

Uma abordagem de Engenharia de Requisitos para Softwares de Gestão do Aprendizado: o caso AMADeUs

Carla Silva, Stephany Vitorio, Mariana Maia

Departamento de Ciências Exatas, Centro de Ciências Aplicadas e Educação –
Universidade Federal da Paraíba (UFPB)
58.297-000 – Rio Tinto – PB – Brazil

{carla, stephany.vitorio, mariana.maia}@dce.ufpb.br

Abstract. *Requirements Engineering (RE) is the activity for understanding the user needs in the context of the problem to be solved, and the discovery and specification of the characteristics of the solution. A Learning Management System (LMS) is a type of Virtual Learning Environment which offers a range of facilities to store, distribute and manage learning content as well as record and report activities and student performance. This paper proposes the use of the i* framework as an ER approach for the development of LMSs.*

Resumo. *A Engenharia de Requisitos (ER) corresponde à atividade de entendimento das necessidades do usuário no contexto do problema a ser resolvido, bem como da descoberta e especificação das características da solução. Um Sistema de Gestão da Aprendizagem (SGA) é um tipo de Ambiente Virtual de Aprendizagem que oferece um conjunto de funcionalidades para armazenar, distribuir e gerenciar conteúdos de aprendizado, bem como registrar e relatar atividades e o desempenho do aluno. Este artigo propõe o uso do framework i* como abordagem de ER para o desenvolvimento de SGAs.*

1. Introdução

A Engenharia de Requisitos (ER) é a fase inicial da Engenharia de Software que realiza estudos sobre o contexto de uso do software e produz a especificação do software a ser desenvolvido [Sommerville e Sawyer 1997]. A ER considera não apenas aspectos técnicos, mas também envolve fatores organizacionais e sociais [Nuseibeh e Easterbrook 2000].

Há vários relatos de situações nas quais os softwares usados não estão adaptados à realidade do trabalho da organização, o que resulta no insucesso da execução dos processos organizacionais. De acordo com os estudos realizados, a incompatibilidade entre os softwares e os processos organizacionais leva ao surgimento de sentimentos de insatisfação, frustração, hostilidade, incapacidade para tomar decisões, depressão, descuido na execução das tarefas executadas e até sabotagem [Ramos, Berry e Carvalho 2005]. A situação não é diferente para o desenvolvimento de software de apoio a aprendizagem e pode tornar-se ainda mais complexa devido ao impacto que uma ferramenta computacional de ensino pode causar na aprendizagem dos alunos.

Os softwares de apoio a aprendizagem atuais propõem canais para mediar a interação e colaboração entre tutores e aprendizes por meio de troca de artefatos (envio, visualização e entrega de matéria nas mais variadas mídias) e mensagens instantâneas ou assíncronas (fóruns, chats, e-mail). É sabido que novas tecnologias afetam e alteram os processos de negócio existentes. Portanto, a ER precisa ajudar a compreender as necessidades dos stakeholders e identificar como um software pode reorganizar o trabalho e estabelecer novos processos [Ramos, Berry e Carvalho 2005]. Espera-se que ao realizar uma atividade de elicitação de requisitos apropriada, que considere os processos de negócio, resulte no desenvolvimento de aplicações que satisfaçam os seus usuários, contribuindo para a realização das tarefas de forma efetiva e eficiente. No desenvolvimento de um software de apoio a aprendizagem, uma abordagem de ER deve considerar, além das tarefas executadas, os fatores sócio-organizacionais relevantes para a concepção do software [Kujala et al. 2005].

Uma abordagem de Engenharia de Requisitos que permite realizar análises sócio-organizacionais do domínio do problema através do uso de modelos estratégicos do processo de negócio é o framework *i** [Yu 1997]. Tais modelos servem para o engenheiro de requisitos relacionar o ambiente organizacional do sistema com seus usuários, compreender o seu contexto e, a partir daí, projetar o sistema. Seguindo esta abordagem este artigo propõe o uso do framework *i** para elicitar as necessidades de professores e alunos, com relação a ferramentas de apoio ao aprendizado.

2. Uma abordagem de Engenharia de Requisitos para Sistemas de Gestão do Aprendizado

A complexidade da Engenharia de Requisitos está justamente na elicitação das necessidades dos clientes e usuários dentro do domínio do problema. Uma abordagem bastante utilizada na academia e na indústria é analisar e especificar as características do ambiente organizacional e do software a ser desenvolvido é o framework *i**. O *i** é uma abordagem orientada a objetivos [Yu, 1997] que permite descrever tanto dependências sociais e intencionais no ambiente organizacional como também atributos de qualidade e funcionalidades do software [Yu, 1997]. O *i** tem dois níveis de abstração: o modelo SD (*Strategic Dependency* – Dependência Estratégica) e o modelo SR (*Strategic Rationale* – Raciocínio Estratégico).

Para ilustrar o uso do framework *i** com uma abordagem adequada de ER para softwares de apoio a aprendizagem, vamos considerar o sistema de gestão do aprendizado AMADeUs. Este software caracteriza-se por um ambiente de ensino colaborativo, onde os professores e alunos podem interagir com o ambiente e entre si, sendo capazes de perceber as ações e atividades dos participantes [Gomes et al. 2007].

O AMADeUs é uma aplicação Open Source, orientada à integração com diversas mídias, tais como: jogos e simulações multi-usuário, os quais agregam recursos da realidade virtual; vídeos; conteúdo textual; áudio e imagens. Busca-se, com isto, explorar da melhor forma os canais da percepção humana e atender às diversas formas de aprendizagem dos usuários através das características inerentes a cada um destes recursos e aplicá-las no contexto da aprendizagem [Gomes et al. 2007].

O modelo SD apresenta os atores e suas dependências, enquanto que o modelo SR apresenta os detalhes internos dos atores. Na Figura 1 aparece o detalhamento do

ator AMADeUs que representa o software, mostrando seus elementos internos e suas relações com as dependências externas. No modelo SD um ator pode depender de outro para satisfazer um objetivo, para executar uma tarefa, fornecer um recurso ou satisfazer um softgoal. Softgoals estão relacionados a requisitos não-funcionais, enquanto objetivos, tarefas e recursos estão relacionados com funcionalidades do software [Yu, 1997]. O ator que depende do outro é chamado de Dependente enquanto que o outro ator que recebe a requisição é chamado de Dependee.

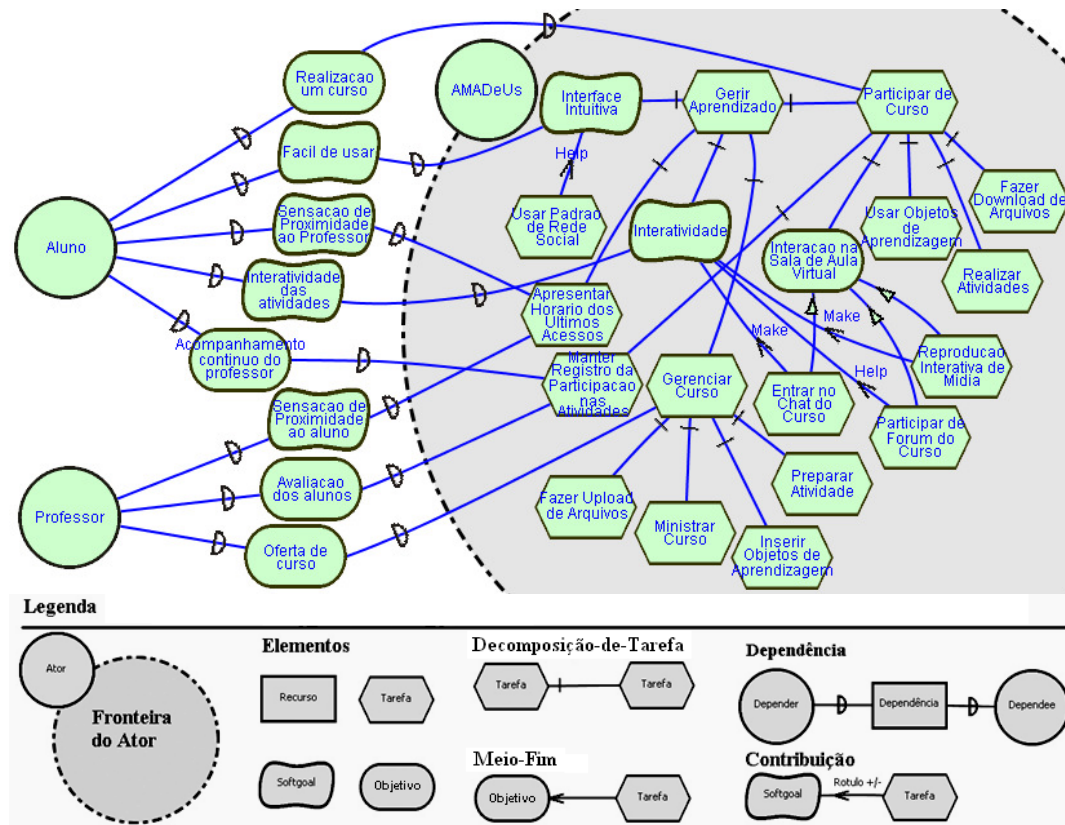


Figura 1. Modelo SR Parcial para o Sistema AMADeUs

O modelo SR é usado para: descrever os interesses e motivações dos participantes no processo, permitir a avaliação de possíveis alternativas para alcançar os softgoals, detalhar as razões existentes em relação as dependências entre atores. No modelo SR, onde os atores são detalhados, aparecem novos tipos de relacionamentos, tais como meio-fim, decomposição-tarefa e contribuição. Um relacionamento de decomposição-tarefa existe entre uma tarefa e suas partes, mostrando como uma tarefa é executada. Na Figura 1, o ator AMADeUs apresenta a tarefa raiz Gerir Aprendizado, que é refinada, através do relacionamento de decomposição-de-tarefa, no softgoal Interface Intuitiva e das tarefas Apresentar Horário dos Últimos Acessos, Gerenciar Curso e Participar de Curso. Esta última tarefa é refinada no objetivo Interação na Sala de Aula Virtual e nas tarefas Usar Objetos de Aprendizagem, Realizar Atividades e Fazer Download de Arquivos. Um relacionamento meio-fim descreve uma ligação de um elemento meio para alcançar um objetivo (fim). Um meio é uma alternativa para realizar um fim (normalmente um elemento do tipo objetivo) mostrando como este objetivo pode

ser atingido. Por exemplo, no ator AMADeUs (Figura 1), o objetivo Interação na Sala de Aula Virtual é refinado, através do relacionamento meio-fim, nas tarefas Entrar no Chat do Curso, Participar de Fórum do Curso e Reprodução Interativa de Mídia.

O relacionamento do tipo contribuição descreve uma contribuição de um meio (tarefa ou objetivo) para a realização de um fim (softgoal). Este tipo de relacionamento fornece um raciocínio qualitativo usando um esquema de valores para representar a contribuição (e.g., Help e Hurt significam Ajuda e Prejudica, respectivamente) [Yu, 1997]. Por exemplo, na Figura 1, a tarefa Usar Padrão de Rede Social contribui positivamente (ver o rótulo Help) para a satisfação do softgoal Interface Intuitiva.

3. Conclusão e Trabalhos Futuros

Neste artigo propomos o uso de uma abordagem de modelagem sócio-organizacional, chamada de framework i^* , o desenvolvimento de sistemas de apoio a aprendizagem. O trabalho apresentado ainda está em evolução, mas a partir da modelagem parcial do sistema AMADeUs, apresentado na Figura 1, observamos vários benefícios do uso dos modelos i^* no desenvolvimento de softwares de gestão da aprendizagem. Por exemplo, os modelos i^* representam como o sistema pretendido atende aos objetivos dos seus usuários, respondem por que o sistema é necessário, consideram várias alternativas para alcançar o mesmo objetivo e quais as implicações destas alternativas para os vários usuários, além de relacionar os requisitos funcionais e não-funcionais no mesmo modelo. Como trabalhos futuros, pretendemos propor um conjunto de atividades sistemáticas para construir modelos i^* que venha a desenvolver softwares de gestão da aprendizagem que sejam mais próximos às necessidades dos alunos e professores.

Referências

- Gomes, A. S., Nibon, R., Caparica, F., Lobato, L. L., Costa, M. C. AMADEUS: sistema de gestão do aprendizado de segunda geração. In: V Congresso Internacional de Tecnologia na Educação, Recife, 2007. Site do projeto: <http://www.amadeus.cin.ufpe.br/index.html/>. Último acesso: 07/10/2010.
- Kujala, S., Kauppinen, M., Lehtola, L., Kojo, T. The Role of User Involvement in Requirement Quality and Project Success. 13th International Conference on Requirement Engineering, IEEE Computer Society Press, 2005.
- Nuseibeh, B., Easterbrook, S. Requirement Engineering: A Roadmap, In: Finkelstein, A. C. W. (ed.): The Future of Software Engineering, IEEE Computer Society Press, 2000.
- Sommerville, I. e Sawyer, P. Requirement Engineering: A good practice guide, John Wiley & Sons, 1997.
- Ramos, I., Berry, M., Carvalho, J. Requirements engineering for organizational transformation. Information and Software Technology, 47(7), 479-495, 2005.
- Yu, E. Towards modelling and reasoning support for early-phase requirements engineering, Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (RE 97), Annapolis, USA, IEEE Computer Society, p. 226-235, 1997.