

# Uma abordagem para extração de artefatos de software de modelos de Arquitetura Corporativa de TI

Gabriel M. Miranda, Paulo S. dos Santos Jr., Rodrigo F. Calhau, Mateus B. Costa

Instituto Federal do Espírito Santo (Ifes) – Campus Serra  
Rodovia ES-010 – Km 6,5 – Manguinhos, 29.173– Serra/ES – Brasil

gabrielmartinsmiranda@gmail.com, {paulo.junior,  
calhau,mcosta}@ifes.edu.br

**Abstract.** *Enterprise Architecture aims at describing organizational aspects at different abstraction levels and viewpoints. Therefore, Enterprise Architecture models provide a major information source for Information Systems Development. This paper discusses the adoption of Enterprise Architecture models in Software Development processes by providing an approach for use cases and software requirement elicitation based on such models.*

**Resumo.** *Uma Arquitetura Corporativa de TI descreve os aspectos tecnológicos de uma organização sob diferentes níveis de abstração e pontos de vista. Dessa forma, modelos utilizados na descrição dessas arquiteturas constituem uma fonte rica de informação para o Desenvolvimento de Sistemas de Informação. Este artigo discute a adoção de modelos de arquiteturas corporativas de TI em processos de desenvolvimento de Software e apresenta uma abordagem que permite extrair requisitos funcionais e casos de usos desses modelos.*

## 1. Introdução

O desenvolvimento de Sistemas de Informação (SI) é uma atividade complexa [Pádua et al., 2004] [Pressman, 2011]. Um dos motivos dessa complexidade é a dificuldade de entender o domínio do problema, que em muitos casos é referente ao entendimento da organização no qual o SI será implantando. Para analisar e entender uma organização é necessário que o responsável pelo desenvolvimento entenda os domínios de conhecimento (e.g., processos de negócio, tecnologias da informação e infraestrutura) e as relações entre esses domínios de conhecimento [DODAF, 2007].

A necessidade do entendimento do domínio do problema é fator recorrente no processo de desenvolvimento de SI [Pressman, 2011]. Devido a essa necessidade, ao longo do anos, a Engenharia de Requisitos desenvolveu um conjunto de técnicas (e.g., entrevistas e questionários) que proporcionam o entendimento do domínio do problema [Pressman, 2011]. Além do entendimento do domínio do problema, as técnicas citadas também proporcionam a extração de artefatos de software (e.g., requisitos funcionais e casos de usos) que são necessários para a construção do SI [Pressman, 2011].

Apesar das técnicas de levantamento de requisitos auxiliarem no entendimento da organização, essas, em muitos casos, não descrevem, claramente, as relações entre os diferentes domínios organizacionais que envolvem o contexto do SI, como, por exemplo, (i) as dependências entre os processos de negócio e (ii) a relação direta e indireta da tecnologia de informação, infraestrutura e processos organizacionais. Consequentemente, dificultando a extração de artefatos de software de qualidade.

Dessa forma, segundo Pádua et al. (2014) e Lankhorst (2009), o não entendimento desses domínios organizacionais pode impactar diretamente na qualidade dos requisitos de sistema e, conseqüentemente, na qualidade do SI.

De acordo com [Pádua et al., 2004], a utilização de modelos que descrevam os diferentes domínios organizacionais na etapa de elicitação de requisitos é uma prática valiosa. Afinal, através de tais modelos é possível realizar um alinhamento eficaz e menos onerosa entre tecnologia da informação os processos organizacionais [Lankhorst e Van Drunen, 2007][Jonkers et al., 2004] e, assim, produzir requisitos que estejam alinhados com a organização.

Os *frameworks* de modelagem de Arquitetura Corporativa de TI (do inglês, *Enterprise Architecture*) possuem um conjunto de modelos que podem ser úteis para a elicitação de requisitos. Conforme descrito por [Lankhorst, 2009], os modelos desses *frameworks* visam descrever os domínios de conhecimento presente em uma organização. Portanto, tais modelos são uma fonte rica para extração de requisitos.

Em complemento as técnicas de levantamento de requisitos, este artigo apresenta uma proposta de abordagem que possui como objetivo extrair requisitos funcionais e casos de usos dos modelos de Arquitetura Corporativa de TI. A principal característica dessa abordagem é o uso da semântica dos elementos das linguagens de modelagem, presentes nas Arquitetura Corporativa de TI, na extração dos requisitos funcionais. Neste artigo, a abordagem será aplicada sobre o *viewpoint Business Process Viewpoint*, presente no *framework* de modelagem de Arquitetura Corporativa de TI chamado ArchiMate [TOG, 2012].

O artigo está organizado da seguinte forma: seção 2 apresenta os detalhes sobre a escolha do ArchiMate e do *Business Process Viewpoint*; a seção 3 apresenta a abordagem proposta e a prova de conceito da abordagem. A seção 4 apresenta uma breve discussão sobre trabalhos correlatos. Por fim, a seção 5 apresenta as conclusões e trabalhos futuros.

## 2. ArchiMate e Business Process Viewpoint

O ArchiMate é um *framework* de modelagem de Arquitetura Corporativa de TI que promove a integração dos diversos pontos de vista da organização, promovendo a comunicação entre os *stakeholders*, através dos conceitos de serviço, camadas de abstração e *viewpoint* [TOG, 2012].

A característica central da arquitetura do ArchiMate é ser orientada a serviço. Serviço representa parte de uma funcionalidade que alguma entidade (e.g., organização, departamento, sistema) fornece à outra. Essa orientação a serviços permite uma visão em camadas dos modelos arquitetônicos, onde o conceito de serviço é uma das principais ligações entre as diferentes camadas [LANKHORST, 2009].

O ArchiMate possui as seguintes camadas de abstração: Camada de negócio, Camada de aplicação e Camada de tecnologia. A primeira camada de abstração é responsável por representar os produtos e serviços que são concretizados pelos processos organizacionais. A segunda camada de abstração contém modelos que representam as aplicações que apoiam os processos organizacionais. A última camada de abstração possui modelos que descrevem a infraestrutura necessária para suportar as aplicações da Camada de aplicação.

Os *viewpoints* definem abstrações sobre um conjunto de modelos que representam a arquitetura organizacional, cada um destinado a um tipo particular de *stakeholder* e abordando um determinado conceito [Lankhorst 2009]. Os *viewpoints* podem ser utilizadas tanto para visualizar um determinado aspecto isolado, como para relacionar dois ou mais aspectos da organização. O ArchiMate possui quinze *viewpoints* que permitem analisar desde a estrutura organizacional (e.g., papéis e departamentos) até a relação entre processos de negócios e a infraestrutura utilizada para suportá-los.

Dentre os diversos *viewpoints* presentes no ArchiMate, este artigo está interessado no *Business Process Viewpoint* da Camada de Negócio. Isso ocorre devido ao fato que o *Business Process Viewpoint* é utilizada para apresentar a estrutura dos processos de negócio em um nível alto de abstração. Essa *viewpoint* possui elementos que apresentam os diversos conceitos relacionadas aos próprios processos organizacionais com, por exemplo, a relação entre serviços e processos de negócio e a relação entre papéis e processos de negócio.

O *Business Process Viewpoint* utiliza os elementos notacionais da Camada de Negócio. A Figura 1 apresenta um fragmento do metamodelo da Camada de Negócio, contendo os elementos notacionais utilizados pelo *Business Process Viewpoint*. A semântica das metaclasses forma retiradas da especificação da linguagem [TOG, 2012].

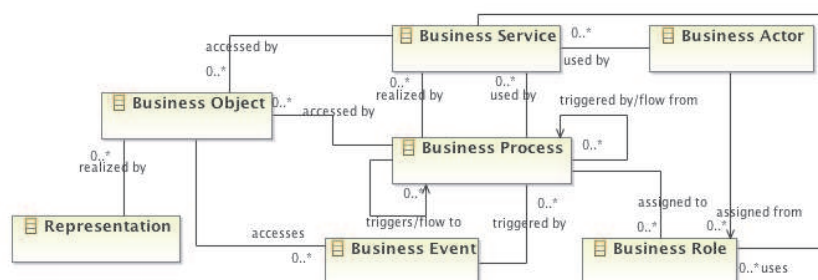
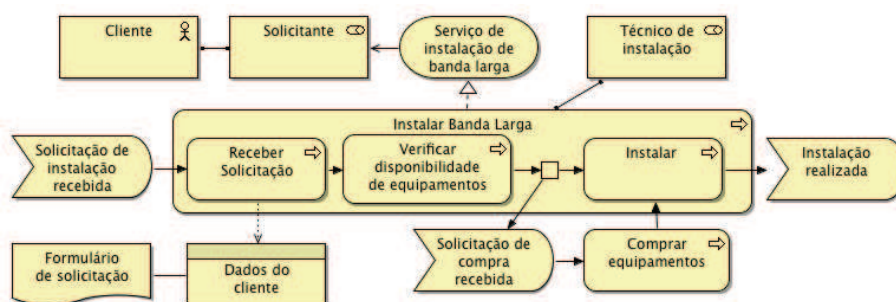


Figura 1. Fragmento do metamodelo da Camada de Negócio [TOG, 2012].

*Business Process* descreve um processo de negócio que é desempenhado por papéis organizacionais, com o objetivo de atingir algum objetivo da organização. Os processos podem possuir outros processos internamente (subprocessos), que são como etapas do processo principal e podem ser executados em sequência ou paralelamente. Exemplos de instâncias de *Business Process* são “Instalar Banda Larga”, “Receber Solicitação”, “Verificar disponibilidade de equipamentos”, “Instalar Banda Larga” e “Comprar Equipamentos” presentes na Figura 2. É importante mencionar que o ArchiMate não possui elementos notacionais para diferenciar processos e subprocessos.

*Business Event* define um evento que influencia o comportamento de um ou processos de negócio. Os processos de negócios podem ser desencadeados ou interrompidos por eventos de negócio [OMG, 2011]. Um evento não possui uma duração definida. Além disso, os eventos podem ser temporais, ou seja, podem representar um dia da semana, por exemplo [TOG, 2012] [OMG, 2011]. Em um processo de negócio, um evento pode ser inicial, intermediário ou final. O evento inicial funciona como um gatilho que inicia um processo de negócio.



**Figura 2. Exemplo de modelo da Business Process View.**

Um exemplo de evento inicial é a “Solicitação de instalação recebida”. O evento intermediário ocorre após o processo ser iniciado, podendo ser o gatilho para outro processo. Um evento intermediário ocorre entre um evento inicial e um evento final em um processo de negócio. Um exemplo de evento intermediário é o *Business Event* “Solicitação de compra recebida”. Um evento final ocorre no fim do processo e, assim, representando o estado final do processo de negócio. Um exemplo de evento final é o *Business Event* “Instalação realizada” [TOG, 2012].

*Business Actor* é uma entidade organizacional capaz de realizar ações na organização. Exemplos típicos desse elemento são as pessoas, grupos e os departamentos. Um ator pode assumir um ou mais papéis em relação a organização. Na Figura 2, o “Cliente” é um exemplo de instância do elemento *Business Actor*.

*Business Role* (papal) descreve um comportamento específico que um *Business Actor* exerce em determinado momento. A partir do momento que um *Business Actor* exerce um papel, um conjunto de responsabilidades e habilidades específicas moldam seu comportamento na organização. Múltiplos atores podem exercer o mesmo papel e o inverso também é válido. Os elementos “Solicitante” e “Técnico de instalação” são exemplos de instâncias de *Business Role*, na Figura 2.

*Business Object* é definido como um elemento passivo que possui relevância na perspectiva do negócio. Esse elemento representa a informação da organização. Na Figura 2, o elemento “Dados do Cliente” é uma instância de *Business Object*.

*Representation* é definida como uma forma perceptível da informação. Em outras palavras, é através do *Representation* que as entidades organizacionais tem acesso aos *Business Object*. Um único objeto de negócio pode possuir várias representações diferentes e uma representação pode conter mais de um *Business Object*. O “Formulário de solicitação”, presente na Figura 2, é uma instância do elemento *Representation*.

As relações *used by* e *uses* são utilizadas para demonstra como os serviços são acessados por processos, funções ou interações e como as interfaces são acessadas por papéis, componentes e colaborações. A seta branca entre “Serviço de Instalação de banda larga” e o “Solicitante” é uma instância dessa relação.

*Accesses* e *accessed by* são utilizadas para descrever que alguma entidade acessa os conceitos do negócio ou objetos de dados presente em um *Business Object*. Esse acesso pode ser realizado das seguintes formas: (i) Leitura: representando uma consulta ao uma informação; (ii) Escrita: representando uma atualização (escrita ou sobrescrita) em um *Business Object* e (iii) Criação: representa uma criação de um *Business Object*.

A seta pontilhada entre o processo “Receber Solicitação” e “Dados do Cliente” é um exemplo dessa relação.

As relações *realized* e *realized by* são utilizadas para descrever um relacionamento entre uma entidade lógica com uma entidade concreta que à realizará. A relação entre “Serviço de Instalação de banda larga” e o processo “Instalar Banda Larga” é um exemplo da relação *realized by*.

As relações *assigned to* e *assigned from* são utilizadas para descrever a associação de unidades de comportamento com elementos ativos (e.g., papéis e componentes) que a executam, ou papéis que os atores cumprem.

As relações *flow to* e *flow from* são utilizadas para descrever um fluxo de informações ou valores entre processos, funções, interações e eventos. As relações *triggers* e *triggered by* são utilizadas para descreve uma relação temporal ou causal entre processos, funções, interações e eventos. As duas primeiras relações são representas pelas setas entre os *Business Process* e as duas últimas são representas pelas as setas entre *Business Process* e *Business Event*, na Figura 2.

Por fim, além dos elementos supracitados, a camada de negócio também possui os *Junction*. A junção é utilizada para controlar o fluxo de execução de um processo. Essa pode ser do tipo *And Junction* que representa que os caminhos do fluxo de execução são paralelos; e do tipo *Or Junction* que representa que um ou mais caminhos podem ser executados no fluxo. O pequeno quadrado preto entre “Verificar disponibilidade de equipamento”, “Instalar” e “Solicitação de compra recebida”, presente na Figura 2, é um exemplo de *Or Junction*.

### 3. Abordagem Proposta

Segundo (PÁDUA et. al., 2004), a compreensão completa de um ambiente corporativo favorece o desenvolvimento de software. Tendo como base esse princípio, essa seção apresenta uma abordagem que visa extrair requisitos funcionais e casos de uso a partir do *Business Process Viewpoint* do ArchiMate. A abordagem proposta é composta por dois conjuntos de transformações. O primeiro conjunto é destinado a extração de casos de usos, enquanto o segundo conjunto é comprometido com a extração de requisitos funcionais. O *Business Process Viewpoint* apresentado na Figura 2, será utilizado como exemplo de uso da abordagem.

#### 3.1. Extração de casos de uso

O primeiro conjunto de transformações é composto pelas seguintes regras de mapeamento: (i) serviços (*Business Service*) em casos de uso e (ii) ator (*Business Actor*) e/ou papel (*Business Role*) em atores de casos de uso.

Basicamente, a primeira regra transforma um serviço em um caso de uso (UC). Portanto, ao aplicar essa regra na *viewpoint* da Figura 2, temos o seguinte caso de uso: (UC1) “Instalar Banda Larga”. Vale ressaltar, que um serviço pode conter outros serviços internos, assim, como um caso de uso pode conter outros casos de uso.

A segunda regra leva em consideração a relação *used by* entre *Business Role* e *Business Actor* e *Business Service*. Nesse caso, os atores e papéis serão transformados em atores do casos de uso. Esse mapeamento possui a regra de precedência: todo *Business Role* é transformado em ator de caso de uso. Caso não existe um *Business*

*Role*, o *Business Actor* é transformado em ator de caso de uso. A Figura 3, apresenta o resultado das duas regras de casos de uso aplicadas sobre o *Business Process Viewpoint* da Figura 2.

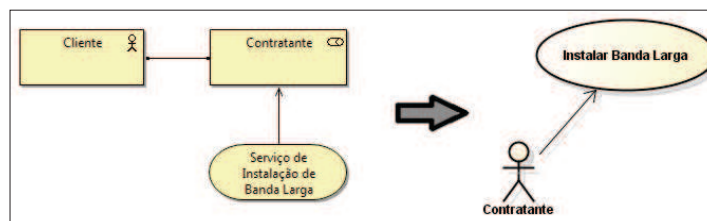


Figura 3. Resultado das duas regras de casos de uso.

### 3.2. Extração de requisitos funcionais

O segundo conjunto de transformação é composto pelas seguintes regras de mapeamento: (i) processos de negócio em requisitos de funcionais; (ii) junções em requisitos funcionais (iii) objetos de negócio em requisitos funcionais e (iv) eventos em requisitos funcionais.

O primeiro mapeamento visa transformar os processos de negócios em requisitos funcionais (RF). Para atingir esse objetivo cada *Business Process* é transformado em um requisito de funcionais (RF). Portanto, ao aplicar o primeiro mapeamento nos *Business Process* presentes no modelo da Figura 2, pode-se extrair os seguintes requisitos funcionais: (RF1) “O sistema deve controlar as solicitações de instalação”, (RF2) “O sistema deve controlar o estoque de equipamentos”, (RF3) “O sistema deve controlar a compra de novos equipamentos” e (RF4) “O sistema deve armazenar informações sobre a instalação”.

O segundo mapeamento visa transformar as junções (*And Junction* e *Or Junction*) em requisitos de funcionais. Isso é feito analisando cada caminho de execução do processo de negócio e transformando esses em requisitos de negócio. Ao aplicar o segundo mapeamento no processo presente na Figura 2, ou seja, ao analisar cada caminho de execução após o processo de “Verificar Disponibilidade de Equipamentos”, pode-se os seguintes requisitos funcionais: (RF5) “Novos equipamentos só devem ser requisitados se não houver os equipamentos necessários para a instalação” e (RF6) “Somente é possível realizar uma instalação se existirem os equipamentos necessários”.

O terceiro mapeamento visa transformar os Objetos de negócio (*Business Objects*) e suas relações de *accessed by/accesses* em requisitos funcionais. Afinal, essas relações podem ser mapeadas como operações do tipo CRUD (*Create, Read, Update e Delete*). Na Figura 2, existe uma relação de *accessed by* do tipo leitura entre “Receber solicitação” (*Business Process*) e “Dados do cliente” (*Business Object*) e uma relação de *realized by* entre “Dados do cliente” e o “Formulário de solicitação” (*Representation*). Analisando esses elementos do modelo e suas relações é possível notar que o processo de negócio necessita das informações contidas na *Business Object*, que serão obtidas através do *Representation*. Aplicando o terceiro mapeamento sobre esse fragmento do modelo, pode-se extrair o seguinte requisito funcional: (RF7) “O Sistema deve consultar os dados do cliente no formulário de solicitação”.

Por fim, o quarto, e último, mapeamento tem como objetivo derivar requisitos funcionais a partir de evento (*Business Event*). Portanto, ao aplicar o quarto

mapeamento sob o modelo, presente na Figura 2, os seguintes requisitos funcionais são extraídos: (RF8) “O sistema deve receber a solicitação de instalação de banda larga”, (RF9) “O sistema deve enviar a solicitação para compra de novos equipamentos quando necessário” e (RF10) “O sistema deve notificar a conclusão da instalação”. Como pode ser visto, ao aplicar o segundo conjunto de regras foi possível extrair 10 requisitos funcionais de alto nível de um exemplo fictício.

### 3.3. Prova de Conceitos

Para realizar a prova de conceitos da abordagem proposta, foi realizado um experimento que envolvia extrair requisitos funcionais e casos de uso de um projeto real chamado SINCAP [UKA 2013].

O experimento envolveu duas equipes de analistas de sistemas. A primeira equipe utilizou técnicas tradicionais de extração de artefatos de software. A segunda equipe ficou com a abordagem descrita neste artigo. Como resultado do experimento, a primeira equipe derivou 21 (vinte e um) requisitos funcionais e 12 (doze) casos de usos e a segunda equipe 27 (vinte e sete) requisitos funcionais e 17 (dezesete) casos de usos. É importante comentar que os alguns requisitos funcionais gerados por ambas as equipes possuem grau níveis diferentes de granularidade.

A abordagem proposta gerou mais requisitos funcionais de alto nível do que a abordagens tradicionais utilizadas, auxiliando, assim, no entendimento global do problema. Por outro lado, as abordagens tradicionais geraram requisitos com um grau de detalhamento melhor, permitindo se aprofundar no problema em questão.

Portanto, pode-se concluir que a abordagem proposta pode ser combinada com as abordagens tradicionais para melhorar o entendimento do domínio do problema.

## 4. Discussão e Trabalhos Correlatos

É importante comentar que o abordagem proposta permite derivar artefatos de software, de alto nível de abstração, que estão alinhados com a organização. No entanto, torna-se necessário combinar a abordagem proposta com as abordagens tradicionais para uma melhora na qualidade dos artefatos durante o processo de desenvolvimento do SI.

Em relação a trabalhos correlatos, um estudo que se assemelha ao apresentado neste artigo é o conduzido por Oliveira et al (2013). Nesse estudo os autores apresentam uma proposta de derivação de requisitos (funcionais e não funcionais) a partir de modelos de processo de negócio feitos em BPMN [OMG, 2011], utilizando a técnica REMO [Oliveira et al 2013]. A principal vantagem dessa abordagem é a possibilidade de extrair requisitos de menor granularidade do que a abordagem proposta neste artigo. Por outro lado, a abordagem proposta neste artigo, permite um melhor entendimento da organização e utiliza a semântica dos elementos e de suas relações para a extração dos requisitos funcionais e casos de uso.

Por fim, o trabalho proposto por Cardoso (2009) também é semelhante ao proposto neste artigo. Nesse trabalho, os requisitos são derivados dos seguintes modelos do ARIS Method: *Event-driven Process Chain (EPC)*, *Value-Added Chain* e *Function Allocation Diagram (FAD)*. No entanto, diferente da abordagem proposta nesse artigo, Cardoso (2009) não é explicitado a forma que o requisitos são extraídos de tais modelos.

## 5. Conclusão e Trabalhos Futuros

Pádua et al. 2004, propõem a utilização de modelos organizacionais para a derivação de artefatos de software que estejam alinhados com a realizada das organização. Tendo em vista essa problemática, este artigo propôs uma abordagem complementar as técnicas de levantamento de requisitos tradicionais. A abordagem proposta permite analisar as informações presentes no *Business Process Viewpoint* do ArchiMate e, assim, extrair requisitos funcionais e casos de uso que serão úteis para o desenvolvimento do sistema.

Uma característica importante da abordagem proposta é a utilização da semântica dos elementos presente no modelos do ArchiMate. Como apresentado nas seções 3 e 4 o estudo aprofundado da semântica dos elementos de modelagem possibilitou um melhor proveito das informações contidas nos modelos e, conseqüentemente, um melhor entendimento do domínio do problema.

Em relação aos trabalhos futuros, os autores pretendem: (i) utilizar outros *viewpoints* do ArchiMate como fonte de extração de artefatos de software, (ii) realizar uma análise ontológica no ArchiMate com o objetivo de melhorar o entendimento dos elementos e corrigir possíveis erros de interpretação e (iii) combinar outros modelos (e.g., BPMN e modelos de objetivos) com os modelos do ArchiMate com o objetivo de melhorar o entendimento da organização e, conseqüentemente, melhorar a qualidade dos artefatos de software extraídos.

## Referências

- Cardoso, E. C. S., Almeida, J. P., Guizzardi, G. (2009) "Requirements Engineering Based on Business Process Models: A Case Study". EDOC Conference
- TOG (2012) "ArchiMate 2.0 Specification, Technical Standard", The Open Group, <http://www.opengroup.org/archimate/>
- OMG (2011) "Business Process Model and Notation (BPMN) 2.0 Specification", Object Management Group, <http://www.omg.org/spec/BPMN/2.0>
- Pádua, et al (2004) "Modelagem Organizacional: captura dos requisitos organizacionais no desenvolvimento", *Gestão & Produção*, São Paulo, v.11, n.2, p. 197-209.
- LANKHORST, M et al (2009) "Enterprise Architecture at Work", 2<sup>a</sup> ed. Germany, Berlin: Springer, 2009. 352 p.
- PRESSMAN, R. S. (2011) "Engenharia de Software: uma abordagem profissional". 7<sup>a</sup> ed. 778 p.
- DODAF (2007) "DoD Architecture Framework Version 1.5", V. II
- Uka, C. T. M.; Erlacher, R.; Santos, T. Z. R.; Komati, K. S.; Santos Jr., P. S.; Calhau, R. F. (2013). "Web-based Notification System for an Organ Transplant Process" No: International Workshop on ADVANCES in ICT Infrastructures and Services.
- Marcos Oliveira, Sérgio Roberto Costa Vieira, Davi Viana, Sabrina Marczak, Tayana Conte (2013) "Elicitação de Requisitos a partir de Modelos de Processos de Negócio e Modelos Organizacionais: Uma pesquisa para definição de técnicas baseadas em heurísticas". Requirement Engineering.