3
6	1	6			7
7	1	6		2	5
8	4	3	1		6
9	2	5			7
10	3	4			7

Analisando os resultados do experimento, percebeu-se que quando o *chatbot* conseguiu contribuir para responder as questões, a maioria dos alunos acertou. Isso mostra que ele foi coerente em suas respostas e que ele pode ajudar no aprendizado de medição de software. Percebeu-se também, que o seu conteúdo precisa ser bastante melhora, pois ele ajudou a responder poucas questões. Apesar disto os alunos gostaram do *chatbot* e fizeram algumas sugestões de melhoria para o mesmo, o que mostrou o interesse deles por esta ferramenta.

4. Conclusões

Este trabalho apresentou o desenvolvimento de um *chatbot* para área de medição de software. Este trabalho foi desenvolvido como Trabalho de Conclusão de Curso, onde se pretendia desenvolver uma ferramenta que conseguisse ajudar na aprendizagem de conceitos de medição de software. Foi desenvolvido um protótipo e este foi avaliado por meio de um experimento para verificar se um *chatbot* pode ser utilizado como ferramenta de apoio ao ensino deste tema.

A etapa de avaliação mostrou que o *chatbot* pode contribuir para o ensino de medição de software, mas precisa ser bastante melhorado. O seu conhecimento em medição de software ainda está bastante fraco e ele não consegue manter um diálogo com o usuário.

Como trabalho futuro, pretende-se melhorar a base de conhecimentos envolvendo mais pessoas, dentre elas um especialista em medição de software, trabalhando principalmente na parte de adicionar mais conhecimentos em medição de software e na parte de deixar as respostas mais curtas e simples possíveis.

Referências

- ALICE – ARTIFICIAL LINGUISTIC INTERNET COMPUTER ENTITY. **A.L.I.C.E. The Artificial Linguistic Internet Computer Entity**. Disponível em: <<http://www.pandorabots.com/pandora/talk?botid=f5d922d97e345aa1>>. Acesso em: 03/04/2009.
- BLOOM, B. S.; KRATHWOHL, D. R.; MASIA, B. B.; CUNHA, J. A. **Taxonomia de objetivos educacionais**. Porto Alegre: Globo, 1972.
- KOCHANSKI, D. **Avaliação Empírica da Aprendizagem em Engenharia de Software**. Dissertação (Mestrado em Computação) Centro de Ciências Tecnológicas da Terra e do Mar, Universidade do Vale do Itajaí, São José, 2009.
- LEITÃO, D. A. **Um chatbot para um ambiente de ensino de gerência de projetos**. Monografia (Graduação em Ciência da Computação), Centro de Informática, Universidade Federal de Pernambuco, Recife, 2004.

-
- LEONHARDT, M. D.; CASTRO, D. D. ; DUTRA, R. L. S. ; TAROUCO, L. M. R. Elektra: Um chatterbot para uso em ambiente educacional. In: **RENTE - Revista Novas Tecnologias na Educação – II Ciclo de Palestras sobre Novas Tecnologias na Educação**. Porto Alegre, 2003.
- LEONHARDT, M. D. **Um estudo sobre Chatterbots**. Trabalho Individual de Mestrado (Pós-Graduação em Ciência da Computação), Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2005a.
- LEONHARDT, M. D. **Doroty: um Chatterbot para Treinamento de Profissionais Atuantes no Gerenciamento de Redes de Computadores**. Dissertação (Pós-Graduação em Ciência da Computação), Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2005b.
- OTT, L. M. Developing Healthy Skepticism not Disbelief - Problems in Teaching Software Metrics. Como, 2005. 1 ST WORKSHOP ON METHODS FOR LEARNING METRICS AT THE 11TH IEEE SOFTWARE METRICS SYMPOSIUM. **Proceedings...** Como, 2005.
- PRESSMAN, R. S. **Engenharia de Software**. São Paulo: Makron Books, 2006.
- ROCKENBACH, R. **Framework para Métricas**. Dissertação (Pós-Graduação em Ciência da Computação), Programa Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2003.
- ROTHERMEL, A. ; DOMINGUES, M. J. C. S. Maria: um Chatterbot Desenvolvido para os Estudantes da Disciplina métodos e Técnicas de Pesquisa em Administração. SEGET - SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA, Resende, 2007. **Anais do...**, Resende, 2007.
- SCHOPF, E. C. **AGENTCHÊ - Agente de Conversação com Linguagem Regionalista Voltado ao Ensino de Tópicos de Redes de Computadores**. Monografia (Curso de Ciência da Computação), Centro de Tecnologia, Universidade Federal de Santa Maria, Santa Maria, 2004.
- SGANDERLA, R. B.; FERRARI, D. N.; GEYER, C. F. R. BonoBOT: Um *Chatterbot* para Interação com Usuários em um Sistema Tutor Inteligente. XIV SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, Rio de Janeiro, 2003. **Anais do...** Rio de Janeiro, 2003.
- SUN Microsystems. JAVA 1.6. Disponível em: <<http://java.sun.com/>>. Acesso em: 15/07/2009.
- TEIXEIRA, S. **Chatterbots – uma proposta para a construção de bases de conhecimento**. Dissertação (Mestrado em Informática), Departamento de Informática, Universidade Federal do Espírito Santo, Vitória, 2005.
- WANGENHEIM, C. G. V. **Curso: Medição e Análise**. Projeto PLATIC/Meta 1, Laboratório de Qualidade e Produtividade de Software, Universidade do Vale do Itajaí. São José, 2006.
- WANGENHEIM, C. G. V.; THIRY, M.; KOCHANSKI, D.; STEIL, L.; SILVA, D.; LINO J. Desenvolvimento de um jogo para ensino de medição de software. SBQS – SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE, Ouro Preto, 2009. **Anais do SBQS**, Ouro Preto, 2009.
-