

Avaliando a utilização da Técnica i* no Processo de Ensino e Aprendizagem na Engenharia de Requisitos – Um Relato de Experiência

Victor Francisco Araya Santander

Universidade Estadual do Oeste do Paraná (UNIOESTE)

Rua Universitária 2069 – Cascavel – PR – Brazil

vfasantander@unioeste.br

***Abstract:** This paper presents an experience account using the i* organizational modeling technique for eight years on the teaching and learning of Computer Science for undergraduate students. This paper also presents lessons learned under a perspective of best practices in software engineering and requirements engineering.*

***Resumo.** Neste artigo apresenta-se um relato de experiência do uso da técnica de modelagem organizacional i* durante o período de oito anos no contexto de atividades de ensino-aprendizagem de alunos de graduação de um curso de Ciência da Computação. São apresentadas as lições aprendidas neste processo sob a perspectiva de boas práticas que devem ser adotadas na engenharia de software e engenharia de requisitos.*

1. Introdução

O desenvolvimento de um sistema computacional deve resultar em um produto que deve atender com a maior precisão e completude os anseios de clientes e usuários. Inúmeros fatores contribuem para o insucesso nesta atividade tais como profissionais mal preparados, falta de experiência destes profissionais no uso de técnicas da engenharia de software, utilização de processos de desenvolvimento inadequados, falta de preocupação em entender o problema do cliente e o domínio ao qual está vinculado este problema, pressões do cliente para concluir rapidamente o sistema computacional solicitado sem compreender ou valorizar o trabalho de engenharia necessários, entre outros aspectos. Desta forma, um dos grandes desafios para a engenharia de software está em propor técnicas, processos e ferramentas para apoiar o engenheiro de software a minimizar ou eliminar estes fatores. Contudo, não basta propor novas abordagens. É necessário avaliar qual é o grau de contribuição destas abordagens em relação a melhoria do processo e do produto de software. No contexto educacional, é importante avaliar como estas propostas podem contribuir na formação de estudantes (futuros profissionais) de cursos de computação em universidades.

Neste sentido, este trabalho apresenta um relato da experiência de oito anos de docência em um curso de graduação em Ciência da Computação do autor deste artigo na utilização da técnica de modelagem organizacional i* (lê-se iestrela) como ferramenta de apoio a elicitação, análise, negociação e validação de requisitos organizacionais, funcionais e não-funcionais de sistemas computacionais pretendidos. A técnica i* possibilita modelar as intencionalidades associadas a atores da organização na forma de dependências entre os mesmos. Também possibilita expressar as razões internas que cada ator possui visando satisfazer suas dependências. Adicionalmente, a técnica permite incluir o sistema computacional pretendido como um ator da

organização. Desta forma, o intuito neste artigo é realizar uma análise crítica dos pontos positivos e negativos percebidos durante o processo de ensino-aprendizagem da técnica *i** nesse contexto. Cabe destacar que esta avaliação é realizada no âmbito de disciplinas de graduação lecionadas bem como na orientação de acadêmicos que realizaram seus trabalhos de conclusão de curso utilizando a técnica estudada. O trabalho está estruturado conforme segue. Na seção 2 apresenta-se a técnica *i** e seus principais conceitos. Na seção 3 são apresentadas algumas características do processo educacional na engenharia de requisitos em nível de cursos de graduação na computação. Na seção 4 apresenta-se o relato de experiência da utilização da técnica *i** com estudantes de graduação e na seção 5 são realizadas as considerações finais do trabalho.

2. A Técnica *i**

A técnica *i** foi originalmente proposta por Yu [Yu 1995] e vem ao longo dos anos sendo utilizada em outras áreas além da engenharia de requisitos [Yu et al. 2011]. A principal vantagem desta técnica é que a mesma permite explorar as intencionalidades associadas aos atores de uma organização sob a forma de objetivos, tarefas, recursos e objetivos-*soft*. Assim, processos organizacionais podem ser compreendidos e modelados já considerando, por exemplo, o impacto da utilização de um sistema computacional na organização. Outro aspecto positivo da técnica *i** é que, a partir da mesma, é possível derivar artefatos de *software* orientados a objetos, como proposto em [Alencar 1999] [Santander e Castro 2002]. Para descrever as intencionalidades e motivações envolvendo atores em um ambiente organizacional são propostos dois modelos: O Modelo de Dependências Estratégicas (**SD**) e o Modelo de Razões Estratégicas (**SR**). O Modelo de Dependências Estratégicas é composto por **nós** e **ligações**. Os nós representam os **atores** no ambiente e as ligações são as **dependências** entre os atores. Por **ator** entende-se uma entidade que realiza ações para obter objetivos no contexto do ambiente organizacional. Atores dependem uns dos outros para atingir objetivos, realizar tarefas, e obter recursos no ambiente organizacional. O ator que depende de alguma forma de outro ator é chamado de **Depender** e o ator que atende e satisfaz o *Depender* é denominado de **Dependee**. O objeto ou elemento de dependência entre *Depender* e *Dependee* é denominado de **Dependum**. Portanto, haverá relacionamentos do tipo *Depender* -> *Dependum* -> *Dependee*. O modelo de Razões Estratégicas é um modelo complementar ao modelo de dependências estratégicas. Este modelo permite compreender e modelar de forma mais detalhada as razões associadas com cada ator e suas dependências. Enquanto o modelo de Dependências Estratégicas provê um nível de abstração, no qual modela-se somente os relacionamentos externos entre atores, o modelo de Razões Estratégicas permite uma maior compreensão a respeito das razões estratégicas de atores em relação a processos da organização e como os mesmos são expressos. O modelo de Razões Estratégicas auxilia no processo de Engenharia de Requisitos, permitindo que elementos de processos e as razões por detrás dos mesmos sejam expressos.

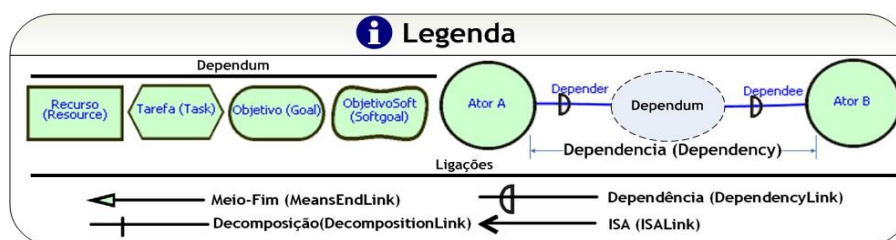


Figura 1. Elementos da técnica *i**

Na Engenharia de Requisitos, o modelo de Razões Estratégicas pode ser utilizado para compreender como sistemas estão relacionados/envolvidos em rotinas de atores da organização para gerar alternativas, bem como para modelar e suportar o raciocínio de atores organizacionais a respeito destas alternativas. A Figura 1 apresenta um resumo dos elementos utilizados na técnica i*.

3. Características do Processo Educacional na Engenharia de Requisitos

O processo de ensino-aprendizagem é complexo em qualquer área do conhecimento [Santander et al. 2007a]. Mais especificamente na área de engenharia de requisitos busca-se fazer com que o acadêmico de graduação de um curso na área de computação obtenha uma base teórico-prática sólida a qual lhe possibilite extrair e documentar de forma consistente requisitos expressos por clientes. Como apontado na literatura da área [Kotonya e Sommerville 1998] este processo de interação e iteração do engenheiro de requisitos com clientes, usuários e demais *stakeholders* acaba envolvendo aspectos associados a disciplinas da psicologia, administração e da própria computação. Assim não basta apenas dominar uma técnica específica de elicitação de requisitos, por exemplo, mas é necessário estudar e aplicar outros conteúdos das áreas mencionadas que auxiliem principalmente no contato com o cliente visando entender e dar uma solução a sua problemática. Entender as características inerentes as relações interpessoais, como também estudar princípios básicos da administração de empresas pode auxiliar fortemente o engenheiro de requisitos nas suas atividades.

Assim, um dos grandes desafios no processo educacional de estudantes de graduação que exercitam a engenharia de requisitos é fazê-los entender a importância de utilizar técnicas e processos que permitam inicialmente compreender em detalhes o problema para posteriormente elaborar soluções consistentes. Desta forma, se deseja que estudantes inicialmente analisem e compreendam detalhadamente os processos organizacionais que geram expectativas do cliente quanto ao desenvolvimento de um sistema computacional. A partir desta análise pode-se então detectar e avaliar a complexidade desses processos, relações entre os atores organizacionais, grau de entendimento dos atores em relação aos processos, grau de comprometimento dos atores em relação a uma possível solução computacional, entre outros aspectos. No nosso entendimento, cabe aos docentes que lecionam disciplinas relacionadas a Engenharia de Requisitos instigar nestes estudantes essa necessidade de análise e modelagem organizacional antes de prosseguir com o detalhamento dos requisitos funcionais e não-funcionais [Chung et al. 2000]. Outro problema está em como diminuir o *gap* entre a modelagem da organização incluindo seus objetivos estratégicos, e a modelagem de sistemas computacionais. Sabe-se que a partir do momento em que certos objetivos organizacionais passam a ser satisfeitos por sistemas computacionais, existe a necessidade de estabelecer e manter uma relação consistente entre estes elementos.

4. Relato de Experiência

O relato de experiência aqui apresentado é oriundo da coleta de informações do autor deste artigo atuando como docente de disciplinas de Engenharia de Software e Engenharia de Requisitos em um curso de Ciência da Computação nos últimos oito anos. Cada disciplina lecionada era integralizada em 136 horas-aula anuais, sendo 102 h/a teóricas e 34 h/a práticas no ano, o que equivale a 3 h/a teóricas e 1 h/a prática semanais. Cada turma era composta por em média 20 alunos. A experiência aqui relatada destaca as lições aprendidas do docente com base no desempenho da maioria dos estudantes. Destaca-se também que ocorreram variações no processo de

aprendizagem de uma turma para outra ao longo dos anos, mas as mesmas não invalidam as lições finais aqui relatadas. A base para o relato envolve aproximadamente 50 projetos¹ de pequeno e médio porte em empresas reais envolvendo cerca 160 alunos ao longo do período relatado. Cada projeto foi desenvolvido por equipes de em média 3 alunos. O cliente na sua maioria era uma empresa real que se comprometia em dedicar um tempo previamente acordado para apoiar a construção bem como revisar os artefatos gerados ao longo do processo de desenvolvimento. Os alunos envolvidos na sua maioria cursavam o 4° e 5° anos do referido curso. Em todas as disciplinas a técnica i* foi utilizada no início dos projetos. Também o estudo inclui 7 alunos de graduação que desenvolveram suas monografias finais de curso ou trabalhos de iniciação científica envolvendo propostas para a melhoria ou utilização da técnica i*. Seguem abaixo algumas questões que foram elaboradas a fim de orientar o relato. Na sequência de cada questão são descritas as lições aprendidas sob a perspectiva direta e subjetiva do docente das disciplinas.

Q1 - Os alunos utilizando a técnica i* melhoraram a percepção da necessidade de modelar o ambiente organizacional antes de iniciar o processo de elicitação e documentação dos requisitos associados ao sistema computacional?

Lições aprendidas: O fato de desenvolver projetos reais os quais na sua maioria envolveram domínios de aplicação não conhecidos pelos alunos e pelo professor, fez com que a compreensão do ambiente organizacional fosse uma necessidade para poder prosseguir com o projeto. Os modelos SD e SR gerados tinham como meta mostrar ao aluno e ao cliente os elementos organizacionais estratégicos envolvidos no problema a ser solucionado. Este conhecimento na maioria das vezes tácito, gradualmente ia sendo construído com a participação direta dos indivíduos mais interessados. Ao final, um acordo formal sobre o conhecimento modelado era estabelecido. A partir dessa análise do ambiente organizacional a maioria dos alunos evoluiu com mais segurança para as demais etapas da engenharia de requisitos e outras fases do processo de desenvolvimento de software.

Q2 - Os elementos definidos pela técnica i* como ator, dependências entre atores do tipo objetivo, tarefa, objetivo-soft e recurso (ver seção 2) permitiram aos alunos aumentar o grau de entendimento do ambiente organizacional no qual o software irá operar?

Lições aprendidas: Uma das vantagens da técnica i* é a definição dos atores, bem como dependências e razões estratégicas existentes em uma organização. A definição de **intencionalidade** associada a um ator em uma organização leva os alunos a averiguar quem ou o que possui intencionalidades no contexto organizacional. Essa averiguação faz com que os alunos detectem as pessoas envolvidas nos processos organizacionais e mais especificamente questionem as mesmas sobre suas “intenções” no processo. Este ator acaba percebendo que essas intencionalidades podem derivar dependências em relação a outros atores se o mesmo não for capaz de satisfazê-las de forma autônoma. Assim, os próprios alunos em um processo iterativo e incremental [Santander e Silva 2004] evoluem na modelagem organizacional descobrindo novos atores e gerando novas dependências. Em uma perspectiva mais específica, os alunos também introduzem um ator para representar o sistema computacional, explorando intencionalidades dos demais atores sociais em relação a este novo ator. Um dos problemas detectados neste processo é que os alunos inicialmente têm dificuldade em

1 Alguns destes projetos estão disponíveis em <<http://www.inf.unioeste.br/~victor/projetos/>>. Acessado em 30/05/2011.

perceber claramente as diferenças entre os tipos de dependências. Isto somente é sanado com aulas práticas nas quais exemplifica-se vários ambientes organizacionais explorando os tipos de dependências existentes.

Q3 - O fato de poder inserir o sistema computacional pretendido como um ator nos modelos SD e SR facilita o processo de detecção de requisitos funcionais e não-funcionais pelo estudante?

Lições aprendidas: Poucas técnicas de modelagem organizacional permitem a inserção explícita do sistema computacional na modelagem. Para um estudo mais detalhado sobre este assunto ver [Grando 2010]. Este aspecto positivo da técnica *i** é determinante para que o estudante entenda como o ambiente é afetado pela introdução de um novo sistema computacional bem como modele quais são os objetivos, objetivos-soft, tarefas e recursos que o sistema deve satisfazer. Isto implica em fazer com que o estudante visualize a organização e explore como algumas das dependências que podem estar alocadas a outros atores podem ser realocadas a um sistema computacional. Essa análise é complexa, integra todos os *stakeholders*, mas é facilitada pelo uso da técnica.

Q4 - Qual é a percepção dos alunos quanto ao reaproveitamento das informações existentes nos modelos SD e SR para documentar os requisitos do sistema computacional ?

Lições aprendidas: Os alunos são conduzidos no processo educacional da engenharia de requisitos a considerar como base para a definição do documento de requisitos do sistema os modelos *i**. São orientados a explorar e refinar todas as dependências nas quais um dos atores envolvidos é o sistema computacional pretendido. Juntamente com outros documentos da empresa e uso de outras técnicas da engenharia de requisitos [Kotonya e Sommerville 1998] o documento de requisitos é gerado. Tanto a descrição textual quanto diagramática de requisitos do sistema pode ser apoiada pelas informações existente nos modelos SD e SR. Por exemplo, a partir das dependências associadas ao sistema computacional tanto como *dependee* quanto como *depende*, pode-se gerar casos de uso em UML [Jacobson 1992] descrevendo assim os requisitos funcionais do sistema usando uma técnica padrão e amplamente utilizada no meio acadêmico e industrial. Cabe salientar que em vários dos projetos orientados, os alunos utilizaram a ferramenta JGOOSE [Vicente et al. 2009] para apoiar este processo de derivação de casos de uso a partir de modelos *i**. Este processo em detalhes pode ser visto em [Santander e Castro 2002]. Assim, observando os projetos orientados pôde-se perceber que o refinamento de tarefas internas ao ator sistema computacional motivou nos alunos a busca por refinamentos associados aos passos necessários para viabilizar a realização de uma tarefa e por consequência a dependência que a mesma satisfaz. Este aspecto mostrou-se extremamente positivo considerando que além da compreensão mais abrangente do ambiente organizacional o aluno pode explorar em um nível de abstração adequado a casos de uso, os requisitos que o sistema deve satisfazer. Neste sentido, é importante destacar que os próprios alunos encontraram níveis aceitáveis de refinamentos das tarefas internas a atores no modelo SR de forma a não comprometer a representação gráfica do modelo com elementos desnecessários e não estratégicos. Quando informações de mais baixo nível computacional começavam a ser representadas nos modelos SR, os próprios alunos tomavam decisões ao que de fato incluir para não perder a visão de alto nível possibilitada pelo modelo. Essas informações, contudo, eram introduzidas na descrição textual detalhada dos requisitos ou em outros modelos tipicamente pertencentes ao paradigma de desenvolvimento orientado a objetos.

Q5 - É possível estabelecer uma base arquitetural do sistema a partir dos modelos SD e SR ?

Lições aprendidas: Alguns projetos realizados apresentaram pontos positivos neste sentido. O sistema computacional inserido foi subdividido através do mecanismo “*Is part of*” gerando outros subsistemas representando uma parte modular do sistema pretendido. Esta subdivisão facilitou e entregou aos alunos em fases posteriores do processo de desenvolvimento, elementos importantes para gerar a decomposição funcional do sistema. Esta subdivisão tornou o processo de definição de dependências mais específico ao subsistema definido e, portanto, permitiu estabelecer estas dependências de forma mais precisa e localizada. Para os alunos, este aspecto da técnica foi considerado positivo auxiliando decisivamente no projeto arquitetural. Por exemplo, um projeto abordado tinha como foco o desenvolvimento de um sistema de apoio a leitura de deficientes visuais. O mesmo possuía quatro relacionamentos do tipo “*Is part of*” dos atores *Leitor, Ampliador, Serviço Remoto e RBC Remoto* para o ator *Xlupa* representando uma possível subdivisão em subsistemas.

Q6 – Qual é o grau de dificuldade de aprendizado apresentado pelos alunos em relação a técnica i*?

Lições aprendidas: As dificuldades recaíram principalmente no entendimento dos elementos dos modelos SD e SR. Contudo, após a realização de vários exercícios em aulas práticas a maioria dos alunos mostrou bom domínio da técnica. A possibilidade de utilizar informações dos modelos organizacionais gerados para outras fases do processo de desenvolvimento foi um fator decisivo para elevar a motivação no uso da técnica.

Q7 - Quais são as desvantagens apresentadas pelos alunos no uso da técnica ?

Lições aprendidas: As maiores desvantagens apontadas pelos alunos foram a falta de ferramentas computacionais mais estáveis e melhores do ponto de vista de usabilidade para construir os modelos organizacionais, bem como a falta de representação de sequencialidade no refinamento de tarefas no modelo SR. Também outro aspecto negativo apresentado foi a expansão elevada dos elementos diagramados dificultando o entendimento dos diagramas. Isto ocorria na maioria das vezes devido a que algumas informações não estratégicas eram adicionadas ao modelo. Os alunos foram orientados a somente modelar intencionalidades estratégicas para a organização e deixar as outras informações fora do modelo podendo representá-las, por exemplo, na descrição textual do artefato de documentos de requisitos. Contudo, naqueles ambientes organizacionais nos quais a quantidade de intencionalidades estratégicas dificultava a visualização em um único modelo, a orientação foi criar outros diagramas parciais mostrando os relacionamentos específicos a alguns dos atores envolvidos. Alguns trabalhos mais recentes que se aprofundam na solução destes problemas são apresentados em [Alencar et al. 2010] e [Lucena et al. 2009].

Q8 - Os trabalhos realizados em projetos de disciplinas, de monografias de conclusão de curso e de projetos de iniciação científica, permitiram aos alunos envolvidos melhorar a sua formação na área bem como propor melhorias em relação ao uso da técnica i*?

Lições aprendidas: Os trabalhos desenvolvidos pelos alunos de graduação do curso de Ciência da Computação melhoraram visivelmente a percepção dos mesmos em relação ao processo de engenharia de requisitos utilizando e propondo melhorias em relação a técnica i*. Desses trabalhos surgiram ferramentas computacionais para derivar casos de uso a partir de modelos i* [Briscke e Santander 2005] [Vicente et al. 2009], contribuições da modelagem i* para o processo de desenvolvimento de sistemas de apoio a deficientes visuais [Santander e Silva 2006] [Santander et al. 2007a], utilização

da técnica i^* no processo de evolução de sistemas legados [Santander et al. 2007], melhoria do processo de construção dos modelos i^* utilizando a teoria da atividade [Teixeira e Santander 2010], desenvolvimento de ferramentas de derivação de casos de uso com “aspectos” a partir de modelos i^* [Lizana et al. 2009], comparação da técnica i^* com outras técnicas de modelagem organizacional [Grando 2010], entre outros. Percebe-se que a formação dos alunos de graduação que utilizaram a técnica i^* em algum projeto durante o curso foi melhorada significativamente no que tange aos seguintes aspectos: em assumir como necessidade vital para o sucesso de um projeto a realização da modelagem organizacional do ambiente no qual o software será implantado; em considerar que modelos dessa natureza podem auxiliar na definição de outros artefatos necessários nas fases posteriores do processo de desenvolvimento; em colocar em prática o uso de técnicas de elicitação de requisitos para interagir com os atores do ambiente organizacional e extrair dos mesmos suas intencionalidades bem como detectar dependências existentes entre os mesmos; em reforçar algo difícil em estudantes de graduação destes cursos que envolve a valorização do trabalho antes do projeto, implementação e testes do sistema; em utilizar um meio mais maduro de modelagem fazendo com que clientes percebam muitos aspectos antes não considerados na sua problemática; em exercitar, no contexto de diversas metodologias de desenvolvimento de software tais como Scrum, XP, RUP, Espiral, Cascata, entre outras, a técnica i^* avaliando suas contribuições específicas considerando as características metodológicas associadas as mesmas.

5. Considerações Finais e Trabalhos Futuros

Neste trabalho foram apresentadas algumas lições aprendidas fruto da experiência vivenciada pelo autor deste artigo no contexto do uso da técnica i^* com alunos de graduação em Ciência da Computação. A lição mais importante neste contexto recai na necessidade de estimular nos alunos a busca do conhecimento utilizando meios que facilitem este processo. Técnicas de comprovado valor devem ser experimentadas e o docente deve ter claro quais são as contribuições do uso das mesmas na formação dos alunos. Especificamente em relação a técnica abordada neste artigo, verifica-se que a mesma expande a visão de mundo que o aluno tem do desenvolvimento de software, motivando os mesmos a interagir com clientes e usuários de forma organizada e com foco em informações estratégicas associadas a problemática apresentada. Percebe-se claramente que a estruturação proposta pela técnica i^* força alunos (futuros profissionais) e clientes a explicitar conhecimentos que poderiam passar despercebidos se utilizadas outras técnicas [Grando 2010]. Como trabalhos futuros pretende-se elaborar questionários que serão aplicados a alunos de graduação do referido curso, os quais permitirão realizar uma avaliação quantitativa de alguns dos pontos descritos na experiência relatada. Também pode-se estender os estudos para o nível de exercício da profissão dos egressos do curso, avaliando as contribuições do uso da técnica nesse contexto.

6. Referencias Bibliográficas

- Alencar, F., Mapeando a Modelagem Organizacional em Especificações Precisas. Recife - PE: Centro de Informática. Universidade Federal de Pernambuco. Dezembro, Tese de Doutorado (1999).
- Alencar, F. M. R., Castro, Jaelson F B., Lucena, M. J. N. R., Santos, E. B., Silva, C. T. L. L., Araujo, J., Moreira, A. M. D., Towards Modular i^* Models. In: 25th ACM Symposium on Applied Computing (SAC 2010), v. 1. p. 292-297, Sierre, Suíça. Proceedings of the 2010 ACM symposium on Applied Computing, (2010).

- Briscke, M., Santander, Victor Francisco Araya., Castro, J. F. B., GOOSE: Uma Ferramenta para Integrar Modelagem Organizacional e Modelagem Funcional In: Jornadas Chilenas de Computación - V Workshop Chileno de Ingeniería de Software, Valdivia, Chile. (2005).
- Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J., *Non-Functional Requirements in Software Engineering (Monograph)*, Kluwer Academic Publishers, 472 pp, (2000).
- Grando, F. L., Avaliando técnicas de modelagem organizacional no contexto de desenvolvimento de sistemas computacionais: uma abordagem empírica. Cascavel – PR: Universidade Estadual do Oeste do Paraná. Novembro, Monografia de Graduação (2010).
- Jacobson, I., *Object Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley (1992).
- Kotonya, G., Sommerville, I.: *Requirements Engineering: Processes and Techniques*. John Wiley & Sons (1998).
- Lizana, K. A., Santander, Victor Francisco Araya., Alencar, F. M. R., Castro, J. F. B., Diaz, J. S. Derivacion de Casos de Uso con Aspectos a partir de Modelos Organizacionales i* In: XII Conferencia Iberoamericana de Ingeniería de Requisitos y Ambientes de Software, p.253 - 258, Medellin: Universidad Eafit (2009).
- Lucena, M. J. N. R., Silva, C. T. L. L., Santos, E. B., Alencar, F. M. R., Castro, Jaelson F.B., Applying Transformation Rules to Improve i* Models. In: 21st International Conference on Software Engineering & Knowledge Engineering (SEKE'2009), p. 43-48.2009, Boston. Proceedings of the 21st International Conference on Software Engineering & Knowledge Engineering (SEKE'2009), (2009).
- Santander, V. F. A., Silva, D. R.: Requirements Engineering Contributions on the Development of Educational Software for the Blind or People with Impaired Vision – An Experience Account. CLEI Electronic Journal 9 (2006).
- Santander, Victor Francisco Araya, Vicente, Andre Abe, Koerich, Fabio, Castro, J. F. B. Modelagem de Requisitos Organizacionais, Não-Funcionais e Funcionais em Software Legado com Ênfase na Técnica i* In: X Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software. p.47 – 60, Isla Margarita. Caracas: Editores F. Losavio, G.H. Travassos, V.Pelechano, I. Diaz, A. Matteo (2007).
- Santander, Victor Francisco Araya., Silva, D. R., Vicente, A. A., Castro, J. F. B., Utilizando a Técnica i* para Modelar a Concepção de Vigotski visando auxiliar o Processo de Desenvolvimento de Software Educacional para Pessoas com Deficiência Visual In: X Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software. p.269 – 282, Isla Margarita, Caracas: F.Losavio, G.H.Travassos, V.Pelechano, I.Diaz, A. Matteo, (2007a).
- Santander, V. F., Castro, J. F. B., *Deriving Use Cases from Organizational Modeling* In: IEEE Joint International Requirements Engineering Conference - RE'02, p. 32-39, Essen, Germany, (2002).
- Teixeira, E.P., Santander, Victor Francisco Araya., Integrando a Teoria da Atividade e a Técnica i* na fase de Requisitos Detalhados. XIII Congresso IberoAmericano em Software Engineering, p.57 – 62. Cuenca, Equador, Universidad del Uzuay, (2010).
- Vicente, André A., Santander, Victor Francisco Araya, Castro, Jaelson B., Freitas da Silva, Ivonei., Reyes Matus, Francisco G., JGOOSE: A Requirements Engineering Tool to Integrate i* Organizational Modeling with Use Cases in UML. *Ingeniare. Revista chilena de ingeniería.* , v.17, p.6 - 20,(2009).
- Yu, E.: *Modelling Strategic Relationships for Processes Reengineering*. Toronto, Canadá: University of Toronto, PhD Thesis (1995).
- Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J., *Social Modeling for Requirements Engineering*. The MIT Press (2011).