



**UNIVERSIDADE SALVADOR – UNIFACS
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E
COMPUTAÇÃO
MESTRADO EM SISTEMAS E COMPUTAÇÃO**

ANTÔNIO OLIVEIRA FILHO

**UMA TÉCNICA DE RASTREABILIDADE PARA ANÁLISE DE
IMPACTO DE MUDANÇAS EM INTERESSES DE NEGÓCIO**

Salvador
2010

ANTÔNIO OLIVEIRA FILHO

**UMA TÉCNICA DE RASTREABILIDADE PARA ANÁLISE DE
IMPACTO DE MUDANÇAS EM INTERESSES DE NEGÓCIO**

Dissertação apresentada ao curso de Mestrado Acadêmico em Sistemas e Computação, Universidade Salvador – UNIFACS, como requisito parcial para obtenção do grau de Mestre.

Orientadora: Prof^a. Dr^a. Christina von Flach G. Chavez

Salvador
2010

TERMO DE APROVAÇÃO
ANTÔNIO OLIVEIRA FILHO

UMA TÉCNICA DE RASTREABILIDADE PARA ANÁLISE DE IMPACTO DE
MUDANÇAS EM INTERESSES DE NEGÓCIO

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre
em Sistemas e Computação, Universidade Salvador - UNIFACS, pela seguinte
banca examinadora:

Christina von Flach G. Chavez - Orientadora _____
Doutora em Informática, Pontifícia Universidade Católica do Rio de Janeiro, PUC-Rio
Universidade Federal da Bahia - UFBA

Julio Cesar Sampaio do Prado Leite _____
Doutor em Ciência da Computação, University of California, Irvine (UMCP)
Pontifícia Universidade Católica do Rio de Janeiro, PUC-Rio

Cássio Vinicius Serafim Prazeres _____
Doutor em Ciências da Computação e Matemática Computacional, Universidade de
São Paulo (USP)
Universidade Salvador - UNIFACS

Salvador, 07 de Janeiro de 2010.

Ficha Catalográfica elaborada pelo Sistema de Bibliotecas da Universidade Salvador
- UNIFACS

Oliveira Filho, Antônio

Uma técnica de rastreabilidade para análise de impacto de mudanças em interesses de negócio. / Antônio Oliveira Filho. – Salvador, 2010.

95 f. : il.

Dissertação apresentada ao Curso de Mestrado em Sistemas e Computação, Universidade Salvador – UNIFACS, como requisito parcial para a obtenção do grau de Mestre.

Orientadora Prof^a. Dr^a. Christina von Flach G. Chavez.

1. Engenharia de Software. I. Chavez, Christina von Flach G., orient. II. Universidade Salvador – UNIFACS. III. Título.

CDD: 005.1

AGRADECIMENTOS

Agradeço a todos aqueles que me ajudaram direta ou indiretamente na realização deste trabalho.

Ao Professor Doutor Celso Saibel que tolerou minhas insistentes investidas por diversas ocasiões. Obrigado Celso!

Ao Professor Doutor Manoel Mendonça simplesmente pela presença! Sua presença é inspiradora!

Ao Professor Doutor Julio Leite que indiretamente me ajudou. Julio, até o presente momento, não houve uma iniciativa minha que não esbarrasse em um trabalho seu!

À Professora Doutora Karin Breitman que lá no início viu potencial na “viagem” que eu acabava de lhe apresentar.

Ao meu amigo Ênio Virgílio que me ajudou com a matemática presente neste trabalho, além do apoio dispensado até o presente momento e durante o resto de nossas vidas. Agora somos colegas de faculdade!

Ao meu amigo Guilherme Lima que me ajudou lá o início, quando a idéia que se materializou neste trabalho ainda era muito confusa para mim.

Aos meus dois amores, minha esposa Eunice e meu filhote Joaquim. Obrigado por vocês existirem na minha vida!

À minha mãe, Carmelita, e ao meu pai, Antônio, pela eterna torcida e apoio incondicional!

Aos professores, colegas e funcionários do curso de Mestrado em Sistemas e Computação da UNIFACS, com os quais convivi nos últimos anos, pelo companheirismo e amizade.

Por fim, quero manifestar, de forma especial, a minha gratidão a toda a minha família e aos amigos que compartilharam comigo momentos de tristeza e de alegria.

Nota: Christina, minha orientadora, não apareceu nos agradecimentos por ela ser “hors-concours”!

RESUMO

Análise de impacto é a identificação das potenciais conseqüências de uma mudança, ou estimativa do que precisa ser modificado para realizar uma mudança, incluindo estimativas de custos e cronograma. Neste trabalho, fazemos a distinção entre dois tipos de preocupações ou interesses relacionados com a análise de impacto: (1) interesses específicos do negócio, relacionados com os interessados em verificar se outras regras de negócios são afetados pela mudança e também precisa ser modificadas e (2) interesses específicos do software, relacionados com as partes interessadas nos artefatos de software afetadas que precisam ser modificados. Várias técnicas de rastreabilidade têm sido estudadas e nenhuma delas apóia a análise de impactos sobre interesses específicos do negócio com valores resultantes razoáveis de precisão e revocação para os impactos descobertos. Nosso trabalho de investigação tem por objetivo apoiar interesses específicos do negócio durante a análise de impacto, ao propor e avaliar uma nova técnica de rastreabilidade chamada FIR, que utiliza um novo modelo de rastreabilidade definido sobre regras de negócio, com resultados de precisão e de revocação esperados de 100%. Uma ferramenta chamada de *FIR-Diagram* é apresentada como prova de conceito para ilustrar a viabilidade da técnica proposta. O Brasil tem vivenciado nos últimos anos um crescimento bastante significativo em relação ao investimento na melhoria e qualidade dos produtos de software.

Palavras-chave: Regras de Negócio. Rastreabilidade de Requisitos. Interesses de Negócio. Análise de Impacto.

ABSTRACT

Impact analysis is the identification of the potential consequences of a change, or estimating what needs to be modified to accomplish a change, including related costs and schedule estimates. In this work, we distinguish between two kinds of concerns related to impact analysis: (1) business-specific concerns, those related to stakeholders interested in checking if other business rules are impacted by the change and also need to be modified; and (2) software-specific concerns, those related to stakeholders interested in the impacted software artifacts that need to be modified. Several traceability techniques have been studied and none of them supported impact analysis that dealt with business-specific concerns with reasonable values of precision and recall for the discovered impacts. Our research work aims to support business-specific concerns during impact analysis, by proposing and evaluating a new traceability technique called FIR, that resorts on a new traceability model defined over business rules, with expected precision and recall values of 100%. A tool called the FIR-diagram is presented as proof of concept to illustrate the feasibility of the proposed technique.

Keywords: Business Rules. Requirement Traceability. Business Concern. Impact Analysis.

LISTA DE FIGURAS

Figura 1 – Exemplo de Emissão de Ingressos	16
Figura 2 - Regras de Negócio	22
Figura 3 - Regras de Negócio	23
Figura 4 - Halle x Cysneiros	24
Figura 5 - Atributos de qualidade de requisitos de negócio	29
Figura 6 – Rastreabilidade para trás e para frente	31
Figura 7 - Processo simplificado de negociação	34
Figura 8 - Consequências de Pr e Rr	35
Figura 9 - Consequências de Pa e Ra	36
Figura 10 - Consolidação das Questões Q1, Q2 e Q3	39
Figura 11 - Elos entre requisitos, feature e elementos de design	43
Figura 12 – Visão Geral do Modelo FIR	47
Figura 13 - Modelo FIR	49
Figura 14 – Regras de Halle x Oliveira	49
Figura 15 – O exemplo de Emissão de Ingressos	50
Figura 16 - O Conjunto Θ_n para o exemplo de Emissão de Ingressos	51
Figura 17 – O Conjunto Γ_n para o exemplo Emissão de Ingresso	52
Figura 18 – O Conjunto Γ_n para o exemplo de Emissão de Ingresso	53
Figura 19 – O Conjunto Ψ_n para o exemplo de Emissão de Ingresso	54
Figura 20 – O Conjunto Ψ_n para o exemplo de Emissão de Ingressos	55
Figura 21 – O Conjunto Φ_n para o exemplo de Emissão de Ingressos	56
Figura 22 – Uma Visão Gráfica do FIR para o exemplo de Emissão de Ingressos	57
Figura 23 – Requisitos de Negócio do exemplo de Emissão de Ingressos	58
Figura 24 – Relação entre o conjunto Θ_{ns} e o conjunto Θ_n	59
Figura 25 – Relação entre o conjunto Γ_{ns} e o conjunto Γ_n	60
Figura 26 – Relação entre o conjunto Ψ_{ns} e o conjunto Ψ_n	61
Figura 27 – Visão Gráfica de Interesses de Negócio	62
Figura 28 – Os Conjuntos $F-s-n$, $B-s-n$ e $I-s-n$ para o exemplo apresentado no Capítulo 1	63
Figura 29 - O conjunto $\Theta-s-n$ para o exemplo apresentado no Capítulo 1	64
Figura 30 - O conjunto $\Gamma-s-n$ para o exemplo apresentado no Capítulo 1	64
Figura 31 - O conjunto $\Psi-s-n$ para o exemplo apresentado no Capítulo 1	65
Figura 32 - O conjunto Γ_{ns} para o exemplo apresentado no Capítulo 1	66
Figura 33 - O conjunto $\Gamma-s-n$ para o exemplo apresentado no Capítulo 1	67
Figura 34 – O conjunto Ψ_{ns} para o exemplo apresentado no Capítulo 1	68
Figura 35 - O conjunto $\Psi-s-n$ para o exemplo apresentado no Capítulo 1	68
Figura 36 - O conjunto $\Phi-s-n$ para o exemplo apresentado no Capítulo 1	69
Figura 37 - O conjunto Φ_{ns} para o exemplo apresentado no Capítulo 1	70
Figura 38 - Visão gráfica do Exemplo apresentado no Capítulo 1	71
Figura 39 - O grafo FIR ⁹ para o exemplo apresentado no Capítulo 1	73
Figura 40 – Detalhamento do grafo FIR ⁹ para o exemplo apresentado no Capítulo 1	74
Figura 41 - Um Grafo FIR ⁹ mínimo para o exemplo apresentado no Capítulo 1	75
Figura 42 – Detalhamento de um grafo FIR ⁹ mínimo para o exemplo apresentado no Capítulo 1	75
Figura 43 - Um grafo que não é um grafo FIR ⁹ para o exemplo do Capítulo 1	75

Figura 44 – Um grafo detalhado que não é um grafo FIR ⁹ para o exemplo do Capítulo 1	76
Figura 45 - Rastreamento do exemplo do Capítulo 1 pelas técnicas estudadas	77
Figura 46 - Rastreamento do exemplo do Capítulo 1 através da Técnica FIR	79
Figura 47 - Satisfação dos Requisitos	81
Figura 48 – Visão geral do GEF	82
Figura 49 - Processo do GMF para Integrar o EMF ao GEF	83
Figura 50 - Relação entre os modelos no processo GMF	84
Figura 51 – Diagrama ecore	85
Figura 52 – Elementos de FIR-Diagram	87
Figura 53 – Paleta de Componentes do FIR-Diagram	88
Figura 54 – Relacionamentos entre as classes do modelo e os correspondentes elementos gráficos	88
Figura 55 – Relacionamentos entre as classes do modelo e os elos no gráfico	88
Figura 56 – Expressão em OCL para o atributo label da classe Funcionalidade	89
Figura 57 - Restrição implementada em Java	89
Figura 58 – Diagrama FIR do Exemplo apresentado no Capítulo 1	90
Figura 59 – Revisitando o exemplo do Capítulo 1	91
Figura 60 – Descoberta de Impactos através do FIR-Diagram	92

LISTA DE ABREVIATURAS E SIGLAS

BRG	Business Rules Group
EBT	Event-Based Approach
EI	Emissão de Ingressos
EMF	Eclipse Modeling Framework
FB	Feature Model-Based Approach
FIR	Funcionalidade-Informação-Regra
FIR ^g	Grafo FIR
GEF	Graphical Editing Framework
GMF	Graphical Modeling Framework
GR	Gerência de Requisitos
IRA	Information Retrieval Approach
JET	Java Emitter Templates
MOF	Meta Object Facility
MVC	Model-View-Controller
OCL	Object Constraint Language
OMG	Object Management Group
SB	Scenario-Based Approach
VBRT	Value-Based Requirement Traceability Approach
XMI	Metadata Interchange
XML	Xtensible Modeling Language

SUMÁRIO

1 INTRODUÇÃO	14
1.1 PROBLEMA	16
1.2 PROPOSTA	18
1.3 ORGANIZAÇÃO	19
2 REGRAS DE NEGÓCIO	20
2.1 REGRAS DE NEGÓCIO POR HALLE	20
2.2 REGRAS DE NEGÓCIO POR CYSNEIROS	22
2.3 CORRESPONDÊNCIA ENTRE AS DEFINIÇÕES	24
2.4 ASPECTOS DA QUALIDADE DAS REGRAS DE NEGÓCIO	26
3 REQUISITOS DE SOFTWARE	28
3.1 ATRIBUTOS DE QUALIDADE DE REQUISITOS DE NEGÓCIO	28
3.2 RASTREABILIDADE DE REQUISITOS	29
3.2.1 Elos de Rastreabilidade	30
3.2.2 Usos da Rastreabilidade	32
3.3 CONTROLE DE MUDANÇAS	33
3.4 ANÁLISE DE IMPACTO	34
3.5 TÉCNICAS DE RASTREABILIDADE DE REQUISITOS	36
4 ESTUDO DE TÉCNICAS DE RASTREABILIDADE DE REQUISITOS	37
4.1 METODOLOGIA	37
4.1.1 Seleção das Técnicas de Rastreabilidade	37
4.1.2 Definição de Hipóteses, Questões e Métricas	38
4.2 TÉCNICAS DE RASTREABILIDADE	39
4.2.1 Information Retrieval Approach (IRA)	40
4.2.1.1 Avaliação	40
4.2.2 Event-Based Approach (EBT)	41
4.2.2.1 Avaliação	41
4.2.3 Value-Based Requirement Traceability	41
4.2.3.1 Avaliação	42
4.2.4 Feature Model-Base Approach (FB)	42
4.2.4.1 Avaliação	43
4.2.5 Scenario-Based Approach (SB)	43
4.2.5.1 Avaliação	44
4.3 DISCUSSÃO	44
5 A TÉCNICA FIR	47

5.1 VISÃO GERAL DE TÉCNICA FIR	47
5.2 IDENTIFICAÇÃO E CONSTRUÇÃO DE INTERESSES DE NEGÓCIO	48
5.2.1 Obtenção de Subconjuntos os Tipos de Regras de Negócio	49
5.2.2 Relações entre os Tipos de Regras de Negócio	51
5.2.3 Agrupamentos entre Funcionalidade, Regra e Informação	53
5.2.4 A Relação Interesse de Negócio	55
5.3 INTERPRETAÇÃO DE INTERESSES DE NEGÓCIO PARA REQUISITOS DE NEGÓCIO	57
5.3.1 Relações entre os tipos de regras de negócio em S	59
5.3.2 Agrupamento Entre Funcionalidade, Regra e Informação em S	60
5.3.3 A Relação Interesse de Negócio em S	61
5.4 DELIMITAÇÃO DAS FRONTEIRAS	62
5.4.1 Negócio fora de S	62
5.4.2 Relações fora de S	63
5.4.3 Agrupamento entre Funcionalidade, Informação e Regra fora de S	64
5.4.4 A Relação Interesse de Negócio fora de S	65
5.4.5 Relacionamentos na Fronteira	65
5.4.6 Agrupamento entre Funcionalidade, Informação e Regra na fronteira de S	67
5.4.7 Interesses de Negócio na fronteira de S	69
6 ANÁLISE DE IMPACTO COM FIR	72
6.1 O GRAFO FIR ⁹ COMPLETO	74
6.2 DESCOBERTA DE IMPACTOS EM UM GRAFO FIR ⁹	76
6.3 DISCUSSÃO	77
7 O PLUGIN FIR-DIAGRAM	80
7.1 REQUISITOS PARA FIR-DIAGRAM	80
7.2 TECNOLOGIA UTILIZADA NA CONSTRUÇÃO DO PLUGIN	81
7.2.1 GEF	81
7.2.2 EMF	82
7.2.3 Integração entre EMF e GEF	82
7.3 CONSTRUÇÃO DO PLUGIN FIR-DIAGRAM	84
7.3.1 Construção do <i>fir.ecore</i>	85
7.3.2 Construção do <i>fir.gmfgraph</i>	87
7.3.3 Construção do <i>fir.gmftool</i>	87
7.3.4 Construção do <i>fir.gmfmap</i>	88
7.3.5 Implementação de Restrições no Modelo	89
7.4 O EXEMPLO DE EMISSÃO DE INGRESSOS	89
8 CONCLUSÃO	93
8.1 CONTRIBUIÇÕES	93

8.2 TRABALHOS FUTUROS	94
REFERÊNCIAS	95

94
95

1 INTRODUÇÃO

A Gerência de Requisitos (GR) é o processo geral de compreender e controlar mudanças e seus impactos nos requisitos de software (SOMMERVILLE, 2006), buscando manter a consistência entre eles e os demais artefatos de software. A descoberta dos impactos relacionados a uma mudança insere-se no processo de análise de impacto de mudanças, ou simplesmente, análise de impacto. Análise de impacto é definida como a identificação das consequências em potencial de uma mudança ou estimativa do que precisa ser modificado para realizar uma mudança (ARNOLD, 1996), possibilitando, por exemplo, que custos e prazos possam ser estimados a partir dos impactos, antes que a mudança seja efetivamente implementada. Para realizar a análise de impacto, são utilizadas informações que relacionam requisitos e artefatos de software, idealmente consultadas a cada nova solicitação de mudanças (KASSE, 2004).

Rastreabilidade de requisitos é definida como a habilidade de descrever e seguir a vida de um requisito em ambas as direções, para trás e para frente da especificação de requisitos (GOTEL; FINKELSTEIN, 1994). Em geral, a rastreabilidade é mantida por elos que relacionam requisitos a outros requisitos (elos de dependência), a artefatos que os atendem (elos de satisfação) e às suas fontes (SAYÃO; LEITE, 2006). Nesse contexto, definimos quais interesses relacionados aos artefatos são aqueles cujos interessados se preocupam em conhecer os impactos nos artefatos utilizados ao longo do ciclo de vida do software, com suporte da rastreabilidade.

Além disso, quando o software presta serviços para uma organização, a evolução pode ocorrer a partir de solicitações de mudança em regras de negócio. Regras de negócio são declarações que definem ou restringem algum aspecto do negócio (BUSINESS RULES GROUP, 2010). Em geral, regras de negócio estão mapeadas em requisitos de negócio (LEITE; LEONARDI, 1998). Por outro lado, mudanças nas regras de negócio podem causar impacto em outras regras de negócio relacionadas, quer elas estejam implementadas no software ou não. Nesse contexto, definimos que interesses relacionados ao negócio são aqueles cujos interessados se preocupam em conhecer os impactos no negócio e regras de negócio associadas, independentemente de haver ou não software que os implemente.

Diversas técnicas de rastreabilidade de requisitos têm sido desenvolvidas e estudadas nos últimos anos (ROCHIMAH; KADIR; ABDULLAH, 2007; RAJA; KAMRAN, 2008). Uma análise preliminar sobre resultados de estudos realizados sobre algumas destas técnicas de rastreabilidade revelou que elas dão suporte a interesses relacionados aos artefatos, ainda que com algumas limitações, por exemplo, necessidade de intervenção manual para realizar ajustes nos elos, dificuldade em gerar elos com semântica clara e precisa (CLELAND-HUANG, 2006), uma vez que, os elos de dependência, por exemplo, não são capazes de informar o real motivo de sua criação. A análise preliminar destas técnicas de rastreabilidade (ROCHIMAH; KADIR; ABDULLAH, 2007; RAJA; KAMRAN, 2008) também revelou que elas não dão suporte adequado a interesses de negócio, em especial, se não houver software que implemente as regras de negócio impactadas.

Para avaliar técnicas de rastreabilidade no contexto da atividade de descoberta de impactos, duas métricas são frequentemente utilizadas: precisão e revocação (recall). Cada uma dessas métricas podem ser consideradas sobre elos de dependência (E_r) e sobre elos de satisfação (E_s), como a seguir: (1) A precisão da descoberta de impactos em outros requisitos (P_r) a partir de elos E_r e a (2) precisão da descoberta de impactos nos artefatos (P_a) a partir de elos E_s ; (3) a revocação da descoberta de impactos em outros requisitos (R_r) a partir de elos E_r e a (4) revocação da descoberta de impactos nos artefatos (R_a) a partir de elos E_s . Precisão P_r e P_a iguais a 100% combinadas com revocações R_r e R_a iguais a 100% podem compor estimativas de custos e prazos 100% ajustados à realidade atual do projeto. Neste caso, se a solicitação de mudança ocorrer para um requisito de negócio, então os interessados pelos impactos no negócio poderão conhecer todos os requisitos de negócio impactados possíveis, bem como, os interessados pelos impactos nos artefatos poderão conhecer todos os artefatos impactados possíveis.

A Figura 1 é utilizada para ilustrar algumas limitações das técnicas de rastreabilidade estudadas. Dificuldades para representar elos de dependência entre regras de um negócio de venda de ingressos para jogos de futebol. Recentemente, uma mudança na lei, obrigou que no ingresso constasse o CPF do torcedor além das informações corriqueiras, tais como, nome e data do evento, para conferência

no acesso ao estádio. Cada torcedor deve ser cadastrado através de um serviço web já disponível.

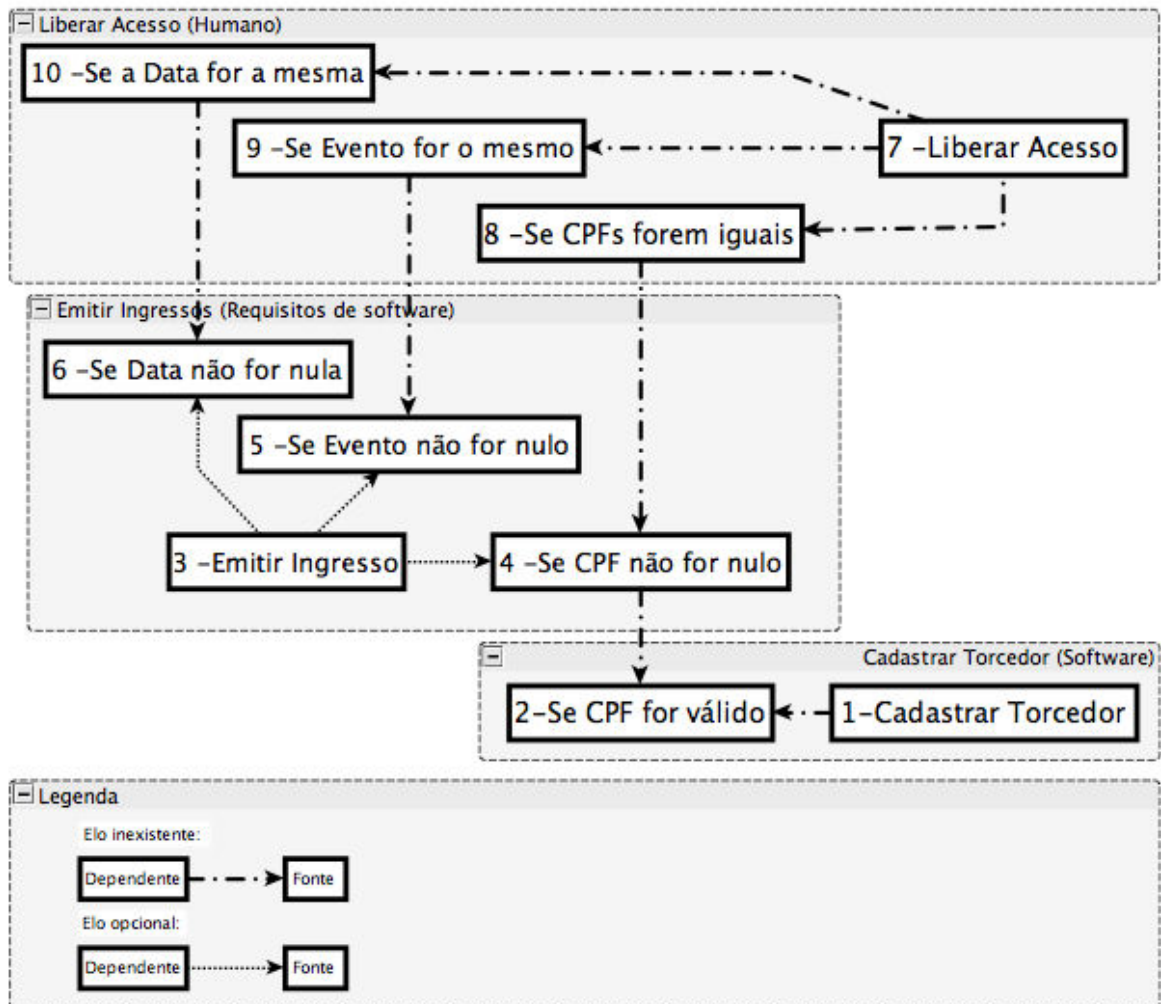


Figura 1 – Exemplo de Emissão de Ingressos
Nota: Autoria Própria

Esse exemplo será utilizado na Seção seguinte para ilustrar alguns problemas das técnicas de rastreabilidade estudadas, além de ser utilizado ao longo dos demais Capítulos sempre que for necessário.

1.1 PROBLEMA

Tendo como ponto de partida os dois tipos de interesses descritos anteriormente, a saber, (i) interesses relacionados ao negócio (cujos interessados se preocupam em conhecer os impactos no negócio, independentemente de haver ou não software) e (ii) interesses relacionados aos artefatos (cujos interessados se

preocupam em conhecer os impactos nos artefatos utilizados ao longo do ciclo de vida do software) e baseando-se em duas situações retiradas do exemplo já exposto, em que: (1) uma mudança no requisito de negócio 4 provoca impactos no requisito de negócio 3 e na regra de negócio 8; e (2) uma mudança no requisito de negócio 2 provoca impactos no requisito de negócio 1 e nas regras de negócio 4 e 8. Alguns problemas foram identificados nas técnicas de rastreabilidade estudadas, como descritos a seguir.

a) **Elos E_r opcionais:** As técnicas estudadas não dispõem de elos de dependência obrigatórios. Na situação 1 do exemplo, um elo de dependência entre os requisitos de negócio 3 e 4 pode não estar presente na rastreabilidade, pois as técnicas não obrigam sua presença. Devido a essa opcionalidade, não se pode garantir que o impacto em 3 seja descoberto;

b) **Elos E_r sem semântica clara:** As técnicas estudadas não dispõem de elos de dependência com semântica clara, de modo que, uma vez estabelecida a dependência entre dois requisitos de negócio através de um elo, não se pode garantir que posteriormente, apenas observando o elo, se descubra o que motivou a criação de tal elo. Na situação 1 do exemplo, devido a ausência de semântica clara, uma vez que a dependência entre os requisitos de negócio 3 e 4 foi estabelecida, não se pode garantir posteriormente o que originou essa dependência. Esse fato poderia levar a futuras inconsistências da rastreabilidade, caso o elo não fizesse mais sentido existir devido à mudança, mas mesmo assim permanecesse definido;

c) **Elos inexistentes:** Na situação 1 e 2 do exemplo, o impacto no requisito de negócio 6 e os impactos nos requisitos de negócio 4 e 6, são desconhecidos, uma vez que os elos entre os requisitos de negócio 2 e 4 e os requisitos de negócio 4 e 8 não são considerados pelas técnicas estudadas. Isso se explica pelas regras de negócio 2 e 8 não pertencerem aos requisitos do software, alvo da rastreabilidade;

d) **Precisão P_r desconhecida:** Os experimentos descritos nos trabalhos das técnicas estudadas não reportam valores de precisão P_r . Essa ausência de valores, talvez seja decorrência dos três problemas anteriormente citados. Esse desconhecimento impossibilita prever o esforço para se estimar custos e prazos de mudanças no negócio.

e) **Revocação R_r desconhecida:** Semelhante à precisão P_r , os estudos não dispõem de valores de revocação R_r (referente a requisitos). Essa ausência de valores, assim como o ocorrido com P_r , talvez deva-se aos três primeiros problemas anteriormente citados. Esse desconhecimento impossibilita prever o quanto nas estimativas de custos e prazos de mudanças no negócio incluem-se requisitos de negócio corretos.

f) **Separação de interesses:** Todos os problemas acima citados contribuem para uma não separação entre interesses no negócio e interesses nos artefatos. Tanto na situação 1, quanto na situação 2, interessados no negócio não conseguem, apenas através da rastreabilidade, conhecer todos os impactos nas demais regras de negócio de forma independente, ou seja, utilizando apenas elos de dependência. Porém, mesmo os elos de satisfação (entre requisitos e artefatos) não apresentam valores de precisão e revocação satisfatórios (CLELAND-HUANG, 2006).

1.2 PROPOSTA

Este trabalho apresenta uma nova técnica de rastreabilidade chamada de FIR (Funcionalidade-Informação-Regra) para apoiar a análise de impacto de mudanças com suporte aos interesses de negócio. A técnica está fundamentada sobre um modelo de rastreabilidade chamado de Modelo FIR, originalmente proposto por Oliveira Filho, Chavez e Mendonça Neto (2008) e refinado neste trabalho.

Nossa proposta compreende uma técnica manual de rastreabilidade que define elos de dependência especializados com a mesma semântica observada nas relações entre regras de negócio, entre regras de negócio e requisitos de negócio e vice-versa. A técnica prioriza de modo incondicional os interessados no negócio e, portanto, os requisitos de negócio. Através da rastreabilidade provida pela técnica, espera-se descobrir impactos de mudanças com 100% de precisão e revocação para mudanças que compreendam situações tais como as 1 e 2 do exemplo apresentado.

Considerando a Figura 1, a técnica de rastreabilidade FIR lida com a separação entre interesses no negócio e interesses nos artefatos. A técnica proposta é capaz de representar elos de dependência com semântica clara e precisa, tendo como base o modelo FIR. Tal modelo exhibe relacionamentos entre

regras de negócio, e entre regras de negócio e requisitos de negócio e vice-versa, de modo a permitir que impactos de mudanças sejam descobertos em cenários como os das duas situações anteriormente descritas. Isto possibilita que verificações de completude da especificação de requisitos de negócio possam ser feitas com o auxílio da rastreabilidade suportada. O modelo FIR é definido como um grafo denominado FIR⁹ e a atividade de descoberta de impactos é apresentada em termos de operações sobre tal grafo.

1.3 ORGANIZAÇÃO

Os Capítulos 2 e 3 apresentam a fundamentação deste trabalho. No Capítulo 2 são apresentados definições e modelos de regras de negócio que, juntos, servem de base para direcionar nosso trabalho e, no Capítulo 3, abordamos requisitos de software obtidos a partir do negócio e conceitos sobre rastreabilidade e análise de impacto.

O Capítulo 4 apresenta uma avaliação sistemática realizada sobre quinze técnicas de rastreabilidade referenciadas em uma revisão sistemática planejada e executada por Raja e Kamran (2008). Os resultados obtidos serviram de motivação para propor mudanças nas técnicas estudadas e identificar características desejáveis para guiar o projeto de uma nova técnica de rastreabilidade que apóie a análise de impacto de maneira mais satisfatória.

O Capítulo 5 apresenta o modelo FIR e a técnica FIR, com descrição dos passos e modelos gerados para o exemplo de Emissão de Ingressos da organização Ingressos & Cia.

O Capítulo 6 apresenta o modelo FIR bem-formado como um grafo denominado FIR⁹ e a atividade de descoberta de impactos é apresentada em termos de operações sobre tal grafo.

O Capítulo 7 apresenta a ferramenta FIR-Diagram, desenvolvida como prova de conceito, ou seja, para dar suporte à criação e validação de modelos FIR⁹, bem como seu uso para apoiar a atividade de análise de impacto sobre uma mudança solicitada para uma regra de negócio do exemplo de Emissão de Ingressos.

O Capítulo 8 apresenta as conclusões e os trabalhos futuros.

2 REGRAS DE NEGÓCIO

Neste Capítulo, apresentamos definições e modelos de regras de negócio que, juntos, servem de base para direcionar nosso trabalho. O Business Rules Group (2010) define regra de negócio como sendo “uma declaração que define ou restringe algum aspecto do negócio”. Essa definição, embora sintética, apresenta o cerne do conceito. Nas Seções seguintes, destacamos alguns tipos das regras de negócio de acordo com os trabalhos de Von Halle (2001) (Seção 2.1) e de Cysneiros e Leite (1999) (Seção 2.2). Em seguida, apresentamos alguns aspectos de qualidade para avaliação de regras de negócio, individualmente e em conjunto (Seção 2.3). Ao longo deste Capítulo é utilizado o exemplo que foi apresentado no Capítulo 1.

2.1 REGRAS DE NEGÓCIO POR HALLE

Von Halle (2001) define regras de negócio em função de subtipos (Figura 2). A seguir, apresentamos algumas definições que aparecem na Figura 2.

- a) **Termo (term):** É um nome ou frase nominal que está de acordo com alguma definição. Um Termo pode ser um conceito, tal como *Torcedor*; uma propriedade, tal como *Torcedor-CPF*; um valor, tal como *Domínio*; um conjunto de valores, tal como *Domingo*, *Sábado*. Von Halle (2001) observa que no domínio da modelagem de dados, por exemplo, termos se tornarão entidades, atributos, domínios ou constantes. No exemplo, *Torcedor* provavelmente se tornará uma entidade; *Torcedor-CPF* provavelmente se tornará um atributo CPF de *Torcedor*; *Domingo* provavelmente se tornará uma constante; *Sábado*, *Domingo* provavelmente se tornará um domínio;
- b) **Fato (fact):** É uma declaração que conecta termos através de preposições e frases verbais, de acordo com observações relevantes para o negócio, por exemplo, *Torcedor pode ter Ingresso*. Aqui, Von Halle (2001) novamente observa que no domínio da modelagem de dados, os fatos se tornarão relacionamentos entre entidades ou entre entidades e atributos. No exemplo acima, *Torcedor pode ter Ingresso* provavelmente se tornará um relacionamento entre a entidade *Torcedor* e a entidade *Ingresso*.

c) **Regra (Rule):** Especificamente, uma regra pode fazer uma das seguintes opções no contexto de eventos de negócio: (1) Apresentar informações sobre o evento de negócio; (2) Restringir informações criadas por um evento de negócio; (3) Iniciar uma ação fora dos limites do sistema alvo ou evento de negócio; (4) Criar uma informação nova. Como apresentado na Figura 2, uma regra pode ser classificada como:

- **Restrição (constraint):** É uma declaração que expressa uma circunstância incondicional que deve ser verdadeira ou falsa para que um evento do negócio seja completado de forma integra. Por exemplo: (1) *O documento de CPF do portador do ingresso deve ser igual ao CPF que consta no ingresso para liberar acesso;* (2) *A data do evento que consta no ingresso emitido deve ser igual à data do atual para liberar acesso;* (3) *O nome do evento que consta no ingresso deve ser igual ao nome do evento para liberar acesso.*
- **Diretriz (guideline):** É uma declaração que expressa uma circunstância que deveria ser verdadeira ou falsa. Uma diretriz não força uma solução, mas apenas deixa para o humano a tomada de decisão. Por exemplo: *A quantidade de ingressos emitidos para um torcedor por jogo não deveria ultrapassar 3.*
- **Capacita Ação (action enabler):** É uma declaração que expressa uma condição que, quando verdadeira, capacita a realização de outra ação. Por exemplo, *se o torcedor não está cadastrado, então cadastre-o.* Von Halle (2001) acrescenta que este tipo de regra de negócio em alguns softwares comerciais serve para criar um fluxo de trabalho orientado a eventos, em que cada ação é um evento. Halle observa também que o tipo Restrição é o oposto do tipo Capacita Ação, pois enquanto o primeiro pode abortar o evento em curso, o segundo pode iniciar um novo evento.
- **Computação (computation):** É uma declaração que expressa um algoritmo para calcular termos a partir dos operadores matemáticos. Por exemplo, *o índice de participação do torcedor é computado pelo quociente do total de acessos liberados pelo total de ingressos vendidos.* Von Halle (2001) observa que a execução de uma computação produz um nova informação.

- **Inferência (inference):** É uma declaração que expressa uma condição que, quando verdadeira, estabelece um novo fato. Por exemplo, se o *CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado*. Von Halle (2001) observa que a execução de uma inferência produz uma nova informação e que na parte da declaração que corresponde ao novo fato pode haver uma computação.

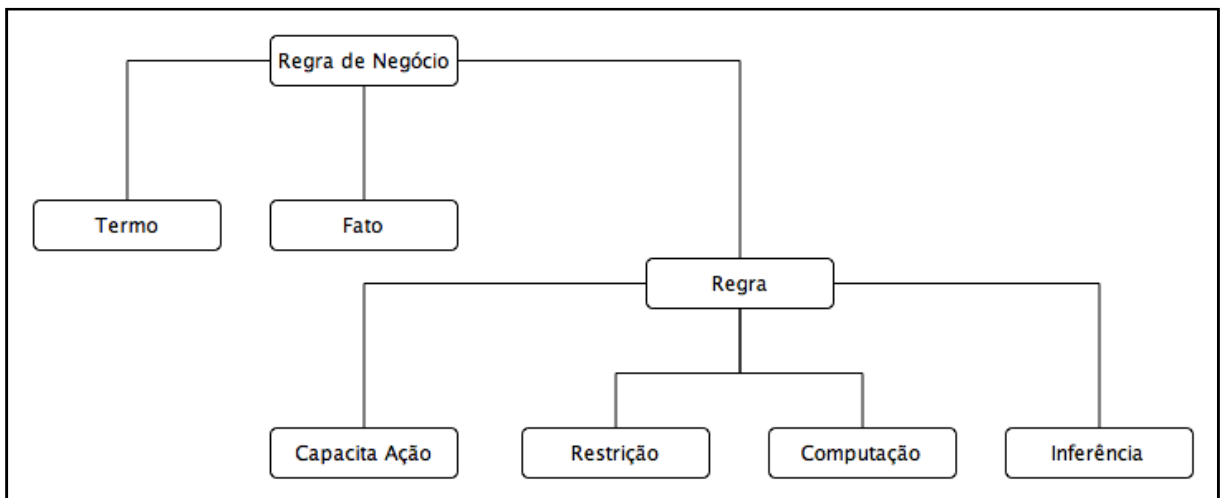


Figura 2 - Regras de Negócio
Nota: Extraído de Von Halle (2001).

2.2 REGRAS DE NEGÓCIO POR CYSNEIROS

Leite e Leonardi (1998) apresentaram uma definição para regras de negócio compilada de Rosca, Feblowitz e Wild (1997), em que regras de negócio “são declarações sobre a forma como as organizações fazem negócios. Elas refletem as políticas do negócio. Organizações têm políticas a fim de satisfazer os objetivos do negócio e os clientes, fazer bom uso dos recursos, e estar em conformidade com leis ou convenções gerais do negócio”. Para os autores, regras de negócio são uma categoria de requisitos de software que deveriam compor a baseline (LEITE; OLIVEIRA, 1995) do projeto, ou seja uma base comum contendo regras de negócio em linguagem natural. Nesse mesmo trabalho, foi apresentada uma taxonomia para regras de negócio procurando relacioná-las a requisitos de software. Cysneiros e Leite (1999) tomaram como base esta taxonomia e propôs um modelo de regra de negócio. A Figura 3 apresenta os elementos deste modelo.

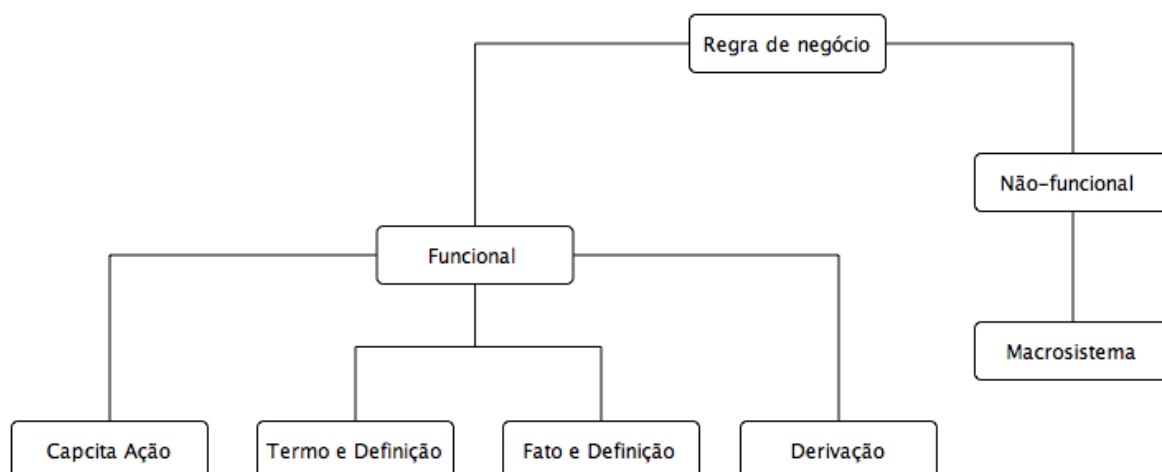


Figura 3 - Regras de Negócio
Nota: Extraído de Cysneiros (1999)

a) Regras funcionais: São políticas gerais relativas às funcionalidades organizacionais. Dividem-se em dois grupos que são distintos pelos padrões das declarações;

- **Grupo 1:** É formado por declarações que obedecem o padrão Frase Não-Verbal + Frase Verbal + [Frase Não-Verbal]. Este grupo divide-se nos seguintes tipos.
 - **Termo/Definição:** Denota algum tipo de definição. Algumas palavras comumente encontradas são: *é definido como, pode ser visto como, é denotado como, é conhecido como, é alguém que, é algo que, é um/uma*. Por exemplo: (1) *um torcedor é alguém que compra ingresso;* (2) *Um CPF é algo que é exigido para Liberar Acesso ao estádio;*
 - **Derivação:** Expressa algum tipo de operação matemática. Algumas palavras comumente encontradas são: *é calculado, é computado, é igual a*. Por exemplo, *o índice de participação do torcedor é computado pelo quociente do total de acessos liberados pelo total de ingressos vendidos;*
 - **Fato:** Representa a associação entre um ou mais termos. Por exemplo, *um ingresso pode ser emitido para um torcedor.*
- **Grupo 2:** É formado por declarações que obedecem aos seguintes padrões: (1) Se + Frase + Então/Implica + Frase; (2) Quando + Frase + Então/Implica + Frase; (3) Porque + Frase; (4) Frase Não-Verbal + Frase Verbal + Porque + Frase.

- **Capacita Ação:** Expressa uma condição e uma ação; quando a condição é verdadeira, capacita a ação. Por exemplo, se *torcedor não estiver cadastrado, então o cadastre*.
- b) Não-funcionais:** Podem ser vistas como regras de negócio que se preocupam com restrições e comportamentos. Existe um subtipo descrito a seguir.
- **Macrosistema:** descreve as políticas necessárias para restringir o comportamento da organização. Pode seguir o padrão [Propriedade] + Frase Não-Verbal + Relação + [Propriedade] + Frase Não-Verbal, onde Propriedade é uma frase não-verbal e Relação é uma frase verbal. Por exemplo, *o CPF do Cliente deve ser informado durante o cadastramento*.

2.3 CORRESPONDÊNCIA ENTRE AS DEFINIÇÕES

Os modelos de Von Halle (2001) e de Cysneiros e Leite (1999) possuem alguns elementos em comum. A Figura 4 apresenta uma correspondência entre os modelos. Em seguida, o exemplo apresentado no início deste Capítulo 1 é usado para ilustrar os conceitos.

Halle	Cysneiros
Termo	Termo
Fato	Fato
Restrição	Macrosistema
Diretriz	–
Computação	Derivação
Inferência	–
Capacita Ação	Capacita Ação

Figura 4 - Halle x Cysneiros
Nota: Autoria Própria

a) Termos

- *Torcedor*;
- *Ingresso*;
- *Torcedor-CPF*;

- *Ingresso-CPF;*
- *Ingresso-Nome-do-Evento;*
- *Ingresso-Data-do-Evento;*

b) Fatos

- *Torcedor pode ter Ingresso;*
- *Torcedor tem CPF;*
- *Ingresso contém CPF;*
- *Ingresso contém Nome-do-Evento;*
- *Ingresso contém Data-do-Evento;*

c) Inferência (Exclusivo de Von Halle (2001))

- *Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado;*
- *Se Nome do Evento informado for válido, então o Ingresso-Nome-do-Evento é igual ao Nome do Evento informado;*
- *Se a Data do Evento informada for válida, então o Ingresso-Data-do-Evento é igual à Data do Evento informada;*
- *Se Torcedor-CPF for válido, então o Ingresso-Torcedor-CPF é igual à Torcedor-CPF informado;*

d) Restrição

- *O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado;*
- *O Ingresso-CPF deve ser igual ao CPF do Torcedor para que o acesso seja liberado;*
- *O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado;*

e) Capacita Ação

- *Se o Torcedor ainda não estiver cadastrado, então o cadastre;*
- *Se o Ingresso ainda não estiver emitido, então o emita;*

- *Se o Torcedor ainda não estiver acessado o estádio, então libere o acesso;*

2.4 ASPECTOS DA QUALIDADE DAS REGRAS DE NEGÓCIO

Von Halle (2001) descreve sete critérios de qualidade usados para avaliar regras de negócio.

- Relevante/Justificada:** Cada regra deve ter sua relevância justificada pelo público alvo envolvido no negócio ao qual a regra está inserida;
- Atômica:** Cada regra deve ser representada de modo que não possa mais ser decomposta e ainda assim puder guiar o comportamento de um ator humano ou não-humano. Cada regra deve representar uma unidade de mudança;
- Declarativa:** Cada regra deve prescrever uma decisão ou de computação, em vez de ditar um procedimento para a realização e execução de decisão ou computação;
- Inteligível/Precisa:** O público-alvo da regra deve entendê-la de tal forma que a regra seja previsível e passível de uso repetitivo.
- Completa:** Cada regra deve possuir todas as propriedades intelectuais necessárias para seu uso. As decisões e computações não devem ser parciais.
- Confiável:** Cada regra deve originar-se de uma fonte autorizada para decidir que a regra é um desejo do negócio. Por saber quem é a fonte, economiza-se tempo na investigação de mudanças nas regras propostas;
- Autêntica:** Mesmo que uma regra seja reproduzida de diversas representações e copiada de uma representação para outra, essa regra deve manter-se fiel a representação declarativa original. Por exemplo, uma regra pode ser copiada de uma representação em linguagem natural para uma representação que siga algum padrão, para código fonte, etc.

Outros três critérios de qualidade dizem respeito ao conjunto de regras de negócio.

- Completo/Previsível:** Todas as regras de negócio necessárias para manter o negócio íntegro precisam estar presentes;

- b) **Único/Não-redundante/Mínimo:** Deve existir um subconjunto controlado de redundâncias;
- c) **Consistente:** Um conjunto de regras não deve conter regras que se contradizem.

3 REQUISITOS DE SOFTWARE

Neste capítulo, abordamos requisitos de software obtidos a partir do negócio, alguns atributos de qualidade e conceitos sobre rastreabilidade e análise de impacto.

Leite e Leonardi (1998) denominaram requisitos de software obtidos a partir do negócio de requisitos de negócio, bem como estabeleceram uma relação entre eles e regras de negócio ao definir que “regras de negócio se tornam requisitos, ou seja, podem ser implementados em um sistema de software como uma forma de requisitos de software desse sistema”. Uma vez que uma regra de negócio passe pelo crivo de atributos de qualidade descritos no Capítulo 2, ela estará apta para se tornar um requisito de software.

3.1 ATRIBUTOS DE QUALIDADE DE REQUISITOS DE NEGÓCIO

A seguir, encontra-se um resumo dos atributos de qualidade de requisitos apresentados por Wieggers (1999) e Young (2001).

- a) **Necessário:** O sistema é capaz de atingir seus objetivos sem este requisito?
Caso afirmativo este é um requisito desnecessário.
- b) **Verificável:** É possível verificar que este requisito está sendo atendido pelo sistema?
- c) **Priorizável:** O requisito é passível de ser priorizado frente aos outros requisitos?
- d) **Livre de Ambiguidades:** O requisito possui mais de uma interpretação possível?
- e) **Atingível:** O requisito pode ser atendido pelo sistema que está sendo desenvolvido?
- f) **Alocável:** O requisito pode ser alocado a um elemento ou componente do sistema?
- g) **Conciso:** O requisito está descrito de forma simples e concisa?
- h) **Correto:** O requisito contém todas as informações necessárias que permitam sua implementação?
- i) **Livre de Implementação:** O requisito descreve o QUE deve ser feito sem descrever COMO?

- j) Rastreável: A origem do requisito é conhecida? O requisito pode ser referenciado e localizado no sistema?

A Figura 5 apresenta uma comparação entre os atributos de qualidade de requisitos de software e os atributos de qualidade de regras de negócio.

Atributos de Requisitos	Atributos de Regras	Comentário
Necessário	Relevante/Justificada	Regras relevantes e justificadas se tornam requisitos necessários.
Verificável		É exclusivo de requisitos de software
Priorizável	Relevante/Justificada	
Livre de Ambiguidades	Inteligível/Precisa	
Atingível	Relevante/Justificada	
Alocável		É exclusivo de requisitos de software.
Concisão	Atômica	
Corretude	Autêntica/Confiável/Completa	
Livre de Implementação	Declarativa	
Rastreável		Este critério aparece em Halle como um princípio, logo, deve ser atendido.

Figura 5 - Atributos de qualidade de requisitos de negócio

Nota: Autoria Própria

Neste trabalho, nós estamos particularmente interessados nos atributos rastreável e priorizável das regras de negócio e, conseqüentemente, dos requisitos de software. O primeiro atributo tem ligação direta com nossa proposta e é abordado amplamente a partir da Seção 3.2 e do Capítulo 4. O segundo atributo também é abordado no Capítulo 4, pois algumas técnicas de rastreabilidade priorizam requisitos para reduzir o volume de elos da rastreabilidade. Ao longo deste capítulo também tratamos de mudanças em requisitos e da gestão destas mudanças.

3.2 RASTREABILIDADE DE REQUISITOS

Segundo Gotel e Finkelstein (1994), rastreabilidade de requisitos é “a habilidade de descrever e seguir a vida de um requisito em ambas direções, para trás e para frente da especificação de requisitos. Para trás, explica a origem do requisito, por exemplo, que decisões estão associadas a sua criação e se possível quem as tomou. Para frente, explica onde o requisito é tratado ao longo das etapas

do seu desenvolvimento”. Os artefatos para frente são ditos que satisfazem o requisitos (RAMESH; JARKE, 2001). Raja e Kamran (2008) apresentaram os resultados de uma quasi-revisão sistemática cujo assunto foi rastreabilidade e esta definição foi a mais citada pelos trabalhos selecionados.

Para Wieringa (1995), os documentos criados e mantidos durante as fases iniciais do desenvolvimento e através do ciclo de vida do software são rastreáveis se “partes de documentos distintos puderem ser relacionadas entre si, quando mudanças em uma parte provocarem mudanças na outra”. As duas definições são complementares. A primeira definição trata do todo, enquanto que a definição de Wieringa trata das partes dos documentos, trazendo um forte apelo por precisão, que é uma das métricas em que estamos interessados.

Para Sommerville (2006), um requisito é rastreável se “é possível descobrir quem sugeriu o requisito (fonte), a razão da existência do requisito (rationale), quais os outros requisitos que estão relacionados a ele (dependência entre requisitos) e como o requisito se relaciona com outras informações, tais como, desenho do sistema, implementação e documentação do usuário”. Esta definição aborda a necessidade de se considerar os elos de rastreabilidade.

3.2.1 Elos de Rastreabilidade

As associações entre artefatos ou entre partes destes artefatos são mantidas através de elos de rastreabilidade com diferentes propósitos associados. Para Gotel e Finkelstein (1994), a especificação de requisitos deve ser utilizada como base para os elos de rastreabilidade. A idéia resumida está presente na Figura 6 que apresenta os elos antes (Pre-RS Traceability) da especificação de requisitos (Requirements Specification) e após (Post-RS Traceability) a especificação.

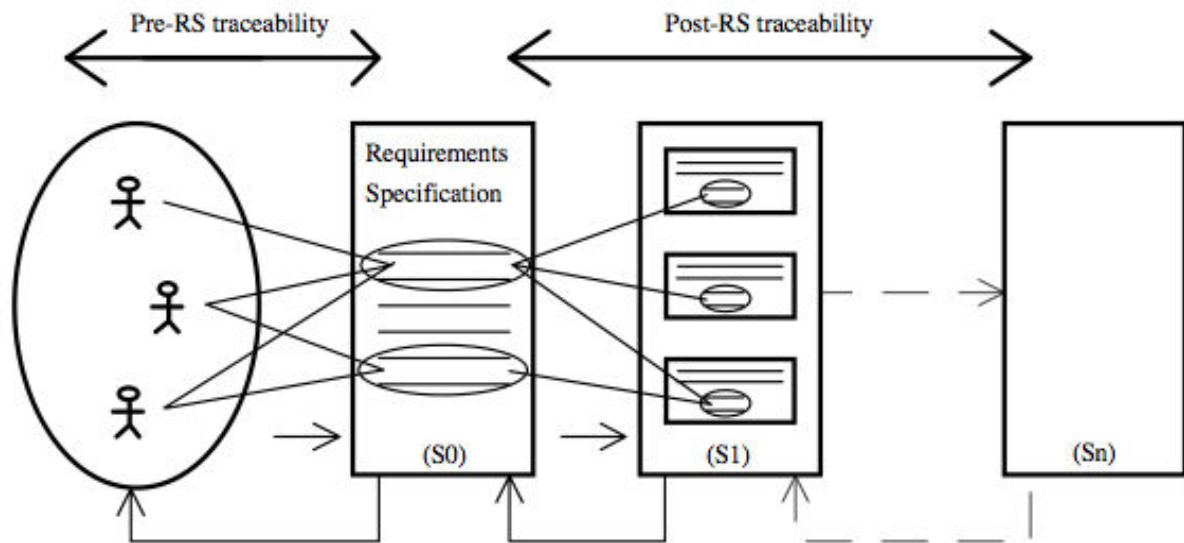


Figura 6 – Rastreabilidade para trás e para frente
 Nota: Extraído de Gotel e Finkelstein (1994)

Tipos distintos de elos têm sido propostos e apresentados, ora em meta-modelos (RAMESH; JARKE, 2001; TORANZO, 2002), ora em definições próprias da rastreabilidade (SOMMERVILLE, 2006).

- a) **Rationale:** Este tipo de elo é explicado por Sayão e Leite (2006), como representante dos motivos pelos quais cada requisito está especificado. Esta definição vale tanto para a criação do requisito quanto para a sua evolução;
- b) **Evolução:** Uma baseline de requisitos é uma estrutura paralela ao desenvolvimento e que tem como objetivo modelar e acompanhar a evolução dos requisitos externos ao software (LEITE; OLIVEIRA, 1995). Na medida em que requisitos evoluem, este tipo de elo mantém e relaciona o estado anterior ao estado após a modificação;
- c) **Fonte:** Este tipo de elo indica a fonte de extração do requisito, como exemplo, atas de reunião, legislação, normas, etc. No modelo de Gotel (1996), as fontes documentadas são chamadas de estruturas de contribuição;
- d) **Responsável:** Este tipo de elo indica as pessoas que se responsabilizam pela criação e modificação dos requisitos.
- e) **Dependência:** Elos de dependência capturam a interdependência entre requisitos;
- f) **Satisfação:** Elos de satisfação indicam em que artefatos os requisitos estão implementados ou de alguma maneira referenciados;

- g) **IR**: Este tipo de elo é um refinamento dos elos de dependência aplicados para requisitos funcionais e foi proposto pela primeira vez por Oliveira Filho, Chavez e Mendonça Neto (2008). O elo IR trata das relações de dependência mais finas que existem entre as regras e as informações. Na seção dedicada ao nosso modelo, nós exploraremos mais este elo.

3.2.2 Usos da Rastreabilidade

A rastreabilidade é um importante meio a partir do qual a Gerência de Requisitos pode se apoiar em um conjunto de situações. Sayão e Leite (2006) destacam algumas destas situações:

- a) **Resolução de requisitos em conflito**: durante o processo de elicitação é comum que diferentes representantes do cliente ou usuário interajam com a equipe de desenvolvimento, trazendo suas necessidades em relação ao sistema a ser desenvolvido. Essas necessidades irão gerar requisitos, e a análise de requisitos pode apontar necessidades ou requisitos conflitantes. A rastreabilidade auxilia a identificação das origens dos requisitos em conflito, para busca de uma solução para o problema detectado.
- b) **Verificação**: na análise de cobertura de requisitos nos testes, a avaliação da rastreabilidade entre requisitos e casos de testes permite identificar requisitos (ou funcionalidades) para os quais não foram previstos casos de testes, indicando à equipe de testes a necessidade de complementar a massa de testes.
- c) **Validação**: a etapa final de validação do sistema criado junto ao conjunto de clientes e usuários se beneficia da rastreabilidade, permitindo mostrar se a implementação atende ao conjunto de requisitos acordados entre clientes e desenvolvedores.
- d) **Análise de impacto na evolução do sistema**: a existência de elos de rastreabilidade entre requisitos e componentes facilita a identificação dos componentes afetados por mudanças em um requisito ou mesmo por inclusão de novos requisitos, sem que haja necessidade de consultas a diferentes artefatos (por exemplo, projeto da arquitetura, modelo de classes, código de componentes). Caso o requisito sendo alterado esteja ligado a outros

requisitos (dependência entre requisitos), estes também deverão ser avaliados.

Em nosso trabalho, estamos particularmente interessados no processo de análise de impacto de mudança.

3.3 CONTROLE DE MUDANÇAS

Software evolui para se manter útil (LEHMAN, 1996). A evolução é um processo que pode se dar por inúmeras razões. Para um software que presta serviços (SOMMERVILLE, 2006) para uma organização, uma das razões pode ser a ocorrência de mudanças no negócio para o qual o software é útil. Neste caso, solicitações de mudanças devem passar por um processo como o abordado na Figura 7 que, em geral, inclui as seguintes atividades:

- a) **Solicitar Mudança:** Esta atividade define a mudança que vai ser solicitada, podendo ser a adição de um novo requisito, uma exclusão ou alteração de algum requisito já existente.
- b) **Analisar Impacto:** Cada solicitação de mudança define o conjunto de requisitos e artefatos impactados. Idealmente, cada requisito impactado é analisado para decidir se o requisito necessitará ou não de mudança. Em caso positivo, o requisito deve ter definido o conjunto de requisitos e artefatos impactados e assim por diante, até não haver mais requisitos impactados. Alguns outros detalhes estão na Seção 3.4;
- c) **Negociar Impactos:** Os custos e prazos resultantes da análise de impacto são negociados com os interessados;
- d) **Atualizar Especificação:** Toda mudança aceita deve ser repercutida na especificação. Para ser mais completo seria necessário avaliar a qualidade da especificação para tentar descobrir se a mudança feriu alguns dos parâmetros de qualidade.
- e) **Armazenar Solicitação Rejeitada:** Esta atividade deve formar uma base com as solicitações rejeitadas com a intenção de servir de suporte a consultas futuras.

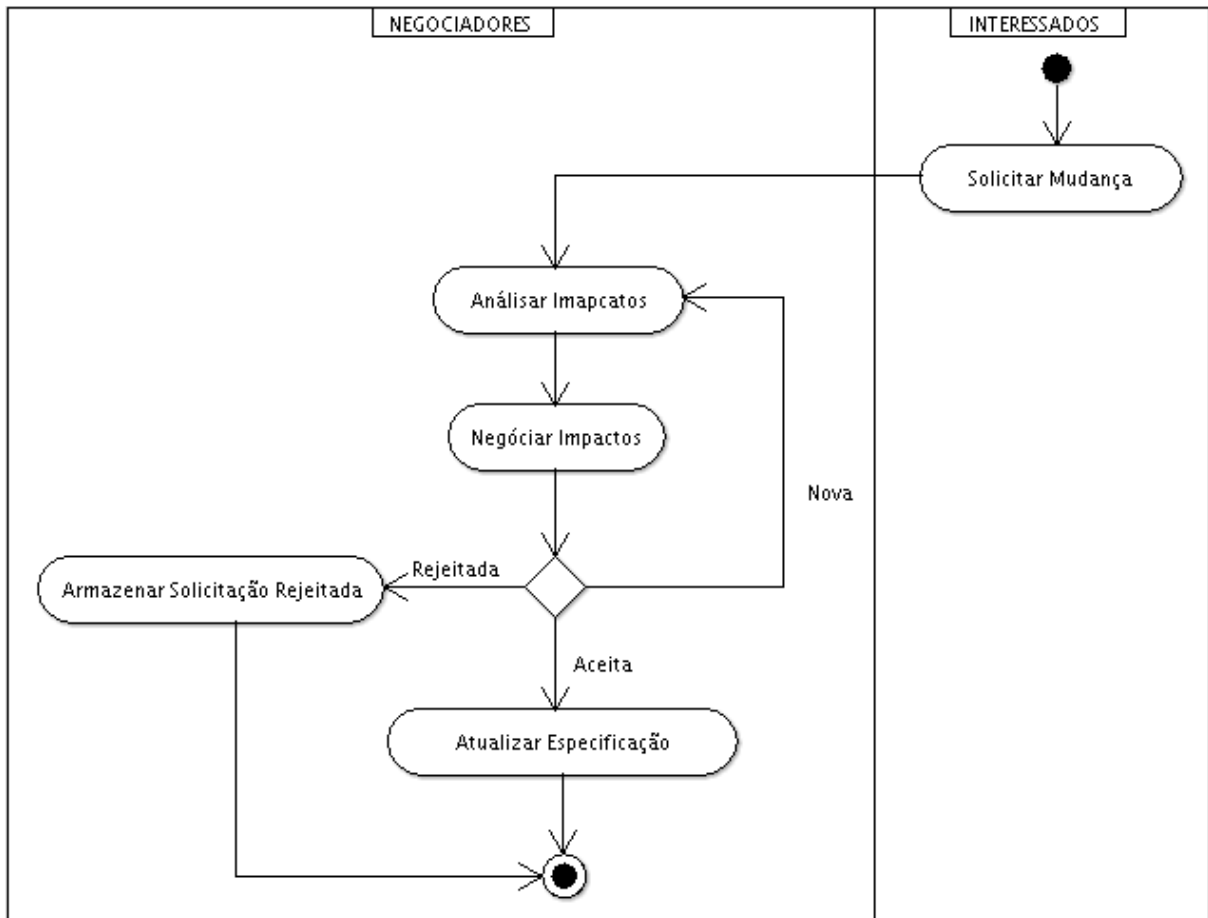


Figura 7 - Processo simplificado de negociação

Nota: Autoria Própria

3.4 ANÁLISE DE IMPACTO

A análise de impacto “permite identificar consequências e prover estimativas em termos de custos e prazos de uma solicitação de mudança, antes que a mesma seja implementada” (ARNOLD, 1996). Para o cálculo das estimativas, devem ser considerados os impactos descobertos através da rastreabilidade de requisitos, idealmente é consultada a cada nova solicitação de mudanças (KASSE, 2004). Deste modo, a qualidade das estimativas depende da qualidade das descobertas de impactos. Para avaliar a qualidade das descobertas, duas importantes métricas são precisão e revocação (recall). Assim, para uma solicitação de mudança de requisito, deseja-se saber:

A precisão (P_r) e a revocação (R_r) da descoberta de impactos nos demais requisitos associados:

- a) $P_r = \frac{\text{requisitos corretos descobertos}}{\text{requisitos descobertos}}$. Se $P_r = 1$ significa que todos os requisitos impactados descobertos são corretos, mas se $P_r < 1$, então existe ao menos um falso positivo;
- b) $R_r = \frac{\text{requisitos corretos descobertos}}{\text{requisitos corretos disponíveis}}$. Se $R_r = 1$ significa que todos os impactos corretos possíveis foram descobertos, mas se $R_r < 1$, então existe ao menos um impacto correto que não foi descoberto.

As consequências para diferentes combinações de P_r e R_r estão na Figura 8 a seguir.

P_r e R_r	Consequências
$P_r = 1$ e $R_r = 1$	Estimativas podem ser justas;
$P_r = 1$ e $R_r < 1$	(1) Estimativas podem ser subvalorizadas; (2) Requisitos podem ficar desatualizados.
$P_r < 1$ e $R_r = 1$	(1) Estimativas podem ser supervalorizadas ou subvalorizadas; (2) Requisitos podem ser desnecessariamente analisados.
$P_r < 1$ e $R_r < 1$	(1) Estimativas podem ser supervalorizadas ou subvalorizadas; (2) Requisitos podem ser desnecessariamente analisados; (3) Requisitos podem ficar desatualizados.

Figura 8 - Consequências de P_r e R_r
Nota: Autoria Própria

A precisão (P_a) e a revocação (R_a) da descoberta de impactos nos artefatos associados ao requisito modificado:

- a) $P_a = \frac{\text{artefatos corretos descobertos}}{\text{artefatos descobertos}}$. Se $P_a = 1$ significa que todos os requisitos impactados descobertos são corretos, mas se $P_a < 1$, então existe ao menos um falso positivo;
- b) $R_a = \frac{\text{artefatos corretos descobertos}}{\text{artefatos corretos disponíveis}}$. Se $R_a = 1$ significa que todos os impactos corretos possíveis foram descobertos, mas se $R_a < 1$, então existe ao menos um impacto correto que não foi descoberto.

As consequências para diferentes combinações de P_a e R_a estão na Figura 9 a seguir.

P_a e R_a	Consequências
$P_a = 1$ e $R_a = 1$	Estimativas podem ser justas.
$P_a = 1$ e $R_a < 1$	(1) Estimativas podem ser subvalorizadas; (2) Artefatos podem ficar desatualizados.
$P_a < 1$ e $R_a = 1$	(1) Estimativas podem ser supervalorizadas ou subvalorizadas; (2) Artefatos podem ser desnecessariamente analisados.
$P_a < 1$ e $R_a < 1$	(1) Estimativas podem ser supervalorizadas ou subvalorizadas; (2) Artefatos podem ser desnecessariamente analisados; (3) Artefatos podem ficar desatualizados.

Figura 9 - Consequências de P_a e R_a
 Nota: Autoria Própria

3.5 TÉCNICAS DE RASTREABILIDADE DE REQUISITOS

Existe uma grande variedade de técnicas de rastreabilidade, desde as mais simples, como as baseadas em matriz de rastreabilidade, até outras mais sofisticadas, como a Event Based Traceability (CLELAND-HUANG; CHANG; CHRISTENSEN, 2003), que se baseia no estilo arquitetural publisher-subscriber para comunicar determinados eventos de mudança nos requisitos de software aos artefatos interessados. Além destas técnicas, outras técnicas foram estudadas em uma revisão sistemática planejada e executada por Raja e Kamran (2008). Nesta revisão, os autores investigaram algumas hipóteses sobre trabalhos publicados das quinze técnicas selecionadas. No Capítulo 4, apresentamos e discutimos os resultados de uma avaliação sistemática realizada sobre os mesmos trabalhos apresentados em Raja e Kamran (2008), porém utilizando um conjunto estendido de hipóteses.

4 ESTUDO DE TÉCNICAS DE RASTREABILIDADE DE REQUISITOS

Este Capítulo apresenta uma avaliação sistemática realizada sobre quinze técnicas de rastreabilidade referenciadas em uma revisão sistemática planejada e executada por Raja e Kamran (2008). Os resultados obtidos serviram de motivação para propor mudanças nas técnicas estudadas e identificar características desejáveis para guiar o projeto de uma nova técnica de rastreabilidade que apóie a análise de impacto de maneira mais satisfatória.

A Seção 4.1 apresenta os critérios para seleção das técnicas de rastreabilidade, bem como hipóteses, métricas e questões usadas ao longo da avaliação. Na Seção 4.2, são apresentados resumos das técnicas estudadas e os resultados da avaliação de cada técnica, considerando as questões definidas na Seção 4.3. A Seção 4.4 apresenta uma discussão geral sobre as técnicas estudadas, enfatizando implicações dos resultados obtidos em relação à análise de impacto e à proposta contida nesta dissertação.

4.1 METODOLOGIA

Nesta Seção, descrevemos os critérios de escolha das técnicas de rastreabilidade, além de hipóteses, questões e métricas utilizadas na avaliação.

4.1.1 Seleção das Técnicas de Rastreabilidade

As técnicas de rastreabilidade selecionadas para nosso estudo vieram de uma revisão sistemática planejada e executada por Raja e Kamran (2008). Neste estudo, os autores formularam quatro questões principais, sendo uma delas: “Quais são as diferentes técnicas de rastreabilidade?”. Eles revisaram quinze técnicas de rastreabilidade publicadas entre 1997 e 2007, incluindo sete técnicas revisadas em outro estudo (ROCHIMAH; KADIR; ABDULLAH, 2007).

Algumas das técnicas estudadas foram: Information Retrieval Approach (IRA) (MALETIC e outros, 2003), Event-Based Approach (EBT) (CLELAND-HUANG; CHANG; GE, 2002; CLELAND-HUANG; CHANG; CHRISTENSEN, 2003), Value-Based Requirement Traceability Approach (VBRT) (HEINDL; BIFFL, 2005; ZEMONT,

2005), Feature Model-Based Approach (FB) (PASHOV; RIEBISCH, 2004) e Scenario-Based Approach (SB) (EGYED, 2001).

4.1.2 Definição de Hipóteses, Questões e Métricas

Três hipóteses (H_1 , H_2 e H_3) nortearam nosso estudo, sendo que cada hipótese tem uma questão e uma métrica associadas.

- a) H_1 : A técnica necessita de intervenção manual para criar ou ajustar elos da rastreabilidade.
- b) Q_1 : A técnica necessita de intervenção manual para criar ou ajustar elos da rastreabilidade?
- c) M_1 : Indicativos nos trabalhos, que a técnica seja manual

Uma resposta positiva para a questão Q_1 significa que a hipótese H_1 é válida para a técnica em análise.

- a) H_2 : Quando um requisito muda, a técnica garante que os demais requisitos impactados sejam precisamente descobertos;
- b) Q_2 : Os valores de precisão e revocação são iguais a 100% em todos os experimentos de caracterização?
- c) M_2 : Valores de precisão (P_r) e revocação (R_r) reportados por experimentos de caracterização da técnica.

Uma resposta negativa para questão Q_2 significa que a hipótese H_2 não é válida para a técnica em análise para os experimentos reportados. As consequências de H_3 negada estão na Figura 8 a partir da segunda linha.

- a) H_3 : Quando um requisito muda, a técnica garante que os artefatos impactados sejam precisamente descobertos.
- b) Q_3 : Os valores de precisão e revocação são iguais a 100% em todos os experimentos de caracterização?
- c) M_3 : Valores de precisão (P_a) e revocação (R_a) reportados por experimentos de caracterização da técnica.

Uma resposta negativa para questão Q_3 significa que a hipótese H_3 não é válida para a técnica em análise para os experimentos reportados. As consequências de H_3 negada estão na Figura 9 a partir da segunda linha.

4.2 TÉCNICAS DE RASTREABILIDADE

Nesta Seção são apresentados os resumos de algumas das técnicas estudadas, selecionadas por apresentarem características distintas entre si e representativas. Para cada técnica avaliada, os resultados de sua respectiva avaliação são apresentados, de acordo com as questões introduzidas na Seção 4.1. A Figura 10 apresenta os resultados gerais de nossa avaliação para todas as quinze técnicas de rastreabilidade selecionadas.

Questões	Evidências	Consequências
Q₁: A técnica necessita de intervenção manual para criar ou ajustar elos da rastreabilidade?	Todas as técnicas necessitam	(1) Elos desatualizados devido ao negligenciamento da manutenção [1]; (2) Os elos gerados por técnicas automáticas precisam ser ajustados ou criados, devido ausência de uma semântica clara e precisa dos elos [3]; (3) Mesmo as técnicas que implementam alguma estratégia de priorização de requisitos, não impedem que que elos sejam manualmente ajustados [33, 14].
Q₂: Os valores de precisão (P_r) e revocação (R_r) são iguais a 100% em todos os experimentos de caracterização?	Nenhum estudo foi reportado.	Não há como garantir que impactos podem ser descobertos a qualquer momento do ciclo de vida do software, consequentemente estimativas não podem ser produzidas.
Q₃: Os valores de precisão (P_a) e revocação (R_a) são iguais a 100% em todos os experimentos de caracterização?	Os resultados foram obtidos apenas para técnicas automáticas	Os números estão da seguinte forma: $10\% \geq P_a \leq 35\%$ and $90\% \geq R_a \leq 95\%$ [3], o que pode trazer as seguintes consequências: (1) Devido P_a estar entre 10% and 35% cerca de 90% and 65% dos elos precisam ser ajustados manualmente, respectivamente; (2) Devido R_a estar entre 90% and 95% cerca de 10% and 5% precisam ser criados manualmente, respectivamente.

Figura 10 - Consolidação das Questões Q1, Q2 e Q3
Nota: Autoria Própria

4.2.1 Information Retrieval Approach (IRA)

O método Information Retrieval Approach (IRA) (ANTONIOLO e outros, 2002), baseado em recuperação de informação (RI), foi proposto com o objetivo de estabelecer e manter elos entre o código fonte e documentos de texto livre. O método baseia-se em um processo, dividido em duas etapas.

Na primeira etapa (document path), documentos são indexados com base em um vocabulário extraído dos próprios documentos. A construção deste vocabulário e a indexação dos documentos são precedidas pela fase de normalização, que inclui transformação de letras em caixa alta para caixa baixa, remoção de artigos, pontuação, números e outros elementos, e análise morfológica para converter palavras no plural em palavras no singular e para transformar verbos em suas formas infinitivas.

Na segunda etapa (code path), constrói-se e indexa-se uma consulta para cada componente de código fonte. A construção da consulta inclui atividades para analisar o componente de código fonte e extrair uma lista de identificadores, separa as palavras de identificadores formados por duas ou mais palavras, e aplicar os passos da etapa document path para indexar os documentos. Após a indexação concluída, sempre que um par de documentos alcançarem mais de uma classificação, os elos candidatos devem ser escolhidos manualmente.

4.2.1.1 Avaliação

Embora a técnica IRA seja uma técnica de recuperação automática de informação, a questão Q_1 tem resposta “positiva”. Isto se deve em parte a falta de semântica apropriada dos elos de satisfação (CLELAND-HUANG, 2006), o que acarreta valores de P_a e R_a inferiores a 100% obtidos nos estudos. Deste modo, a resposta para Q_3 é “negativa”, enquanto para Q_2 é “não se aplica”, pois os trabalhos não reportaram valores de P_r e R_r .

4.2.2 Event-Based Approach (EBT)

Na técnica Event-Based Approach (EBT) (CLELAND-HUANG; CHANG; GE, 2002; CLELAND-HUANG; CHANG; CHRISTENSEN, 2003) relacionamentos entre requisitos e artefatos dependentes são criados, de modo que os artefatos dependentes sejam notificados sempre que mudanças ocorram nos requisitos dos quais dependem. Os eventos encaminhados são descritos por operações que podem ser realizadas sobre os elos de dependência ou sobre o próprio requisito:

- a) **Create**: para criar um novo requisito. $\rightarrow r_i$.
- b) **Modify**: para modificar atributos do elo associado ao requisito, por exemplo, o seu status. $r_i:Status(velho) \rightarrow r_i:Status(novo)$.
- c) **Merge**: $r_i + r_j + \dots + r_m \rightarrow r_n$, para formar um novo requisito a partir de mais de um requisito;
- d) **Refine**: $r_i \rightarrow r_i + r_j + \dots + r_n$, para adicionar partes a um requisito.
- e) **Decompose**: $r_i \rightarrow r_j + r_k + \dots + r_n$, para formar novos requisitos a partir das partes de um requisito existente.
- f) **Replace**: $r_i \rightarrow r_j$, para substituir um requisito por outro.

4.2.2.1 Avaliação

A técnica EBT, embora torne automática a comunicação da mudança, ainda assim existe a necessidade de ajuste manual dos elos, portanto a resposta para Q1 é “positiva”. As respostas para Q2 e Q3 são “não se aplica”, pois em nenhum dos trabalhos avaliados valores de precisão P_a e P_r e revocação R_a e R_r foram reportados. A técnica prioriza o relacionamento entre requisitos e artefatos, o que nos leva a concluir que não haja interesse por P_r e R_r .

4.2.3 Value-Based Requirement Traceability

A técnica Value-Based Requirement Traceability (VBRT) tem como objetivo reduzir a quantidade de informações presentes na rastreabilidade (HEINDL; BIFFL, 2005; ZEMONT, 2005). O esforço para se manter a rastreabilidade é reduzido pela priorização dos requisitos feita pelos interessados (stakeholders). Inclui atividades de definição de requisitos, priorização de requisitos, empacotamento de requisitos e

criação de elos entre artefatos. Nesta última atividade, os membros do projeto estabelecem os elos da rastreabilidade entre requisitos e artefatos. Os requisitos importantes são rastreados com maior nível de detalhe do que os menos importantes. Por fim, os elos criados podem ser usados para estimar o impacto de mudanças sobre os requisitos.

4.2.3.1 Avaliação

A resposta é “positiva” para a questão Q_1 , pois a técnica necessita de intervenção manual mesmo com a priorização. Na verdade a priorização em si é uma intervenção manual.

A resposta “não se aplica” para as questões Q_2 e Q_3 decorre do fato de que os estudos de casos apresentados (HEINDL; BIFFL, 2005; ZEMONT, 2005) não reportam valores de precisão e revocação, apenas de esforço associado à manutenção dos elos da rastreabilidade. A única etapa de criação de elos envolve apenas requisitos e artefatos, nada é mencionado acerca de elos entre requisitos. Este fato, leva-nos a concluir que P_r e R_r não sejam relevantes para a técnica.

4.2.4 Feature Model-Base Approach (FB)

A técnica de Feature Model-Based Approach (FB) usa o modelo de características (features), descrito por Pashov e Riebisch (2004) no contexto de linhas de produto. As características representam abstrações de requisitos e são intermediárias entre estes e os “elementos de design” que as implementam (PASHOV; RIEBISCH, 2004).

Na Figura 11, a característica (feature) intermedia o relacionamento entre requisitos e “elementos de design” (Design Element). Este modelo traz vantagens como a manutenção de um número menor de elos. Por exemplo: se m fosse o número de requisitos relacionados a uma característica, e esta se relacionasse com outros n elementos de design, então $m + n$ seria o total de elos formados, enquanto que, se o relacionamento fosse direto entre os requisitos e os elementos de design, a quantidade de elos seria $m*n$.

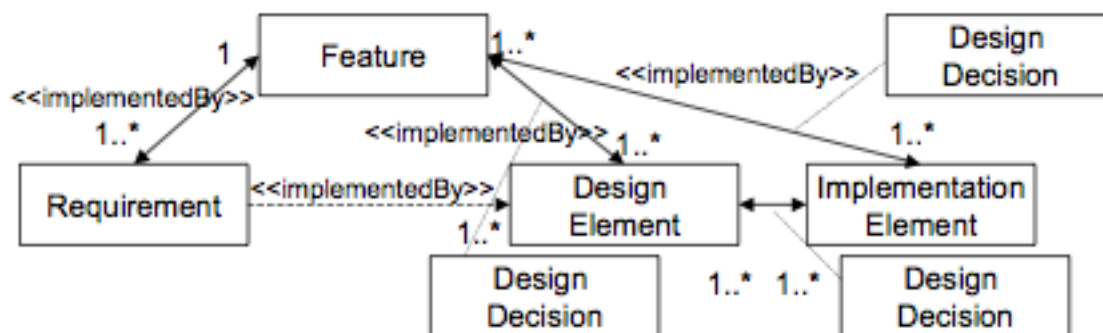


Figura 11 - Elos entre requisitos, feature e elementos de design
 Nota: Extraída de Pashov e Riebisch (2004)

4.2.4.1 Avaliação

A técnica FB é manual, portanto a resposta para Q1 é “positiva”. A resposta para as questões Q2 e Q3 são “não se aplica”, pois o estudo de caso apresentado em (PASHOV; RIEBISCH, 2004) não reporta valores de precisão e revocação, apenas a construção da rastreabilidade.

4.2.5 Scenario-Based Approach (SB)

A técnica de Scenario-Based Approach (SB) (EGYED, 2001) é uma abordagem que adiciona elementos de reengenharia na detecção e validação de informações da rastreabilidade. A abordagem requer: (i) a existência de um software executável passível de ser observado através de alguma ferramenta de monitoramento, (ii) um modelo do software, e (iii) a descrição de cenários de testes ou cenários de uso de software ou de componentes.

Basicamente, a técnica consiste em criar elos de hipóteses entre os elementos do modelo e o cenário, e entre os elementos do modelo e o software ou componentes. Em seguida, o software é executado para se verificar, através de casos de testes, se os elos de hipóteses correspondem aos elos observados durante o monitoramento. Sempre que existir um desvio entre os elos de hipóteses e os elos observados durante o monitoramento, os elos de hipóteses são atualizados para contemplar o que foi observado.

4.2.5.1 Avaliação

A resposta para Q_1 é “positiva”, pois a técnica necessita de intervenção manual. Não foram encontrados experimentos para técnica SB, resultando na resposta “não se aplica” para as questões Q_2 e Q_3 . A técnica SB não considera o relacionamento entre requisitos diretamente, isto indica que P_r e R_r não são relevantes para a técnica.

4.3 DISCUSSÃO

A Figura 10 (Seção 4.2) mostra resumo do estudo sistemático realizado sobre as técnicas de rastreabilidade selecionadas. As respostas dadas às questões para as técnicas analisadas são discutidas em termos de possíveis implicações negativas para a análise de impacto. Por fim, algumas considerações são apresentadas.

Implicações da resposta “positiva” para Q_1 :

- a) A inconsistência dos elos é um fenômeno que ocorre pelo negligenciamento da manutenção da rastreabilidade. Isto se deve principalmente pelos mantenedores da rastreabilidade, perceberem mais custos que benefícios na atividade de manutenção dos elos. Neste caso, entende-se que o esforço despendido seja inútil (SAYÃO; LEITE, 2006). Uma das causas do negligenciamento é o grande volume de elos que precisam ser mantidos consistentes com a evolução dos artefatos (ANTONIOLO e outros, 2002). Um ciclo vicioso pode ser iniciado se o negligenciamento da manutenção dos elos levar à evolução separada dos artefatos (EGYED, 2001) e esta leva ao negligenciamento, e assim por diante. Para a atividade de descoberta de impactos, a inconsistência dos elos traz sérias implicações de precisão e revocação, explicados com mais detalhes a seguir.
- b) Algumas técnicas de rastreabilidade adotaram critérios para priorizar o que deve ou não participar da rastreabilidade, com o objetivo de reduzir o grande volume de elos para serem mantidos manualmente. As técnicas apresentadas em (ZEMONT, 2005; HEINDL; BIFFL, 2005) adotaram a estratégia de priorizar requisitos através de um processo manual que se

dá para cada novo conjunto de requisitos. Neste processo, os interessados elegem os requisitos mais críticos para serem incluídos na rastreabilidade. Embora a priorização contribua para reduzir o volume de elos, por outro lado, não garante que futuras intervenções manuais, sejam necessárias, para realizar ajustes da rastreabilidade.

Implicações das respostas “negativa” ou “não se aplica” para Q₂:

- a) Qualquer imprecisão mínima nos resultados dos impactos associados a uma mudança pode representar um grande desvio nas estimativas finais da análise de impacto. A ocorrência de falsos positivos significa que requisitos foram analisados sem necessidade, o que além de tornar a análise de impacto menos eficiente, podem fazer com que as estimativas sejam acrescidas de custos e prazos extras.
- b) Além destes problemas, a maioria das técnicas de rastreabilidade não fornece suporte a registro de elos para requisitos novos (CLELAND-HUANG; CHANG; GE, 2002), não dispendo portanto de elos de dependência. Este fato impõe sérias limitações à capacidade da rastreabilidade de suportar a atividade de descoberta de impacto em qualquer momento do ciclo de vida do software. Esta capacidade é fundamental para requisitos de negócio que podem sofrer mudanças a qualquer momento.

Implicações das respostas “negativa” ou “não se aplica” para Q₃:

- a) As mesmas implicações negativas apontadas para a questão Q₂ ocorrem para Q₃, apenas trocando artefato por requisito. Um risco importante a ser considerado é o de se atualizar os artefatos falsos positivos.
- b) Um pequeno desvio na revocação da descoberta dos impactos de mudanças significa que artefatos deixaram de ser considerados. Devido à ausência de artefatos, os custos e prazos podem ser subestimados. Um risco importante a ser considerado é o da desatualização dos artefatos que ficaram ausentes da análise.

Em nosso estudo, ficou clara a baixa prioridade por parte das técnicas em revelar valores de precisão e revocação para descoberta de impacto entre

requisitos. Acreditamos que este desinteresse venha da ausência de semântica apropriada dos modelos para representar elos de dependência. Tratar todos os elos de dependência de igual forma significa não conseguir capturar a semântica da dependência em si. Assim, não se tem a capacidade a partir do modelo, uma vez definida a rastreabilidade, responder: o que determinou a criação de cada elo de dependência. Devido a esta incapacidade, a manutenção da rastreabilidade é sempre um processo separado da manutenção do modelo em si.

As técnicas de rastreabilidade estudadas não são capazes de lidar com a separação do interesse dos que se preocupam com o negócio e do interesse dos que se preocupam com os artefatos de software. Mesmo técnicas que dispõem de elos de dependência, estes elos não dispõem de semântica clara e precisa (CLELAND-HUANG, 2006) de modo que representem as relações específicas de dependência entre regras de negócio. A ausência de uma semântica apropriada dos elos impossibilita que verificações de completude da especificação de requisitos, com o foco no negócio, possam ser feitas com o auxílio da rastreabilidade, uma vez que certas exigências específicas das relações entre regras de negócio não precisam ser cumpridas. Esta decorrência também impede que a rastreabilidade sirva de base para a descoberta de impactos no negócio, não atendendo, portanto os interessados no negócio.

As técnicas de rastreabilidade de requisitos se limitam a seguir a vida dos requisitos enquanto parte do software. Mas, um requisito de negócio pode ter dependências ou ser dependente de outros requisitos de negócio ou de outras regras de negócio que não fazem parte do software. Esta incapacidade impossibilita qualquer técnica de rastreabilidade de requisitos de descobrir impactos no negócio que não é software ou descobrir impactos no software para mudanças fora deste.

5 A TÉCNICA FIR

Esse trabalho apresenta uma nova técnica de rastreabilidade chamada de FIR (Funcionalidade-Informação-Regra). A técnica relaciona os elementos definidos originalmente em (OLIVEIRA FILHO; CHAVEZ; MENDONÇA NETO, 2008) de modo a compreender o conceito de interesse de negócio proposto neste trabalho através da Definição 1 e Definição 2 e ilustrado na Figura 12.

Definição 1: Interesses de Negócio são agrupamentos de regras de negócio que compartilham os mesmos propósitos, de modo que, quando uma mudança é solicitada para uma regra de negócio de um determinado agrupamento, isso significa que existem interessados da organização em descobrir os impactos relacionados a essa mudança que se manifestam no outro agrupamento.

Definição 2: Propósitos podem ser vistos como elos que relacionam regras de negócio de interesses de negócio distintos entre si através do compartilhamento de informações.

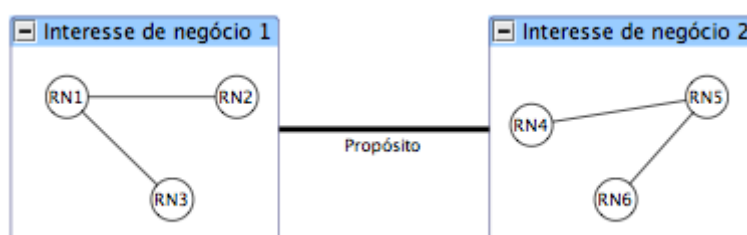


Figura 12 – Visão Geral do Modelo FIR
Nota: Autoria Própria

Na Seção 5.1, a técnica FIR é apresentada e seus passos são descritos nas Seções seguintes. O exemplo introduzido no Capítulo 1 – Venda de Ingressos para partidas de futebol – será utilizado para ilustrar as definições.

5.1 VISÃO GERAL DE TÉCNICA FIR

A técnica FIR é uma interpretação da Definição 1 de interesses de negócio. A técnica permite o relacionamento através de elos de dependência especializados (a) entre regras de negócio e (b) entre regras de negócio e requisitos de negócio (e vice-versa). No modelo FIR, um interesse de negócio é a unidade básica fundamental para se descobrir e, posteriormente, analisar impactos de uma mudança.

A técnica está estruturada em três passos:

- a) Construção de interesses de negócio, em que são identificadas e construídas dependências entre as regras de negócio, independentemente da existência de um software que as implemente;
- b) Interpretação de interesses de negócio para requisitos de negócio, onde as regras de negócio identificadas no passo anterior são mapeadas para requisitos de software;
- c) Delimitação das fronteiras, isto é, entre os requisitos de negócio do software e as regras de negócio.

Esses passos são explicados nas próximas seções, incluindo definições rigorosas dos elementos propostos por Oliveira Filho, Chavez e Mendonça Neto (2008).

5.2 IDENTIFICAÇÃO E CONSTRUÇÃO DE INTERESSES DE NEGÓCIO

Inicialmente, é necessário estabelecer uma correspondência entre os tipos de regras descritos no Capítulo 2 e os tipos Funcionalidade, Regra e Informação definidos por Oliveira Filho, Chavez e Mendonça Neto (2008) e apresentados na Figura 13. A correspondência é apresentada na Figura 14.

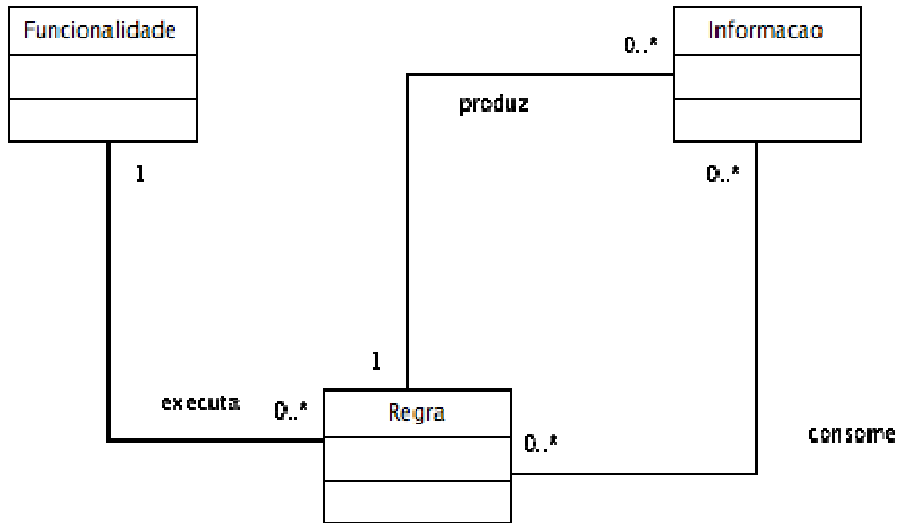


Figura 13 - Modelo FIR
Nota: Autoria Própria

Halle	Oliveira
Action Anabler	Functionality
Constraint	Rule
Guideline	Rule
Computation	Rule
Inference	Rule
Term	Information

Figura 14 – Regras de Halle x Oliveira
Nota: Autoria Própria

5.2.1 Obtenção de Subconjuntos os Tipos de Regras de Negócio

A técnica FIR estabelece o relacionamento entre as regras de negócio descritas na Definição 3.

Definição 3: Seja $R^n = \{r_1, r_2, \dots, r_k\}$ um conjunto não-vazio de regras de negócio utilizadas por uma organização para conduzir um negócio N qualquer (denotado por n sobrescrito em R). As regras de negócio são atômicas e estão de acordo com a Figura 14 de correspondência.

Os três elementos base da técnica FIR são a funcionalidade, a informação e a regra que estão rigorosamente definidos a seguir.

Definição 4: Seja $F^n = \{r_{a_1}, r_{a_2}, \dots, r_{a_z}\}$ um subconjunto não-vazio de R^n (Definição 3), tal que, $r_{a_z} \in F^n$, se r_{a_z} for um elemento do tipo Funcionalidade. O elemento r_{a_z} corresponde à própria ação da regra de negócio do tipo Capacita Ação.

Definição 5: Seja $B^n = \{r_{b_1}, r_{b_2}, \dots, r_{b_j}\}$ um subconjunto não-vazio de R^n (Definição 3), tal que, $r_{b_j} \in B^n$, se r_{b_j} for um elemento do tipo Regra.

Definição 6: Seja $I^n = \{r_{i_1}, r_{i_2}, \dots, r_{i_p}\}$ um subconjunto não-vazio de R^n (Definição 3), tal que, $r_{i_p} \in I^n$, se r_{i_p} for um elemento do tipo Informação. Os elementos do conjunto I^n são informações utilizadas por uma organização para conduzir um negócio N qualquer.

Exemplo. Os conjuntos F^n , B^n e I^n estão descritos na Figura 15 para o exemplo apresentado no Capítulo 1.

F^n	r_{a_1} : Cadastrar Torcedor
	r_{a_2} : Emitir Ingresso
	r_{a_3} : Liberar Acesso
B^n	r_{b_1} : Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado
	r_{b_2} : Se Nome do Evento informado for válido, então o Ingresso-Nome-do-Evento é igual ao Nome do Evento informado
	r_{b_3} : Se a Data do Evento informada for válida, então o ingresso-Data-do-Evento é igual à Data do Evento informada
	r_{b_4} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado
	r_{b_5} : O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado
	r_{b_6} : O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado
	r_{b_7} : O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado
I^n	r_{i_1} : Torcedor-CPF
	r_{i_2} : Ingresso-CPF
	r_{i_3} : Ingresso-Data-do-Evento
	r_{i_4} : Ingresso-Nome-do-Evento

Figura 15 – O exemplo de Emissão de Ingressos
Nota: Autoria Própria

5.2.2 Relações entre os Tipos de Regras de Negócio

Esta Seção aborda as definições 7, 8 e 9 que tratam do relacionamento entre os elementos dos conjuntos F^n , B^n e I^n entre si, ora com o intuito de agrupar elementos ora para estabelecer uma relação de dependência entre os elementos.

A Definição 7 a seguir indicada pela letra grega Θ (Theta) e sobrescrita pela letra n indicadora de um negócio N qualquer, é um agrupamento de elementos do conjunto B^n das regras de negócio do tipo Regra, por elementos do conjunto F^n das regras de negócio do tipo Funcionalidade. Para o FIR, elementos de B^n devem sempre estar agrupados aos elementos de F^n . Em outras palavras: quando uma funcionalidade é executada, o correspondente elemento de Θ^n indica quais regras potencialmente também deveriam ser executadas.

Definição 7: Seja $\Theta^n = \{\theta_{r_j} = (r_{b_j}, r_{a_z}) \mid r_{b_j} \in B^n \wedge r_{a_z} \in F^n\}$ um subconjunto não-vazio formado pela relação entre elementos do tipo regra e elementos do tipo funcionalidade. O par $\theta_{r_j} = (r_{b_j}, r_{a_z}) \in \Theta^n$, se a funcionalidade r_{a_z} executar a regra r_{b_j} .

Exemplo. O conjunto Θ^n para o exemplo apresentado no Capítulo 1 está descrito na Figura 16.

Θ^n <hr/> $\theta_{r_1} = (r_{b_1}, r_{a_1})$ <hr/> $\theta_{r_2} = (r_{b_2}, r_{a_2})$ <hr/> $\theta_{r_3} = (r_{b_3}, r_{a_2})$ <hr/> $\theta_{r_4} = (r_{b_4}, r_{a_2})$ <hr/> $\theta_{r_5} = (r_{b_5}, r_{a_3})$ <hr/> $\theta_{r_6} = (r_{b_6}, r_{a_3})$ <hr/> $\theta_{r_7} = (r_{b_7}, r_{a_3})$	F^n <hr/> r_{a_1} : Cadastrar Torcedor. <hr/> r_{a_2} : Emitir Ingresso. <hr/> r_{a_3} : Liberar Acesso. <hr/> B^n <hr/> r_{b_1} : Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado. <hr/> r_{b_2} : Se Nome do Evento informado for válido, então o Ingresso-Nome-do-Evento é igual ao Nome do Evento informado. <hr/> r_{b_3} : Se a Data do Evento informada for válida, então o ingresso-Data-do-Evento é igual à Data do Evento informada. <hr/> r_{b_4} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado. <hr/> r_{b_5} : O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado. <hr/> r_{b_6} : O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado. <hr/> r_{b_7} : O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado.
---	---

Figura 16 - O Conjunto Θ^n para o exemplo de Emissão de Ingressos
Nota: Autoria Própria

A Definição 8 a seguir indicada pela letra grega Γ (Gamma) combinada com o símbolo indicador de produção $\hat{}$ (Chapéu) e sobrescrita pela letra n indicadora de um negócio N qualquer, é um subconjunto da relação $B^n \times I^n$. Os pares de $\hat{\Gamma}^n$ indicam a relação de produção de informação. Para o FIR: são as regras de B^n que produzem elementos do conjunto I^n de informações.

Definição 8: Seja $\hat{\Gamma}^n = \{\gamma_{pj} = (r_{bj}, r_{ip}) \mid r_{bj} \in B^n \wedge r_{ip} \in I^n\}$ um subconjunto não-vazio formado pela relação elementos do tipo regra e elementos do tipo informação. O par $\gamma_{pj} = (r_{bj}, r_{ip}) \in \hat{\Gamma}^n$, se a informação r_{ip} resultar da regra r_{bj} por meio de uma computação ou de uma inferência. A regra r_{bj} é denominada regra produtora da informação r_{ip} .

Exemplo. O conjunto $\hat{\Gamma}^n$ para o exemplo apresentado no Capítulo 1 está descrito na Figura 17.

$\hat{\Gamma}^n$ <hr/> $\hat{\gamma}_{r_1} = (r_{b_1}, r_{i_1})$ <hr/> $\hat{\gamma}_{r_2} = (r_{b_2}, r_{i_4})$ <hr/> $\hat{\gamma}_{r_3} = (r_{b_3}, r_{i_3})$ <hr/> $\hat{\gamma}_{r_4} = (r_{b_4}, r_{i_2})$	B^n
	r_{b_1} : Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado
	r_{b_2} : Se Nome do Evento informado for válido, então o Ingresso-Nome-do-Evento é igual ao Nome do Evento informado
	r_{b_3} : Se a Data do Evento informada for válida, então o ingresso-Data-do-Evento é igual à Data do Evento informada
	r_{b_4} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado
	I^n
	r_{i_1} : Torcedor-CPF
	r_{i_2} : Ingresso-CPF
	r_{i_3} : Ingresso-Data-do-Evento
	r_{i_4} : Ingresso-Nome-do-Evento

Figura 17 – O Conjunto $\hat{\Gamma}^n$ para o exemplo Emissão de Ingresso Digite a equação aqui.

Nota: Autoria Própria

A Definição 9 a seguir indicada pela letra grega Γ (Gamma) combinada com o símbolo indicador de consumo $\check{}$ (Chapéu invertido) e sobrescrita pela letra n indicadora de um negócio N qualquer, é um subconjunto da relação $B^n \times I^n$. Os pares de $\check{\Gamma}^n$ indicam a relação de consumo de informação. Para o FIR: são as regras de B^n que consomem elementos do conjunto I^n de informações.

Definição 9: Seja $\check{\Gamma}^n = \{\gamma_{rj} = (r_{bj}, r_{ip}) \mid r_{bj} \in B^n \wedge r_{ip} \in I^n\}$ um subconjunto não-vazio formado pela relação elementos do tipo regra e elementos do tipo

Informação. O par $\gamma_{r_j}=(r_{b_j},r_{i_p}) \in \check{\Gamma}^n$, se a informação r_{i_p} for utilizada (consumida) pela regra de negócio r_{b_j} . A regra r_{b_j} é denominada regra consumidora da informação r_{i_p} .

Exemplo. O conjunto $\check{\Gamma}^n$ para o exemplo apresentado no Capítulo 1 está descrito na Figura 18.

$\check{\Gamma}^n$	B^n
	<i>r_{b_4}: Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado</i>
	<i>r_{b_5}: O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado</i>
	<i>r_{b_6}: O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado</i>
	<i>r_{b_7}: O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado</i>
	I^n
	<i>r_{i_1}: Torcedor-CPF</i>
	<i>r_{i_2}: Ingresso-CPF</i>
<i>r_{i_3}: Ingresso-Data-do-Evento</i>	
<i>r_{i_4}: Ingresso-Nome-do-Evento</i>	

Figura 18 – O Conjunto $\check{\Gamma}^n$ para o exemplo de Emissão de Ingresso

Nota: Autoria Própria

5.2.3 Agrupamentos entre Funcionalidade, Regra e Informação

Esta Seção trata dos agrupamentos $\hat{\Psi}^n$ (Definição 10) e $\check{\Psi}^n$ (Definição 11) que relacionam os elementos de Θ^n (Definição 7), $\hat{\Gamma}^n$ (Definição 8) e $\check{\Gamma}^n$ (Definição 9) entre si.

A Definição 10 a seguir indicada pela letra grega Ψ (Psi) combinada com o símbolo indicador de produção $\hat{\ } (Chapéu)$ e sobrescrita pela letra n indicadora de um negócio N qualquer, é uma relação entre as relações Θ^n (Definição 7) e $\hat{\Gamma}^n$ (Definição 8). Para o FIR, os pares de Θ^n e os pares de $\hat{\Gamma}^n$ devem participar de uma relação cujo significado é dado como: a funcionalidade que executa potencialmente a regra que produz a informação.

Definição 10: Seja $\hat{\Psi}^n = \{\psi=(\theta_{r_j},\gamma_{r_p}) \mid \theta_{r_j} \in \Theta^n \wedge \gamma_{r_p} \in \hat{\Gamma}^n\}$. O par $\psi=(\theta_{r_j},\gamma_{r_p}) \in \hat{\Psi}^n$, para $\theta_{r_j}=(r_{b_j},r_{a_z})$ e $\gamma_{r_j}=(r_{b_w},r_{i_p})$, se $r_{b_j}=r_{b_w}$. O par ψ é denominado *semi-interesse de negócio produtor*.

Exemplo. O conjunto $\hat{\Psi}^n$ para o exemplo apresentado no Capítulo 1 está descrito na Figura 19.

$\hat{\Psi}^n$	F^n
$\hat{\psi}_{r_1} = \left(\overbrace{((r_{b_1}, r_{a_1}), (r_{b_1}, r_{i_1}))}^{\theta_{r_1}}, \overbrace{(\hat{\gamma}_{r_1})}^{\hat{\gamma}_{r_1}} \right)$	r_{a_1} : Cadastrar Torcedor
$\hat{\psi}_{r_2} = \left(\overbrace{((r_{b_2}, r_{a_2}), (r_{b_2}, r_{i_4}))}^{\theta_{r_2}}, \overbrace{(\hat{\gamma}_{r_2})}^{\hat{\gamma}_{r_2}} \right)$	r_{a_2} : Emitir Ingresso
$\hat{\psi}_{r_3} = \left(\overbrace{((r_{b_3}, r_{a_2}), (r_{b_3}, r_{i_3}))}^{\theta_{r_3}}, \overbrace{(\hat{\gamma}_{r_3})}^{\hat{\gamma}_{r_3}} \right)$	r_{a_3} : Liberar Acesso
$\hat{\psi}_{r_4} = \left(\overbrace{((r_{b_4}, r_{a_2}), (r_{b_4}, r_{i_2}))}^{\theta_{r_4}}, \overbrace{(\hat{\gamma}_{r_4})}^{\hat{\gamma}_{r_4}} \right)$	B^n
	r_{a_1} : Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado
	r_{b_2} : Se Nome do Evento informado for válido, então o Ingresso-Nome-do-Evento é igual ao Nome do Evento informado
	r_{b_3} : Se a Data do Evento informada for válida, então o ingresso-Data-do-Evento é igual à Data do Evento informada
	r_{b_4} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado
	I^n
	r_{i_1} : Torcedor-CPF
	r_{i_2} : Ingresso-CPF
	r_{i_3} : Ingresso-Data-do-Evento
	r_{i_4} : Ingresso-Nome-do-Evento

Figura 19 – O Conjunto $\hat{\Psi}^n$ para o exemplo de Emissão de Ingresso

Nota: Autoria Própria

A Definição 11 a seguir indicada pela letra grega Ψ (Psi) combinada com o símbolo indicador de consumo \sim (Chapéu invertido) e sobrescrita pela letra n indicadora de um negócio N qualquer, é uma relação entre as relações θ^n (Definição 7) e $\check{\Gamma}^n$ (Definição 9). Para o FIR, os pares de θ^n e os pares de $\check{\Gamma}^n$ devem participar de uma relação cujo significado é dado como: a funcionalidade que executa potencialmente a regra que consome a informação.

Definição 11: Seja $\check{\Psi}^n = \{\psi = (\theta_{r_j}, \gamma_{r_p}) \mid \theta_{r_j} \in \theta^n \wedge \gamma_{r_p} \in \check{\Gamma}^n\}$. O par $\psi = (\theta_{r_j}, \gamma_{r_p}) \in \check{\Psi}^n$, para $\theta_{r_j} = (r_{b_j}, r_{a_z})$ e $\gamma_{r_p} = (r_{b_w}, r_{i_p})$, se $r_{b_j} = r_{b_w}$. O par ψ é denominado *semi-interesse de negócio consumidor*.

Exemplo. O conjunto $\check{\Psi}^n$ para o exemplo apresentado no Capítulo 1 está descrito na Figura 20.

$\check{\Psi}^n$	F^n
$\check{\psi}_{r_1} = \left(\overbrace{(r_{b_4}, r_{a_2})}^{\theta_{r_4}}, \overbrace{(r_{b_4}, r_{i_1})}^{\check{\gamma}_{r_5}} \right)$	r_{a_1} : Cadastrar Torcedor
$\check{\psi}_{r_2} = \left(\overbrace{(r_{b_5}, r_{a_3})}^{\theta_{r_5}}, \overbrace{(r_{b_5}, r_{i_4})}^{\check{\gamma}_{r_6}} \right)$	r_{a_2} : Emitir Ingresso
$\check{\psi}_{r_4} = \left(\overbrace{(r_{b_7}, r_{a_3})}^{\theta_{r_7}}, \overbrace{(r_{b_7}, r_{i_3})}^{\check{\gamma}_{r_8}} \right)$	r_{a_3} : Liberar Acesso
$\check{\psi}_{r_3} = \left(\overbrace{(r_{b_6}, r_{a_3})}^{\theta_{r_6}}, \overbrace{(r_{b_6}, r_{i_2})}^{\check{\gamma}_{r_7}} \right)$	B^n
	r_{b_4} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado
	r_{b_5} : O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado
	r_{b_6} : O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado
	r_{b_7} : O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado
	I^n
	r_{i_1} : Torcedor-CPF
	r_{i_2} : Ingresso-CPF
	r_{i_3} : Ingresso-Data-do-Evento
	r_{i_4} : Ingresso-Nome-do-Evento

Figura 20 – O Conjunto $\check{\Psi}^n$ para o exemplo de Emissão de Ingressos
Nota: Autoria Própria

5.2.4 A Relação Interesse de Negócio

A Definição 12 a seguir indicada pela letra grega Φ (Phi) e sobrescrita pela letra n indicadora de um negócio N qualquer, é uma relação entre as relações $\hat{\Psi}^n$ (Definição 10) e $\check{\Psi}^n$ (Definição 11). Para o FIR, os pares de $\hat{\Psi}^n$ e os pares de $\check{\Psi}^n$ devem participar de uma relação cujo significado é o mesmo da Definição 1 de interesse de negócio: o agrupamento de regras em torno de um mesmo propósito para que impactos de mudanças possam ser identificados.

Definição 12: Seja $\Phi^n = \{\varphi = (\hat{\psi}_{r_j}, \check{\psi}_{r_p}) \mid \hat{\psi}_{r_j} \in \hat{\Psi}^n \wedge \check{\psi}_{r_p} \in \check{\Psi}^n\}$. O par $\varphi = (\hat{\psi}_{r_j}, \check{\psi}_{r_p})$, se a informação r_{i_p} produzida a partir de $\hat{\psi}_{r_j}$ for a mesma consumida por $\check{\psi}_{r_p}$. Os pares $\hat{\psi}_{r_j}$ e $\check{\psi}_{r_p}$ são agora denominados de interesse de negócio produtor e interesse de negócio consumidor, respectivamente. A informação r_{i_p} é denominada propósito de $\hat{\psi}_{r_j}$ para $\check{\psi}_{r_p}$.

Exemplo. O conjunto Φ^n para o exemplo apresentado no Capítulo 1 está descrito na Figura 21.

Φ^n	F^n
$\underbrace{\underbrace{\theta_{r_1}}_{((r_{b_1}, r_{a_1}), (r_{b_1}, r_{i_1}))}, \underbrace{\hat{\gamma}_{r_1}}_{((r_{b_4}, r_{a_2}), (r_{b_4}, r_{i_1}))}}_{\hat{\psi}_{r_1}}, \underbrace{\underbrace{\theta_{r_4}}_{((r_{b_4}, r_{a_2}), (r_{b_4}, r_{i_1}))}, \underbrace{\check{\gamma}_{r_5}}_{((r_{b_4}, r_{i_1}))}}_{\check{\psi}_{r_1}}$	r_{a_1} : Cadastrar Torcedor
$\underbrace{\underbrace{\theta_{r_2}}_{((r_{b_2}, r_{a_2}), (r_{b_2}, r_{i_4}))}, \underbrace{\hat{\gamma}_{r_2}}_{((r_{b_5}, r_{a_3}), (r_{b_5}, r_{i_4}))}}_{\hat{\psi}_{r_2}}, \underbrace{\underbrace{\theta_{r_5}}_{((r_{b_5}, r_{a_3}), (r_{b_5}, r_{i_4}))}, \underbrace{\check{\gamma}_{r_6}}_{((r_{b_5}, r_{i_4}))}}_{\check{\psi}_{r_2}}$	r_{a_2} : Emitir Ingresso r_{a_3} : Liberar Acesso
$\underbrace{\underbrace{\theta_{r_3}}_{((r_{b_3}, r_{a_2}), (r_{b_3}, r_{i_3}))}, \underbrace{\hat{\gamma}_{r_3}}_{((r_{b_7}, r_{a_3}), (r_{b_7}, r_{i_3}))}}_{\hat{\psi}_{r_3}}, \underbrace{\underbrace{\theta_{r_7}}_{((r_{b_7}, r_{a_3}), (r_{b_7}, r_{i_3}))}, \underbrace{\check{\gamma}_{r_8}}_{((r_{b_7}, r_{i_3}))}}_{\check{\psi}_{r_4}}$	B^n
$\underbrace{\underbrace{\theta_{r_4}}_{((r_{b_4}, r_{a_2}), (r_{b_4}, r_{i_2}))}, \underbrace{\hat{\gamma}_{r_4}}_{((r_{b_6}, r_{a_3}), (r_{b_6}, r_{i_2}))}}_{\hat{\psi}_{r_4}}, \underbrace{\underbrace{\theta_{r_6}}_{((r_{b_6}, r_{a_3}), (r_{b_6}, r_{i_2}))}, \underbrace{\check{\gamma}_{r_7}}_{((r_{b_6}, r_{i_2}))}}_{\check{\psi}_{r_3}}$	r_{b_1} : Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado r_{b_2} : Se Nome do Evento informado for válido, então o Ingresso-Nome-do-Evento é igual ao Nome do Evento informado r_{b_3} : Se a Data do Evento informada for válida, então o ingresso-Data-do-Evento é igual à Data do Evento informada r_{b_4} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado r_{b_5} : O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado r_{b_6} : O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado r_{b_7} : O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado
	I^n
	r_{i_1} : Torcedor-CPF r_{i_2} : Ingresso-CPF r_{i_3} : Ingresso-Data-do-Evento r_{i_4} : Ingresso-Nome-do-Evento

Figura 21 – O Conjunto Φ^n para o exemplo de Emissão de Ingressos

Nota: Autoria Própria

A Figura 22 mostra graficamente o resultado da aplicação dessa primeira etapa da técnica FIR para o exemplo apresentado no Capítulo 1.

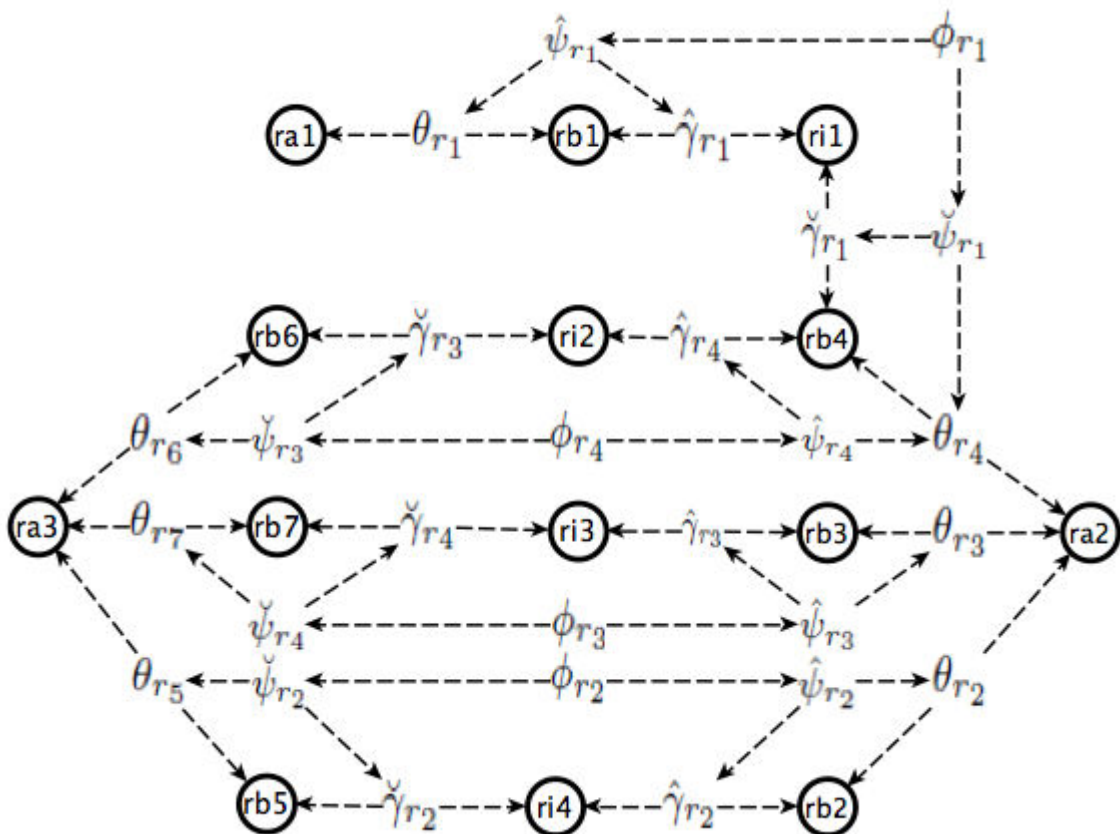


Figura 22 – Uma Visão Gráfica do FIR para o exemplo de Emissão de Ingressos
Nota: Autoria Própria

5.3 INTERPRETAÇÃO DE INTERESSES DE NEGÓCIO PARA REQUISITOS DE NEGÓCIO

Esta Seção apresenta as definições referentes aos requisitos de negócio de um software S qualquer (Definição 14), bem como, os agrupamentos formados a partir desses requisitos (Definições 15 a 23). Cada requisito de negócio é apresentado juntamente com sua correspondente regra de negócio de um negócio N qualquer, de modo que os agrupamentos e os relacionamentos de dependência entre as regras de negócio apresentados nas Definições de 3 a 12 preservem os interesses de negócio já estabelecidos.

Definição 13: Seja $Q^S = \{r_1, r_2, \dots, r_k\}$ um conjunto não-vazio de requisitos de um software S qualquer.

Definição 14: Seja Q_n^S um subconjunto não-vazio de Q^S , em que cada um dos seus elementos é um requisito de negócio de S que se refere a um elemento do conjunto R^n (Definição 3).

As demais Definições de 15 até 23 correspondem às Definições de 4 até 12, respectivamente.

Definição 15: Seja $F_n^s = \{q_{a_1}, q_{a_2}, \dots, q_{a_z}\}$ um subconjunto não-vazio do conjunto Q_n^s , em que cada um dos seus elementos é uma funcionalidade-requisito de um software S que se refere a um elemento do conjunto F^n (Definição 4).

Definição 16: Seja $B_n^s = \{q_{b_1}, q_{b_2}, \dots, q_{b_j}\}$ um subconjunto não-vazio do conjunto Q_n^s , em que cada um dos seus elementos é uma regra-requisito de um software S que se refere a um elemento do conjunto B^n (Definição 5).

Definição 17: Seja $I_n^s = \{q_{i_1}, q_{i_2}, \dots, q_{i_p}\}$ um subconjunto de Q_n^s , em que cada um dos seus elementos é uma informação utilizada ou produzida por um software S para conduzir o negócio N . Cada um de seus elementos é uma informação-requisito que se refere a um elemento do conjunto I^n (Definição 6).

Exemplo. A correspondência entre os elementos dos conjuntos F_n^s , B_n^s e I_n^s e os elementos dos conjuntos F^n , B^n e I^n , respectivamente, para o exemplo apresentado no Capítulo 1 está descrita na Figura 23.

F_n^s	F^n
q_{a_1} : Emitir Ingresso	r_{a_2}
B_n^s	B^n
q_{b_1} : Se Nome do Evento informado for válido, então o Ingresso- Nome-do-Evento é igual ao Nome do Evento informado	r_{b_2}
q_{b_2} : Se a Data do Evento informada for válida, então o ingresso- Data-do-Evento é igual à Data do Evento informada	r_{b_3}
q_{b_3} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado	r_{b_4}
I_n^s	I^n
q_{i_1} : Ingresso-CPF	r_{i_2}
q_{i_2} : Ingresso-Data-do-Evento	r_{i_3}
q_{i_3} : Ingresso-Nome-do-Evento	r_{i_4}

Figura 23 – Requisitos de Negócio do exemplo de Emissão de Ingressos
Nota: Autoria Própria

5.3.1 Relações entre os tipos de regras de negócio em S

Esta Seção trata das definições 18, 19 e 20 que relacionam os elementos dos conjuntos F_n^s (Definição 15), B_n^s (Definição 16) e I_n^s (Definição 17) entre si, ora com o intuito de agrupar elementos a partir de outro elemento ora para estabelecer uma relação de dependência entre os elementos.

Definição 18: Seja $\Theta_n^s = \{\theta_{q_1}, \theta_{q_2}, \dots, \theta_{q_k}\}$ um subconjunto não-vazio de $B_n^s \times F_n^s$, em que cada um dos seus elementos participa de uma relação biunívoca com elementos do conjunto Θ^n (Definição 7).

Exemplo. A correspondência entre os elementos do conjunto Θ_n^s e os elementos do conjunto Θ^n para o exemplo apresentado no Capítulo 1 está descrita na Figura 24.

Θ_n^s	Θ^n	F_n^s
		q_{a_1} : Emitir Ingresso
		B_n^s
		q_{b_1} : Se Nome do Evento informado for válido, então o Ingresso-Nome-Evento é igual ao Nome do Evento informado
$q_{q_1} = (q_{b_1}, q_{a_1})$	θ_{r_2}	q_{b_2} : Se a Data do Evento informada for válida, então o ingresso-Data-Evento é igual à Data do Evento informada
$q_{q_2} = (q_{b_2}, r_{a_1})$	θ_{r_3}	q_{b_3} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado
$q_{q_3} = (q_{b_3}, r_{a_1})$	θ_{r_4}	

Figura 24 – Relação entre o conjunto Θ_n^s e o conjunto Θ^n

Nota: Autoria Própria

Definição 19: Seja $\hat{\Gamma}_n^s = \{\gamma_{q_1}, \gamma_{q_2}, \dots, \gamma_{q_k}\}$ um subconjunto não-vazio da relação $B_n^s \times I_n^s$, em que cada um dos seus elementos participa de uma relação biunívoca com elementos do conjunto $\hat{\Gamma}^n$ (Definição 8).

Exemplo. A correspondência entre os elementos de $\hat{\Gamma}_n^s$ e os elementos de $\hat{\Gamma}^n$ para o exemplo apresentado no Capítulo 1 está descrito na Figura 25.

		B_n^s
		q_{b_1} : Se Nome do Evento informado for válido, então o Ingresso-Nome-do-Evento é igual ao Nome do Evento informado
		q_{b_2} : Se a Data do Evento informada for válida, então o ingresso-Data-do-Evento é igual à Data do Evento informada
		q_{b_3} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado
		I_n^s
		q_{i_1} : Ingresso-CPF
		q_{i_2} : Ingresso-Data-do-Evento
		q_{i_3} : Ingresso-Nome-do-Evento
$\hat{\Gamma}_s^n$	$\hat{\Gamma}^n$	
$\hat{\gamma}_{q_1} = (q_{b_1}, q_{i_3})$	$\hat{\gamma}_{r_2}$	
$\hat{\gamma}_{q_2} = (q_{b_2}, q_{i_2})$	$\hat{\gamma}_{r_3}$	
$\hat{\gamma}_{q_3} = (q_{b_3}, q_{i_1})$	$\hat{\gamma}_{r_4}$	

Figura 25 – Relação entre o conjunto $\hat{\Gamma}_s^n$ e o conjunto $\hat{\Gamma}^n$

Nota: Autoria Própria

Definição 20: Seja $\check{\Gamma}_n^s = \{\gamma_{q_1}, \gamma_{q_2}, \dots, \gamma_{q_k}\}$ um subconjunto não-vazio da relação $B_n^s \times I_n^s$, em que cada um dos seus elementos participa de uma relação biunívoca com elementos do conjunto $\check{\Gamma}^n$ (Definição 9).

Exemplo. Para o exemplo apresentado no Capítulo 1 o conjunto $\check{\Gamma}_n^s$ é vazio.

5.3.2 Agrupamento Entre Funcionalidade, Regra e Informação em S

Esta Seção trata dos agrupamentos $\hat{\Psi}_n^s$ (Definição 21) e $\check{\Psi}_n^s$ (Definição 22), que relacionam entre si os elementos de Θ_n^s (Definição 18), $\hat{\Gamma}_n^s$ (Definição 19) e $\check{\Gamma}_n^s$ (Definição 20).

Definição 21: Seja $\hat{\Psi}_n^s$ um subconjunto da relação $\Theta_n^s \times \hat{\Gamma}_n^s$, em que cada um dos seus elementos participa de uma relação biunívoca com elementos do conjunto $\hat{\Psi}^n$ (Definição 10).

Exemplo. A correspondência entre os elementos de $\hat{\Psi}_n^s$ e os elementos de $\hat{\Psi}^n$ para o exemplo apresentado no Capítulo 1 está descrita na Figura 26.

$\hat{\Psi}_n^s$	$\hat{\Psi}^n$	F_n^s
$\hat{\psi}_{q_1} = \left(\overbrace{(q_{b_1}, q_{a_1})}^{\theta_{q_1}}, \overbrace{(q_{b_1}, q_{i_3})}^{\hat{\gamma}_{q_1}} \right)$	$\hat{\psi}_{r_2}$	q_{a_1} : Emitir Ingresso
$\hat{\psi}_{q_2} = \left(\overbrace{(q_{b_2}, q_{a_1})}^{\theta_{q_2}}, \overbrace{(q_{b_2}, q_{i_2})}^{\hat{\gamma}_{q_2}} \right)$	$\hat{\psi}_{r_3}$	B_n^s q_{b_1} : Se Nome do Evento informado for válido, então o Ingresso-Nome-do-Evento é igual ao Nome do Evento informado
$\hat{\psi}_{q_3} = \left(\overbrace{(q_{b_3}, q_{a_1})}^{\theta_{q_3}}, \overbrace{(q_{b_3}, r_{i_1})}^{\hat{\gamma}_{q_3}} \right)$	$\hat{\psi}_{r_4}$	q_{b_2} : Se a Data do Evento informada for válida, então o ingresso-Data-do-Evento é igual à Data do Evento informada q_{b_3} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado

Figura 26 – Relação entre o conjunto $\hat{\Psi}_n^s$ e o conjunto $\hat{\Psi}^n$

Nota: Autoria Própria

Definição 22: Seja $\check{\Psi}_n^s$ um subconjunto da relação $\theta_n^s \times \check{I}_n^s$, em que cada um dos seus elementos participa de uma relação biunívoca com elementos do conjunto $\check{\Psi}^n$ (Definição 11).

Exemplo. Para o exemplo apresentado no Capítulo 1 o conjunto $\check{\Psi}_n^s$ é vazio.

5.3.3 A Relação Interesse de Negócio em S

Esta Seção trata da relação Φ_n^s (Definição 23) que preserva em S os mesmo interesses de negócio definidos em Φ^n (Definição 12).

Definição 23: Seja Φ_n^s um subconjunto da relação $\hat{\Psi}_n^s \times \check{\Psi}_n^s$, em que cada um dos seus elementos participa de uma relação biunívoca com elementos do conjunto Φ^n (Definição 12).

Exemplo. Para o exemplo apresentado no Capítulo 1 o conjunto Φ_n^s é vazio.

A Figura 27 representa graficamente um resumo da aplicação dessa etapa da técnica FIR para o exemplo apresentado no Capítulo 1.

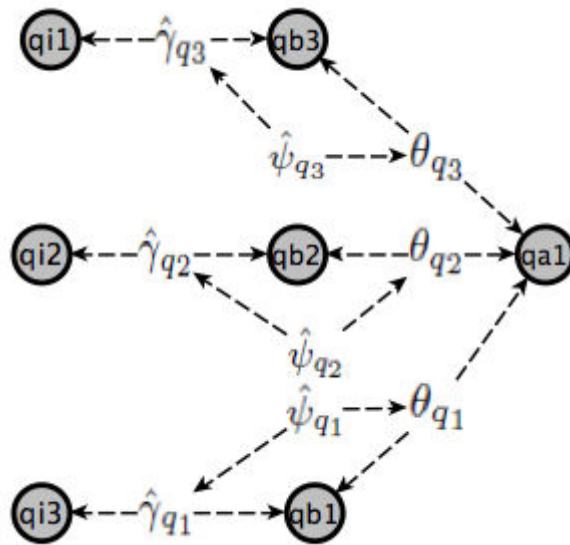


Figura 27 – Visão Gráfica de Interesses de Negócio
Nota: Autoria Própria

5.4 DELIMITAÇÃO DAS FRONTEIRAS

Esta Seção corresponde a um dos passos da técnica FIR, responsável por estabelecer as fronteiras entre um software S qualquer e um negócio N qualquer para o qual S presta serviços. Esse passo torna evidentes os relacionamentos entre os requisitos de negócio de S e as regras de negócio de N, de modo que os interesses de negócio estabelecidos sejam preservados.

5.4.1 Negócio fora de S

Para que se possa estabelecer as fronteiras entre S e N, antes é necessário definir com clareza quais são as regras de negócio de N que estão fora de S e que ao mesmo tempo se relacionam com os requisitos de negócio de S. Para diferenciar os conjuntos compostos por regras de negócio que estão fora de S, utiliza-se símbolo s que significa não se tratar do software s , por exemplo: R_s^n para o conjunto das regras de negócio R^n que não são requisitos de negócio Q^s .

As definições R_s^n (Definição 24), F_s^n (Definição 25), B_s^n (Definição 26) e I_s^n (Definição 27) tratam das regras de negócio que estão fora de um software S qualquer, enquanto as Definições $\vec{\Gamma}_n^s$ (Definição 34) e $\vec{\Gamma}_s^n$ (Definição 35) tratam do estabelecimento da fronteira propriamente dita entre S e um negócio N qualquer.

Definição 24: Seja R_S^n o conjunto das regras de negócio de um negócio N qualquer, que não são requisitos de negócio de um software S qualquer.

Definição 25: Seja F_S^n um subconjunto não-vazio do conjunto $R_S^n \cap F^n$, em que cada um dos seus elementos é uma regra de negócio do tipo Funcionalidade de N , mas que não é funcionalidade-requisito de S .

Definição 26: Seja B_S^n um subconjunto não-vazio do conjunto $R_S^n \cap B^n$, em que cada um dos seus elementos é uma regra de negócio do tipo Regra de N , mas que não são regras-requisito de S .

Definição 27: Seja I_S^n um subconjunto não-vazio do conjunto $R_S^n \cap I^n$, em que cada um dos seus elementos é uma regra de negócio do tipo Informação de N , mas que não são informações-requisito de S .

Exemplo. A Figura 28 apresenta um resumo dos conjuntos F_S^n , B_S^n e I_S^n para o exemplo apresentado no Capítulo 1.

F_S^n	r_{a1} : Cadastrar Torcedor. <hr/> r_{a3} : Liberar Acesso.
B_S^n	r_{b1} : Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado <hr/> r_{b5} : O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado <hr/> r_{b6} : O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado <hr/> r_{b7} : O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado
I_S^n	r_{i1} : Torcedor-CPF

Figura 28 – Os Conjuntos F_S^n , B_S^n e I_S^n para o exemplo apresentado no Capítulo 1
Nota: Autoria Própria

5.4.2 Relações fora de S

As definições Θ_S^n (Definição 28), $\hat{\Gamma}_S^n$ (Definição 29), $\check{\Gamma}_S^n$ (Definição 30), que relacionam apenas as regras de negócio que não são requisitos de S , estão descritas a seguir.

Definição 28: Seja $\Theta_{\mathcal{S}}^n$ um subconjunto não-vazio de $B_{\mathcal{S}}^n \times F_{\mathcal{S}}^n$ (Definições 26 e 25, respectivamente), tal que $\Theta_{\mathcal{S}}^n$ também seja subconjunto de Θ^n (Definição 7).

Exemplo. O conjunto $\Theta_{\mathcal{S}}^n$ para o exemplo apresentado no Capítulo 1 está descrito na Figura 29.

$\Theta_{\mathcal{S}}^n$ <hr/> $\theta_{r_1} = (r_{b_1}, r_{a_1})$ <hr/> $\theta_{r_5} = (r_{b_5}, r_{a_3})$ <hr/> $\theta_{r_6} = (r_{b_6}, r_{a_3})$ <hr/> $\theta_{r_7} = (r_{b_7}, r_{a_3})$	$F_{\mathcal{S}}^n$
	<i>r_{a1}: Cadastrar Torcedor.</i>
	<i>r_{a3}: Liberar Acesso.</i>
	$B_{\mathcal{S}}^n$
	<i>r_{b1}: Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado.</i>
<i>r_{b5}: O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado.</i>	
<i>r_{b6}: O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado.</i>	
<i>r_{b7}: O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado.</i>	

Figura 29 - O conjunto $\Theta_{\mathcal{S}}^n$ para o exemplo apresentado no Capítulo 1

Nota: Autoria Própria

Definição 29: Seja $\hat{\Gamma}_{\mathcal{S}}^n$ um subconjunto da relação $B_{\mathcal{S}}^n \times I_{\mathcal{S}}^n$, tal que $\hat{\Gamma}_{\mathcal{S}}^n$ também seja subconjunto de $\hat{\Gamma}^n$ (Definição 8).

Exemplo. O conjunto $\hat{\Gamma}_{\mathcal{S}}^n$ para o exemplo apresentado no Capítulo 1 está descrito na Figura 30.

$\hat{\Gamma}_{\mathcal{S}}^n$ <hr/> $\hat{\gamma}_{r_1} = (r_{b_1}, r_{i_1})$	$B_{\mathcal{S}}^n$
	<i>r_{b1}: Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado</i>
	$I_{\mathcal{S}}^n$
	<i>r_{i1}: Torcedor-CPF</i>

Figura 30 - O conjunto $\hat{\Gamma}_{\mathcal{S}}^n$ para o exemplo apresentado no Capítulo 1

Nota: Autoria Própria

Definição 30: Seja $\check{\Gamma}_{\mathcal{S}}^n$ um subconjunto da relação $B_{\mathcal{S}}^n \times I_{\mathcal{S}}^n$, tal que $\check{\Gamma}_{\mathcal{S}}^n$ também seja subconjunto de $\check{\Gamma}^n$ (Definição 9).

Exemplo. Para o exemplo apresentado no Capítulo 1 o conjunto $\check{\Gamma}_{\mathcal{S}}^n$ é vazio.

5.4.3 Agrupamento entre Funcionalidade, Informação e Regra fora de S

Os agrupamentos $\hat{\Psi}_{\mathcal{S}}^n$ e $\check{\Psi}_{\mathcal{S}}^n$ que compreendem as regras de negócio que estão fora de S estão definidas a seguir.

Definição 31: Seja $\hat{\Psi}_{\mathcal{S}}^n$ um subconjunto da relação $\Theta_{\mathcal{S}}^n \times \hat{\Gamma}_{\mathcal{S}}^n$ (Definições 28 e 29 respectivamente), tal que $\hat{\Psi}_{\mathcal{S}}^n$ também seja subconjunto de $\hat{\Psi}^n$ (Definição 10).

Exemplo. O conjunto $\hat{\Psi}_{\mathcal{S}}^n$ para o exemplo apresentado no Capítulo 1 está descritos na Figura 31.

$\hat{\Psi}_{\mathcal{S}}^n$ <hr/> $\underbrace{\theta_{r_1} \quad \hat{\gamma}_{r_1}}_{\hat{\psi}_{r_1}}$ $\underbrace{((r_{b_1}, r_{a_1}), (r_{b_1}, r_{i_1}))}_{\hat{\psi}_{r_1}}$ <hr/>	$F_{\mathcal{S}}^n$ <hr/> r_{a_1} : Cadastrar Torcedor <hr/> $B_{\mathcal{S}}^n$ <hr/> r_{b_1} : Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado <hr/> $I_{\mathcal{S}}^n$ <hr/> r_{i_1} : Torcedor-CPF
---	--

Figura 31 - O conjunto $\hat{\Psi}_{\mathcal{S}}^n$ para o exemplo apresentado no Capítulo 1
Nota: Autoria Própria

Definição 32: Seja $\check{\Psi}_{\mathcal{S}}^n$ um subconjunto da relação $\Theta_{\mathcal{S}}^n \times \check{\Gamma}_{\mathcal{S}}^n$ (Definições 28 e 30 respectivamente), tal que $\check{\Psi}_{\mathcal{S}}^n$ também seja subconjunto de $\check{\Psi}^n$ (Definição 11).

Exemplo. Para o exemplo apresentado no Capítulo 1 o conjunto $\check{\Psi}_{\mathcal{S}}^n$ é vazio.

5.4.4 A Relação Interesse de Negócio fora de S

A definição a seguir considera exclusivamente os interesses de negócio que estão fora de um software S qualquer.

Definição 33: Seja $\Phi_{\mathcal{S}}^n$ um subconjunto da relação $\hat{\Psi}_{\mathcal{S}}^n \times \check{\Psi}_{\mathcal{S}}^n$ (Definições 31 e 32 respectivamente), tal que $\Phi_{\mathcal{S}}^n$ também seja subconjunto de Φ^n (Definição 12).

Exemplo. Para o exemplo apresentado no Capítulo 1 o conjunto $\Phi_{\mathcal{S}}^n$ é vazio.

5.4.5 Relacionamentos na Fronteira

Uma vez que as regras de negócio que estão fora de S estão definidas é possível estabelecer as fronteiras, o que é feito nesta Seção. Duas são as fronteiras entre S e N. A fronteira de consumo de S definida pelo conjunto $\vec{\Gamma}_{\mathcal{S}}^n$ (Definição 35) e a fronteira de consumo fora de S definida pelo conjunto $\vec{\Gamma}_n^s$ (Definição 34).

Definição 34: Seja $\vec{\Gamma}_n^s$ um subconjunto não-vazio da relação $B_{\mathcal{S}}^n \times I_n^s$, em que cada um dos seus elementos participa de uma relação biunívoca com elementos do conjunto $\check{\Gamma}^n$ (Definição 9). Se $(r_b, q_i) \in \vec{\Gamma}_n^s$, então a regra r_b que está fora do software S , consome a informação-requisito q_i que é produzida em S .

Exemplo. O conjunto $\vec{\Gamma}_n^s$ e a correspondência com $\check{\Gamma}^n$ para o exemplo apresentado no Capítulo 1, estão descritos na Figura 32.

		$B_{\mathcal{S}}^n$
		r_{b_5} : O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado
		r_{b_6} : O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado
		r_{b_7} : O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado
		I_n^s
		q_{i_1} : Ingresso-CPF
		q_{i_2} : Ingresso-Data-do-Evento
		q_{i_3} : Ingresso-Nome-do-Evento
$\vec{\Gamma}_n^s$	$\check{\Gamma}^n$	
$\vec{\gamma}_{q_1} = (r_{b_5}, q_{i_3})$	$\check{\gamma}_{r_6}$	
$\vec{\gamma}_{q_2} = (r_{b_6}, q_{i_1})$	$\check{\gamma}_{r_7}$	
$\vec{\gamma}_{q_3} = (r_{b_7}, q_{i_2})$	$\check{\gamma}_{r_8}$	

Figura 32 - O conjunto $\vec{\Gamma}_n^s$ para o exemplo apresentado no Capítulo 1

Nota: Autoria Própria

Definição 35: Seja $\vec{\Gamma}_{\mathcal{S}}^n$ um subconjunto não-vazio da relação $B_n^s \times I_{\mathcal{S}}^n$, em que cada um dos seus elementos participa de uma relação biunívoca com elementos do conjunto $\check{\Gamma}^n$ (Definição 9). Se $(q_b, r_i) \in \vec{\Gamma}_{\mathcal{S}}^n$, então a regra q_b que é um requisito de negócio do software S , consome a informação r_i que é produzida fora de S .

Exemplo. O conjunto $\vec{\Gamma}_{\mathcal{S}}^n$ e a correspondência com $\check{\Gamma}^n$ para o exemplo apresentado no Capítulo 1, estão descritos na Figura 33.

$\vec{\Gamma}_{s'}^n$	$\check{\Gamma}^n$	B_n^s
$\vec{\gamma}_{q_1} = (q_{b_3}, r_{i_1})$	$\check{\gamma}_{r_5}$	<i>q_{b₃}</i> : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado
		$I_{s'}^n$
		<i>r_{i₁}</i> : Torcedor-CPF

Figura 33 - O conjunto $\vec{\Gamma}_{s'}^n$ para o exemplo apresentado no Capítulo 1

Nota: Autoria Própria

5.4.6 Agrupamento entre Funcionalidade, Informação e Regra na fronteira de S

Os demais agrupamentos que contém elementos de $\vec{\Gamma}_n^s$, tais como $\vec{\Psi}_n^s$ e $\vec{\Phi}_n^s$. Os agrupamentos que contém elementos de $\vec{\Gamma}_{s'}^n$, tais como $\vec{\Psi}_{s'}^n$ e $\vec{\Phi}_{s'}^n$ estão descritos a seguir.

Definição 36: Seja $\vec{\Psi}_n^s$ um subconjunto não-vazio da relação $\Theta_{s'}^n \times \vec{\Gamma}_n^s$ (Definições 28 e 34 respectivamente), em que cada um dos seus elementos participa de uma relação biunívoca com elementos do conjunto $\check{\Psi}^n$ (Definição 11).

Exemplo. O conjunto $\vec{\Psi}_n^s$ referente a $\check{\Psi}^n$ para o exemplo apresentado no Capítulo 1 estão descritos na Figura 34.

$\vec{\Psi}_n^s$	Ψ^n	I_n^s
$\underbrace{\underbrace{\theta_{r_5}}_{((r_{b_5}, r_{a_3}), (r_{b_5}, q_{i_3}))}}_{\vec{\psi}_{q_1}} \quad \underbrace{\vec{\gamma}_{q_1}}$	ψ_{r_2}	q_{i_1} : Ingresso-CPF q_{i_2} : Ingresso-Data-do-Evento q_{i_3} : Ingresso-Nome-do-Evento
$\underbrace{\underbrace{\theta_{r_7}}_{((r_{b_7}, r_{a_3}), (r_{b_7}, q_{i_2}))}}_{\vec{\psi}_{q_3}} \quad \underbrace{\vec{\gamma}_{q_3}}$	ψ_{r_4}	$F_{s'}^n$ r_{a_3} : Liberar Acesso $B_{s'}^n$ r_{b_5} : O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado
$\underbrace{\underbrace{\theta_{r_6}}_{((r_{b_6}, r_{a_3}), (r_{b_6}, q_{i_1}))}}_{\vec{\psi}_{q_2}} \quad \underbrace{\vec{\gamma}_{q_2}}$	ψ_{r_3}	r_{b_6} : O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado r_{b_7} : O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado

Figura 34 – O conjunto $\vec{\Psi}_n^s$ para o exemplo apresentado no Capítulo 1

Nota: Autoria Própria

Definição 37: Seja $\vec{\Psi}_{s'}^n$ um subconjunto da relação $\theta_n^s \times \vec{\Gamma}_{s'}^n$ (Definições 18 e 35 respectivamente), em que cada um dos seus elementos participa de uma relação biunívoca com elementos do conjunto Ψ^n (Definição 11).

Exemplo. O conjunto $\vec{\Psi}_{s'}^n$ referente a Ψ^n para o exemplo apresentado no Capítulo 1 estão descritos na Figura 35.

$\vec{\Psi}_{s'}^n$	Ψ^n	F_n^s
$\underbrace{\underbrace{\theta_{r_4}}_{((q_{b_3}, q_{a_1}), (q_{b_3}, r_{i_1}))}}_{\vec{\psi}_{q_1}} \quad \underbrace{\vec{\gamma}_{q_1}}$	ψ_{r_1}	q_{a_1} : Emitir Ingresso B_n^s q_{b_3} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado $I_{s'}^n$ r_{i_1} : Torcedor-CPF

Figura 35 - O conjunto $\vec{\Psi}_{s'}^n$ para o exemplo apresentado no Capítulo 1

Nota: Autoria Própria

5.4.7 Interesses de Negócio na fronteira de S

Definição 38: Seja $\vec{\Phi}_{\mathcal{F}}^n$ um subconjunto da relação $\widehat{\Psi}_n^s \times \vec{\Psi}_n^n$ (Definições 21 e 37 respectivamente), em que cada um dos seus elementos participa de uma relação biunívoca com elementos do conjunto Φ^n (Definição 12).

Exemplo. O conjunto $\vec{\Phi}_{\mathcal{F}}^n$ referente a Φ^n para o exemplo apresentado no Capítulo 1 estão descritos na Figura 36.

$\vec{\Phi}_{\mathcal{F}}^n$	Φ^n
$\vec{\phi}_{q_1} = \left(\underbrace{\left(\underbrace{(r_{b_1}, r_{a_1})}_{\theta_{r_1}}, \underbrace{(r_{b_1}, r_{i_1})}_{\hat{\gamma}_{r_1}} \right)}_{\hat{\psi}_{r_1}}, \underbrace{\left(\underbrace{(q_{b_3}, q_{a_1})}_{\theta_{r_4}}, \underbrace{(q_{b_3}, r_{i_1})}_{\vec{\gamma}_{q_1}} \right)}_{\vec{\psi}_{q_1}} \right)$	ϕ_{r_1}
	F_n^s <hr/> q_{a_1} : Emitir Ingresso <hr/> B_n^s <hr/> q_{b_3} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado <hr/> $F_{\mathcal{F}}^n$ <hr/> r_{a_1} : Cadastrar Torcedor <hr/> $B_{\mathcal{F}}^n$ <hr/> r_{b_1} : Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado <hr/> $I_{\mathcal{F}}^n$ <hr/> r_{i_1} : Torcedor-CPF

Figura 36 - O conjunto $\vec{\Phi}_{\mathcal{F}}^n$ para o exemplo apresentado no Capítulo 1

Nota: Autoria Própria

Definição 39: Seja $\vec{\Phi}_n^s$ um subconjunto não-vazio da relação $\widehat{\Psi}_n^n \times \vec{\Psi}_n^s$ (Definições 31 e 36 respectivamente), em que cada um dos seus elementos participa de uma relação biunívoca com elementos do conjunto Φ^n (Definição 12).

Exemplo. O conjunto $\vec{\Phi}_n^s$ referente a Φ^n para o exemplo apresentado no Capítulo 1 estão descritos na Figura 37.

$\vec{\Phi}_n^s$		Φ^n
$\vec{\phi}_{q_1} = \left(\underbrace{\left(\underbrace{\theta_{q_1}, \hat{\gamma}_{q_1}}_{\hat{\psi}_{q_1}}, \underbrace{\theta_{r_5}, \vec{\gamma}_{q_1}}_{\vec{\psi}_{q_1}} \right)}_{\left((q_{b_1}, q_{a_1}), (q_{b_1}, q_{i_3}), (r_{b_5}, r_{a_3}), (r_{b_5}, q_{i_3}) \right)} \right)$	ϕ_{r_2}	F_n^s q_{a_1} : Emitir Ingresso <hr/> B_n^s q_{b_1} : Se Nome do Evento informado for válido, então o Ingresso-Nome-do-Evento é igual ao Nome do Evento informado <hr/> q_{b_2} : Se a Data do Evento informada for válida, então o ingresso-Data-do-Evento é igual à Data do Evento informada <hr/> q_{b_3} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado <hr/> I_n^s q_{i_1} : Ingresso-CPF q_{i_2} : Ingresso-Data-do-Evento q_{i_3} : Ingresso-Nome-do-Evento <hr/> F_n^s r_{a_3} : Liberar Acesso <hr/> B_n^s r_{b_5} : O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado <hr/> r_{b_6} : O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado <hr/> r_{b_7} : O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado
$\vec{\phi}_{q_2} = \left(\underbrace{\left(\underbrace{\theta_{q_2}, \hat{\gamma}_{q_2}}_{\hat{\psi}_{q_2}}, \underbrace{\theta_{r_7}, \vec{\gamma}_{q_3}}_{\vec{\psi}_{q_3}} \right)}_{\left((q_{b_2}, q_{a_1}), (q_{b_2}, q_{i_2}), (r_{b_7}, r_{a_3}), (r_{b_7}, q_{i_2}) \right)} \right)$	ϕ_{r_3}	
$\vec{\phi}_{q_3} = \left(\underbrace{\left(\underbrace{\theta_{q_3}, \hat{\gamma}_{q_3}}_{\hat{\psi}_{q_3}}, \underbrace{\theta_{r_6}, \vec{\gamma}_{q_2}}_{\vec{\psi}_{q_2}} \right)}_{\left((q_{b_3}, q_{a_1}), (q_{b_3}, q_{i_1}), (r_{b_6}, r_{a_3}), (r_{b_6}, q_{i_1}) \right)} \right)$	ϕ_{r_4}	

Figura 37 - O conjunto $\vec{\Phi}_n^s$ para o exemplo apresentado no Capítulo 1

Nota: Autoria Própria

A Figura 38 mostra graficamente o resultado da aplicação desta etapa da técnica FIR.

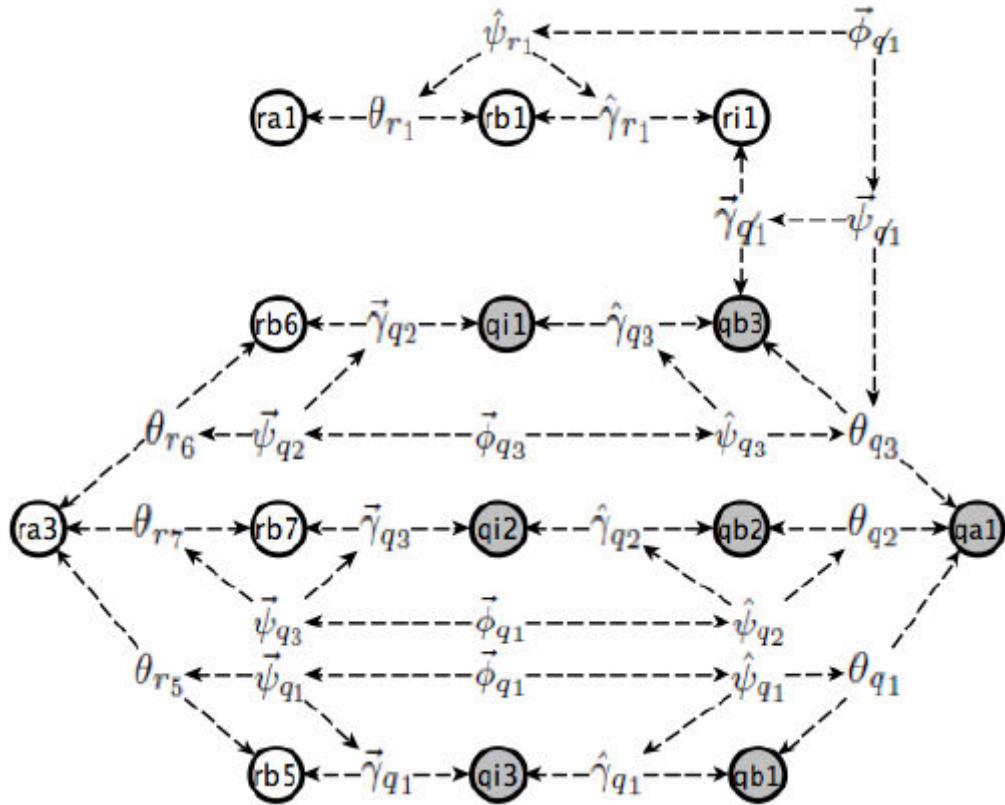


Figura 38 - Visão gráfica do Exemplo apresentado no Capítulo 1
Nota: Autoria Própria

6 ANÁLISE DE IMPACTO COM FIR

Neste Capítulo, o FIR bem-formado é apresentado como um grafo denominado FIR^g , para que a atividade de descoberta de impactos ocorra a partir de operações sobre tal grafo. A Definição 40 a seguir trata dessa formalização.

Definição 40: Seja FIR^g um grafo colorido formado pela sêxtupla $(V_n^s, V_s^n, \vec{P}_n^s, \vec{P}_s^n, P_n^s, P_s^n)$, tal que:

- a) V_n^s é o conjunto não-vazio de vértices coloridos de FIR^g , tal que, qualquer que seja $u \in V_n^s$, as afirmações a seguir são verdadeiras:
 - u tem cor c_1 , se existir $a \in F_n^s$ (Definição 15), tal que u tem relação com a ;
 - u tem cor c_2 , se existir $b \in B_n^s$ (Definição 16), tal que u tem relação com b ;
 - u tem cor c_3 , se existir $i \in I_n^s$ (Definição 17) tal que u tem relação com i ;
- b) V_s^n é o conjunto não-vazio de vértices coloridos de FIR^g , tal que, qualquer que seja $u \in V_s^n$, as afirmações a seguir são verdadeiras:
 - u tem cor c_4 , se existir $a \in F_s^n$ (Definição 25), tal que u tem relação com a ;
 - u tem cor c_5 , se existir $b \in B_s^n$ (Definição 26), tal que u tem relação com b ;
 - u tem cor c_6 , se existir $i \in I_s^n$ (Definição 27), tal que u tem relação com i ;
- c) \vec{P}_n^s é o conjunto não-vazio de todos os caminhos formados por quintuplas $(u_k, u_{k+1}, u_{k+2}, u_{k+3}, u_{k+4})$, tal que u_p tem cor c_1 , u_{k+1} tem cor c_2 , u_{k+2} tem cor c_3 , u_{k+3} tem cor c_5 e u_{k+4} tem cor c_4 . Para todo $p \in \vec{P}_n^s$, existe $\varphi \in \vec{\Phi}_n^s$ (Definição 39), tal que p tem relação com φ ;
- d) \vec{P}_s^n é o conjunto de todos os caminhos formados por quintuplas $(u_k, u_{k+1}, u_{k+2}, u_{k+3}, u_{k+4})$, tal que u_p tem cor c_4 , u_{k+1} tem cor c_5 , u_{k+2} tem cor

- c_6 , u_{k+3} tem cor c_2 e u_{k+4} tem cor c_1 , para todo $p \in \vec{P}_s^n$, existe $\varphi \in \vec{\Phi}_s^n$ (Definição 38), tal que p tem relação com φ ;
- e) P_n^s é o conjunto de todos os caminhos formado por quintuplas $(u_k, u_{k+1}, u_{k+2}, u_{k+3}, u_{k+4})$, tal que u_p tem cor c_1 , u_{k+1} tem cor c_2 , u_{k+2} tem cor c_3 , u_{k+3} tem cor c_2 e u_{k+4} tem cor c_1 , para todo $p \in P_n^s$, existe $\varphi \in \Phi_n^s$ (Definição 23), tal que p tem relação com φ ;
- f) P_s^n é um conjunto de todos os caminhos formado por quintuplas $(u_k, u_{k+1}, u_{k+2}, u_{k+3}, u_{k+4})$, tal que u_p tem cor c_4 , u_{k+1} tem cor c_5 , u_{k+2} tem cor c_6 , u_{k+3} tem cor c_5 e u_{k+4} tem cor c_4 , para todo $p \in P_s^n$, existe $\varphi \in \Phi_s^n$ (Definição 33), tal que p tem relação com φ ;

Um grafo FIR⁹ para o exemplo apresentado no Capítulo 1 está representado nas Figuras 39 e 40. Para esse exemplo as cores escolhidas foram as seguintes:

- a) A cor c_1 corresponde ao preto;
- b) A cor c_2 corresponde ao cinza a 80%;
- c) A cor c_3 corresponde ao cinza a 60%;
- d) A cor c_4 corresponde ao branco;
- e) A cor c_5 corresponde ao cinza a 20%;
- f) A cor c_6 corresponde ao cinza a 40%.

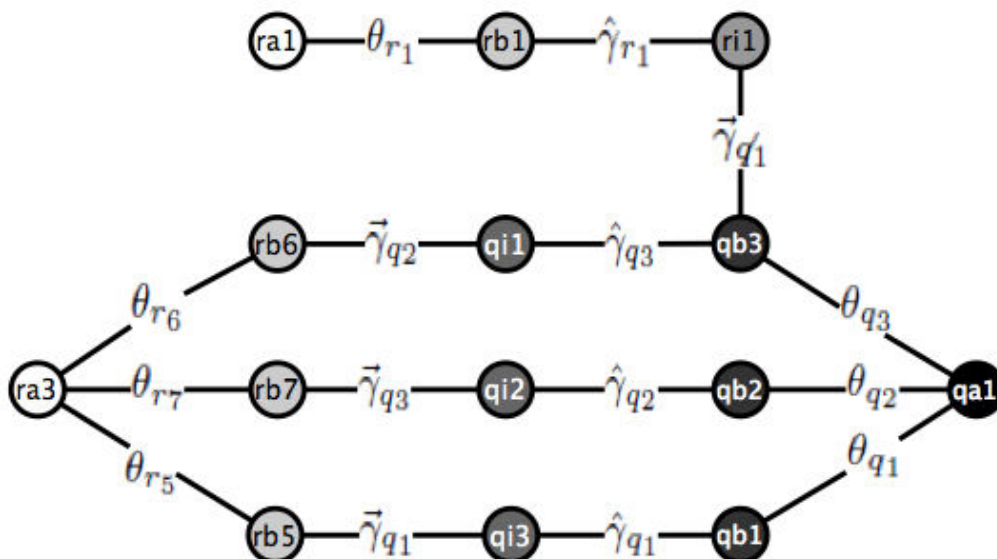


Figura 39 - O grafo FIR⁹ para o exemplo apresentado no Capítulo 1
Nota: Autoria Própria

F_n^s
q_{a_1} : Emitir Ingresso
B_n^s
q_{b_1} : Se Nome do Evento informado for válido, então o Ingresso-Nome-do-Evento é igual ao Nome do Evento informado
q_{b_2} : Se a Data do Evento informada for válida, então o ingresso-Data-do-Evento é igual à Data do Evento informada
q_{b_3} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado
I_n^s
q_{i_1} : Ingresso-CPF
q_{i_2} : Ingresso-Data-do-Evento
q_{i_3} : Ingresso-Nome-do-Evento
I_s^n
r_{i_1} : Torcedor-CPF
B_s^n
r_{b_1} : Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado
r_{b_5} : O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado
r_{b_6} : O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado
r_{b_7} : O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado
F_s^n
r_{a_1} : Cadastrar Torcedor
r_{a_3} : Liberar Acesso

Figura 40 – Detalhamento do grafo FIR^g para o exemplo apresentado no Capítulo 1

Nota: Autoria Própria

A Seção 6.1 trata da definição de um grafo FIR^g completo. A Seção 6.2 trata da atividade de descoberta de impactos a partir de um grafo FIR^g completo. Na Seção 6.3, discutimos a técnica FIR comparando-a com as demais técnicas estudadas, tendo como base o exemplo apresentado no Capítulo 1.

6.1 O GRAFO FIR^g COMPLETO

A Definição 41 trata do estabelecimento de um grafo FIR^g completo.

Definição 41: Seja um grafo $FIR^g = (V_n^s, V_s^n, \vec{P}_n^s, \vec{P}_s^n, P_n^s, P_s^n)$ qualquer, FIR^g é dito completo, se para todo componente c de FIR^g existir um caminho p , tal que $p \in \vec{P}_n^s$.

As Figuras 39, 40 e 41, 42 são exemplos de grafos FIR^g completos para o exemplo apresentado no Capítulo 1. Por outro lado, as Figuras 43 e 44 não representam um grafo FIR^g completo, uma vez que o conjunto \vec{P}_n^s é vazio.

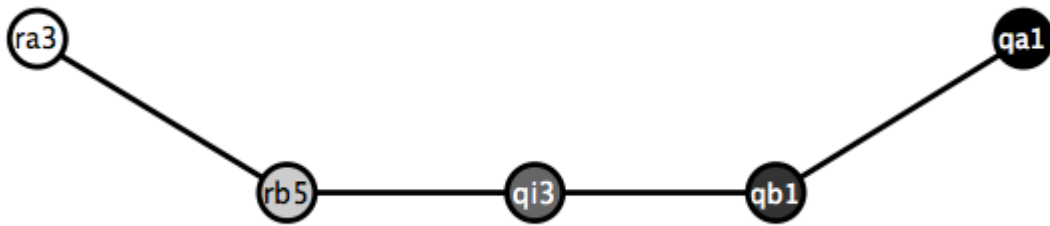


Figura 41 - Um Grafo FIR⁹ mínimo para o exemplo apresentado no Capítulo 1
Nota: Autoria Própria

F_n^s
q_{a_1} : Emitir Ingresso
B_n^s
q_{b_1} : Se Nome do Evento informado for válido, então o Ingresso-Nome-do-Evento é igual ao Nome do Evento informado
I_n^s
q_{i_3} : Ingresso-Nome-do-Evento
B_s^n
r_{b_5} : O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado
F_s^n
r_{a_3} : Liberar Acesso

Figura 42 – Detalhamento de um grafo FIR⁹ mínimo para o exemplo apresentado no Capítulo 1

Nota: Autoria Própria

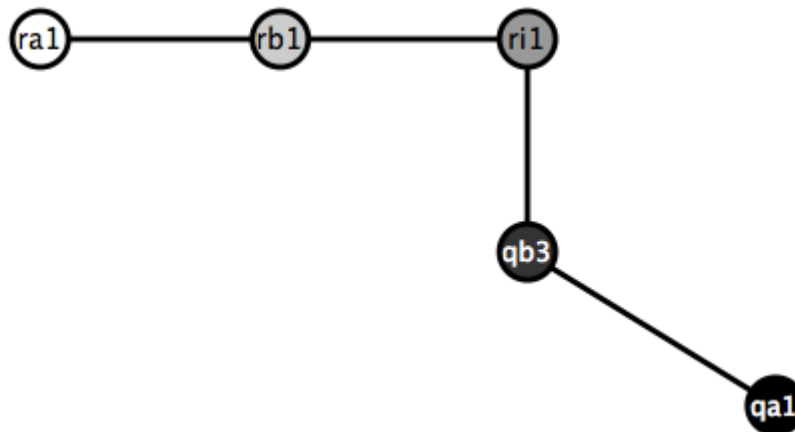


Figura 43 - Um grafo que não é um grafo FIR⁹ para o exemplo do Capítulo 1
Nota: Autoria Própria

F_n^s
q_{a_1} : Emitir Ingresso
B_n^s
q_{b_3} : Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado
$I_{s'}^n$
r_{i_1} : Torcedor-CPF
$B_{s'}^n$
r_{b_1} : Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado
$F_{s'}^n$
r_{a_1} : Cadastrar Torcedor

Figura 44 – Um grafo detalhado que não é um grafo FIR⁹ para o exemplo do Capítulo 1
Nota: Autoria Própria

6.2 DESCOBERTA DE IMPACTOS EM UM GRAFO FIR⁹

Uma vez que se disponha de um grafo FIR⁹ completo, a atividade de descoberta de impactos decorrentes de mudanças se resume a um procedimento de se caminhar no grafo. Tomando-se como base as Figuras 39 e 40, os dois cenários de mudanças descritos a seguir têm fácil consideração de impactos.

Cenário 1: Uma mudança é solicitada para a regra de negócio r_{b_1} : “Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado”. A atividade de descoberta de impactos se inicia a partir do corresponde vértice r_{b_1} e descobre impactos nos vértices q_{b_3} referente ao requisito de negócio “Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual a Torcedor-CPF informado”, e no vértice r_{b_6} referente à regra de negócio “O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado”.

Cenário 2: Uma mudança é solicitada para o requisito de negócio q_{b_3} : “Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual a Torcedor-CPF informado”. A atividade de descoberta de impactos se inicia a partir do correspondente vértice q_{b_3} e descobre impacto no vértice r_{b_6} referente à regra de negócio “O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado”.

A partir do FIR⁹, tornou-se possível que mudanças solicitadas em regras de negócio possam ter impactos descobertos em requisitos de negócio no software e fora do software (cenário 1), bem como, mudanças solicitadas em requisitos de negócio tenham impactos descobertos em regras de negócio fora do software.

6.3 DISCUSSÃO

Nesta Seção, discutimos a técnica FIR comparando-a com as demais técnicas estudadas, tendo como base o exemplo apresentado no Capítulo 1. A Figura 45 apresenta as limitações das técnicas estudadas para rastrear regras de negócio, de modo que os interesses de negócio sejam preservados independentemente de haver ou não software. Enquanto isto, a Figura 46 que representa um grafo FIR⁹ completo, corrige essas limitações. A seguir é realizado um contraponto entre ambas as Figuras.

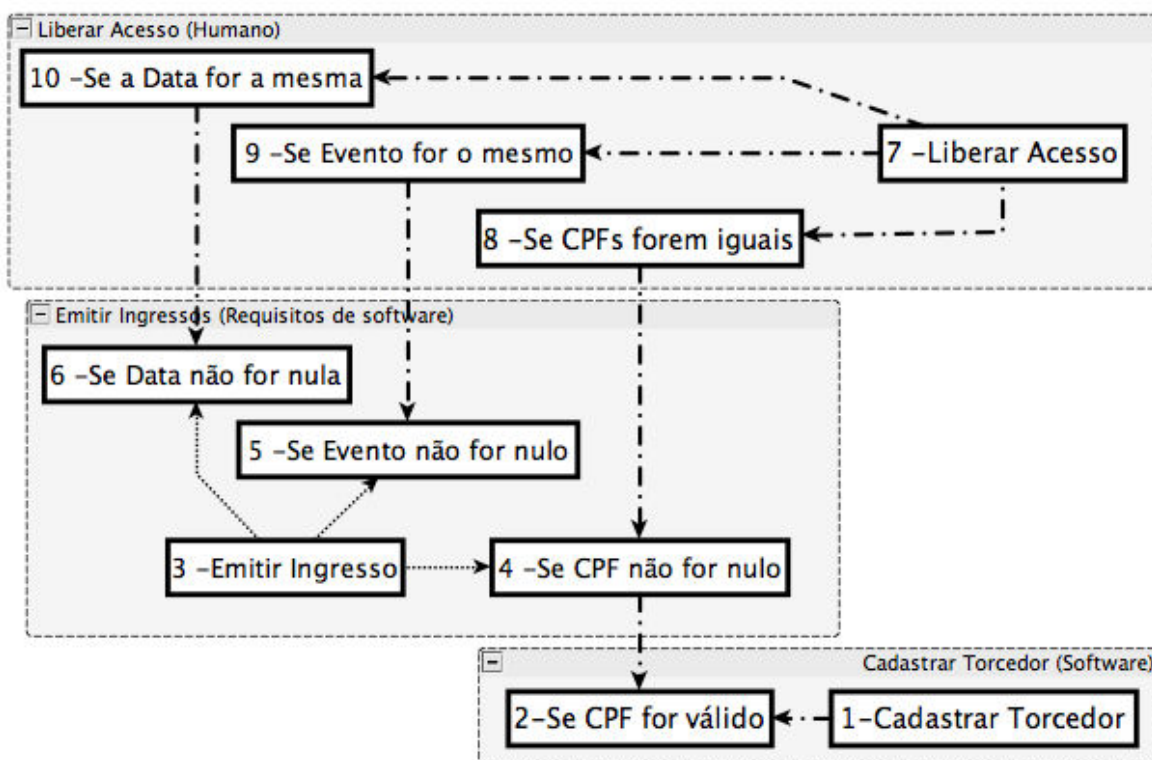


Figura 45 - Rastreamento do exemplo do Capítulo 1 pelas técnicas estudadas
Nota: Autoria Própria

Presença de elos de dependência. As Técnicas estudadas, quando dispõem de elos de dependência, esses são opcionais. Na Figura 45 esse fato está indicado

pela linha pontilhada entre os requisitos de negócio 3 e 4, 3 e 5 e 3 e 6. Por sua vez, a técnica FIR obriga a presença de elos de dependência entre as regras, o que está indicado na Figura 46 através de linhas sólidas entre os requisitos de negócio 3 e 4, 3 e 5 e 3 e 6.

Semântica apropriada. As técnicas estudadas não dispõem de uma semântica apropriada para lidar com dependência entre requisitos de negócio. Uma vez que o elo de dependência esteja estabelecido, não se sabe posteriormente o que provocou a dependência. Na Figura 45 este fato está indicado por linhas pontilhadas entre os requisitos de negócio 3 e 4, 3 e 5 e 3 e 6. Por sua vez, a técnica FIR dispõe de uma semântica específica para relacionar requisitos de negócio entre si através das Definições 18, 19 e 20. Na Figura 46 os relacionamentos entre os requisitos de negócio 3 e 4, 3 e 5 e 3 e 6 indicam a relação exposta através da Definição 18, enquanto o relacionamento entre o requisitos 4 e a informação Ingresso-CPF, 5 e a informação Ingresso-nome-evento e o requisito 6 e a informação Ingresso-data-evento indicam a relação exposta através da Definição 19.

Elos externos ao software. As técnicas estudadas não dispõem de elos que relacionem requisitos de negócio a regras de negócio que estejam externas ao software. Na Figura 45, o elo entre a regras de negócio 1 e 2; o elo entre a regra de negócio 2 e o requisito de negócio 4; o elo entre o requisito de negócio 4 e a regra de negócio 8; o requisito de negócio 5 e a regra de negócio 9; e o elo entre o requisito de negócio 6 e a regra de negócio 10, não estão disponíveis nas técnicas estudadas. Por sua vez, a técnica FIR não apenas dispõem desses elos, mas também garante uma semântica específica para eles. Na Figura 46, a regra de negócio 1 do tipo funcionalidade agrupa a regra de negócio do tipo regra de acordo com a Definição 7; a regra de negócio 2 se relaciona com o requisito de negócio 4 através do consumo da informação Torcedor-CPF de acordo com as Definições 8, 35 e 37; o requisito de negócio 4 se relaciona com a regra de negócio 8 através do consumo da informação Ingresso-CPF de acordo com as Definições 19, 34 e 36; o requisito de negócio 5 se relaciona com a regra de negócio 9 através do consumo da informação Ingresso-nome-evento; o requisito de negócio 6 se relaciona com a regra de negócio 10 através do consumo da informação Ingresso-data-evento, de acordo com as Definições 19, 34 e 36.

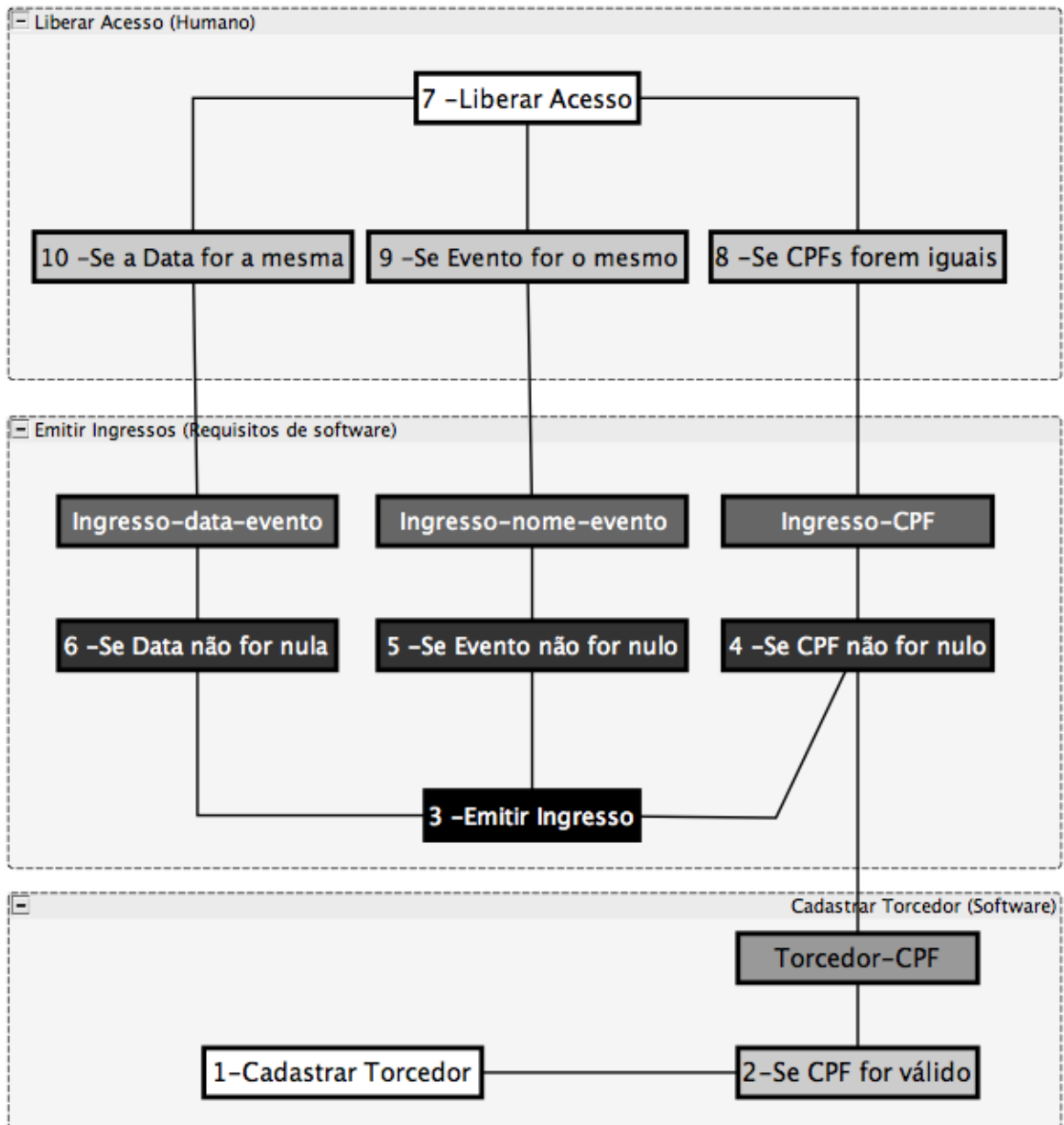


Figura 46 - Rastreamento do exemplo do Capítulo 1 através da Técnica FIR

Nota: Autoria Própria

7 O PLUGIN FIR-Diagram

Neste Capítulo apresentamos a ferramenta FIR-Diagram, desenvolvida como prova de conceito para dar suporte à criação e validação de instâncias de FIR⁹, bem como seu uso para apoiar a atividade de análise de impacto de mudanças solicitadas tanto para regras de negócio, quanto para requisitos de negócio. Na Seção 7.1, os requisitos definidos para o projeto da ferramenta FIR-Diagram são apresentados. Na Seção 7.2 é apresentada a tecnologia utilizada para a construção da ferramenta. A Seção 7.3 trata da construção da ferramenta em si. Finalmente, na Seção 7.4 o exemplo apresentado no Capítulo 1 é construído e validado com o auxílio da ferramenta.

7.1 REQUISITOS PARA FIR-DIAGRAM

A concepção de FIR-Diagram observou os seguintes requisitos:

- a) Q1: A ferramenta deve permitir a criação de instâncias de FIRg em modo gráfico;
- b) Q2: A ferramenta deve permitir a validação de instâncias de elementos a partir das restrições definidas para FIRg;
- c) Q3: A ferramenta deve permitir a realização da atividade de descoberta de impactos sobre uma instância de FIRg;
- d) Q4: A ferramenta deve permitir que instâncias de FIRg possam estar disponíveis para integração com outras ferramentas;
- e) Q5: A tecnologia utilizada no desenvolvimento da ferramenta deve ser relevante tanto na indústria quanto na academia, de modo a motivar futuros interessados na evolução da ferramenta.

A ferramenta foi implementada como um plugin para o ambiente de desenvolvimento (FOUNDATION, 2010), devido à grande popularidade e relevância desse ambiente, tanto na indústria quanto na academia, satisfazendo assim o requisito Q₅. O código fonte do projeto está hospedado no repositório do Google sob a licença Eclipse Public License 1.0 e está disponível através da url: <https://firplugin.googlecode.com/svn/trunk/>. A ferramenta é open source e estará aberta a contribuições da comunidade de desenvolvedores de software. A satisfação

dos requisitos de Q_1 a Q_4 (Figura 47) é detalhada nas seções seguintes. Descrição: O líder de avaliação exclui o modelo.

Q_1	Dentre os <i>frameworks</i> utilizados para desenvolver o <i>plugin</i> , o Graphical Editing Framework (GEF) é o que permite a criação de elementos gráficos. O GEF é apresentado na Seção 7.2.1.
Q_2	No <i>plugin</i> , a funcionalidade de validação dos elementos está disponível a partir de um item de menu. A validação foi implementada parte através de cláusulas OCL e parte através de codificação na linguagem Java.
Q_3	No <i>plugin</i> , a funcionalidade de descoberta de impactos está disponível a partir da seleção de uma instância de elemento do tipo Regra.
Q_4	Dentre os <i>frameworks</i> utilizados para desenvolver o <i>plugin</i> , o Eclipse Modeling Framework (EMF), por ser uma implementação do MOF (Meta Object Facility), predispõe os modelos criados à integração. O EMF é apresentado na Seção 7.2.2.

Figura 47 - Satisfação dos Requisitos
Nota: Autoria Própria

7.2 TECNOLOGIA UTILIZADA NA CONSTRUÇÃO DO PLUGIN

Para desenvolver o plugin foi utilizado o Graphical Modeling Framework (GMF) que está disponível a partir de www.eclipse.org/gmf. O GMF é um framework para geração de editores gráficos de modelos de domínio. O GMF facilita a integração de dois outros frameworks: o Eclipse Modeling Framework (EMF) que está disponível a partir de www.eclipse.org/emf; e o Graphical Editing Framework (GEF) que está disponível a partir de www.eclipse.org/gef.

7.2.1 GEF

GEF é um framework MVC (Model-View-Controller) formado por um conjunto de ferramentas utilizadas para renderizar e exibir elementos gráficos. A Figura 48 mostra elementos do GEF, a saber: Menu e Barra de Ferramentas (toolbar) utilizados para criar editores gráficos. Os eventos gerados pelos elementos gráficos são associados a ações que encaminham requisições para controladores.

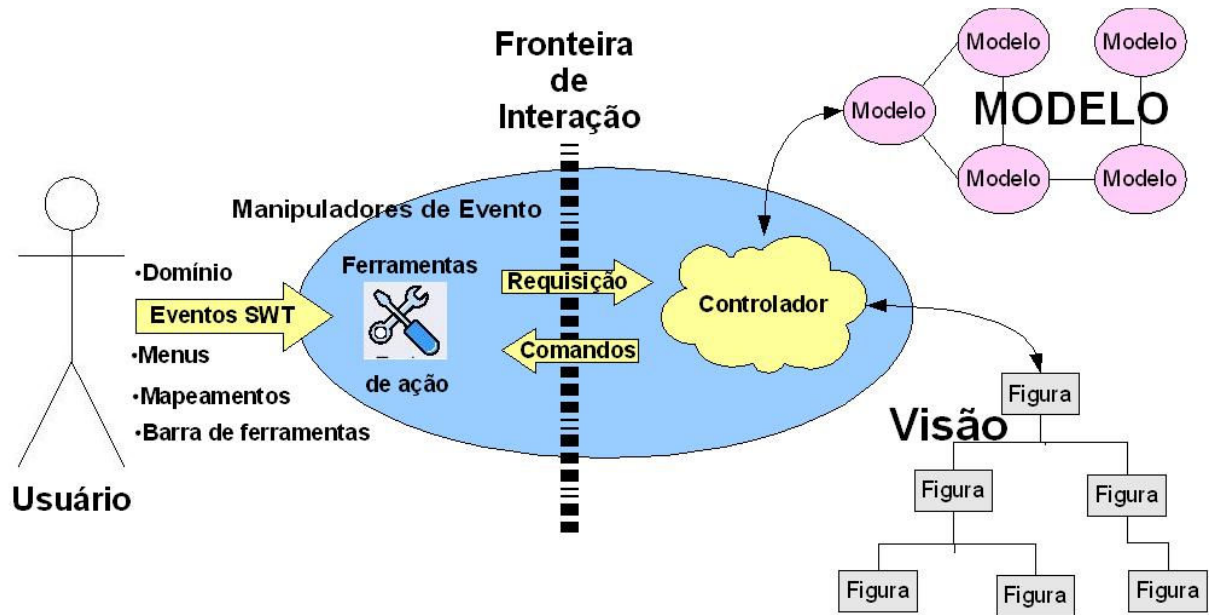


Figura 48 – Visão geral do GEF
Nota: Extraído de Foundation (2010)

7.2.2 EMF

EMF é um framework Java que facilita a geração de código para a construção de ferramentas e outras aplicações. EMF pode ser encarado como uma aplicação Java altamente eficiente de um subconjunto do núcleo da API MOF (Meta Object Facility) especificado pela OMG (Object Management Group).

No EMF, o meta-modelo é chamado de Ecore e utiliza XMI (XML Metadata Interchange) como forma canônica para representar os modelos. O Ecore pode ser obtido das seguintes formas:

- Escrever diretamente em editor de texto;
- Exportar o XMI de uma ferramenta que suporta esta operação;
- A partir de anotações em interfaces Java;
- A partir de XML Schema para definir como o modelo vai ser serializado.

7.2.3 Integração entre EMF e GEF

Para integrar o EMF ao GEF, o GMF define o processo apresentado na Figura 49.

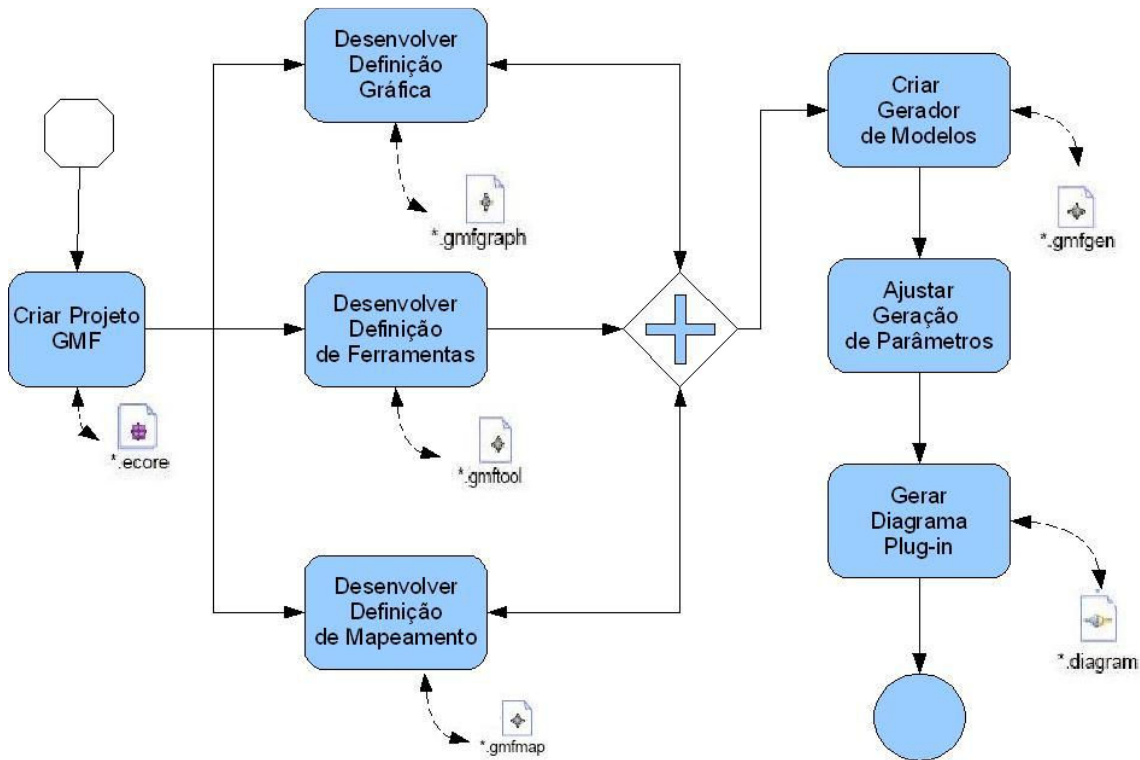


Figura 49 - Processo do GMF para Integrar o EMF ao GEF
 Notas: Extraído de Foundation (2010)

A seguir a Figura é detalhada:

- a) **Definição Gráfica (Graphical Definition):** Estabelece uma plataforma neutra para a definição de figuras, nós e nós filhos, elos entre os nós, compartimentos, formas (shapes), rótulos (labels) e conexões, além de permitir a customização de figuras.
- b) **Definição de Ferramenta (Tooling Definition):** Um modelo para definir as partes de ferramentas, tais como: paleta, menus, ações, barra de ferramenta.
- c) **Modelo de Mapeamento (Mapping Model):** Um modelo para especificar os relacionamentos entre: elementos de domínio (EMF em si), elementos gráficos (uma parte do GEF), elementos da ferramenta (outra parte do GEF), além de permitir que validações sejam definidas juntamente com restrições via OCL e expressões regulares em Java.
- d) **Modelo de Geração (Generation Model):** Um modelo utilizado para especificar a geração de código a partir do modelo gerado no passo anterior. Para a geração do código Java é utilizado o JET (Java Emitter Templates).

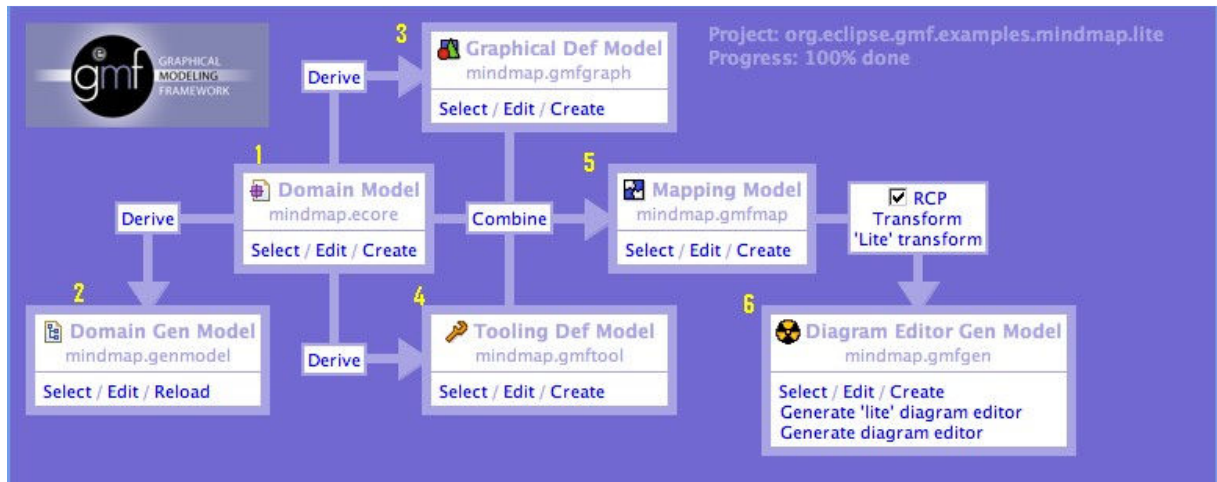


Figura 50 - Relação entre os modelos no processo GMF
 Notas: Extraído de Foundation (2010)

A Figura 50 exibe uma ferramenta utilizada a partir do Eclipse para guiar o desenvolvedor no processo descrito acima. São seis passos que estão descritos a seguir.

- O meta-modelo Ecore com elementos específicos de domínio é definido;
- O Ecore definido em (1) é transformado para um modelo específico de plataforma, nesse caso um modelo que permite a geração de código Java;
- O Ecore definido em (1) é transformado para um meta-modelo que representa a parte gráfica dos elementos;
- O Ecore definido em (1) é transformado para um meta-modelo que define as ferramentas do editor gráfico;
- Os dois meta-modelos criados em (3) e (4) são mapeados para gerar o código. No mapeamento é possível criar restrições utilizando a linguagem OCL (Object Constraint Language) ou JAVA;
- O meta-modelo definido em (5) é utilizado para gerar o código específico para o editor gráfico.

7.3 CONSTRUÇÃO DO PLUGIN FIR-DIAGRAM

A construção do plugin FIR-Diagram foi feita a partir do processo apresentado na Seção 7.2.3

7.3.1 Construção do *fir.ecore*

Nessa etapa foi utilizado o EMF para a construção do modelo FIR. A Figura 51 representa o diagrama de classes do modelo FIR.

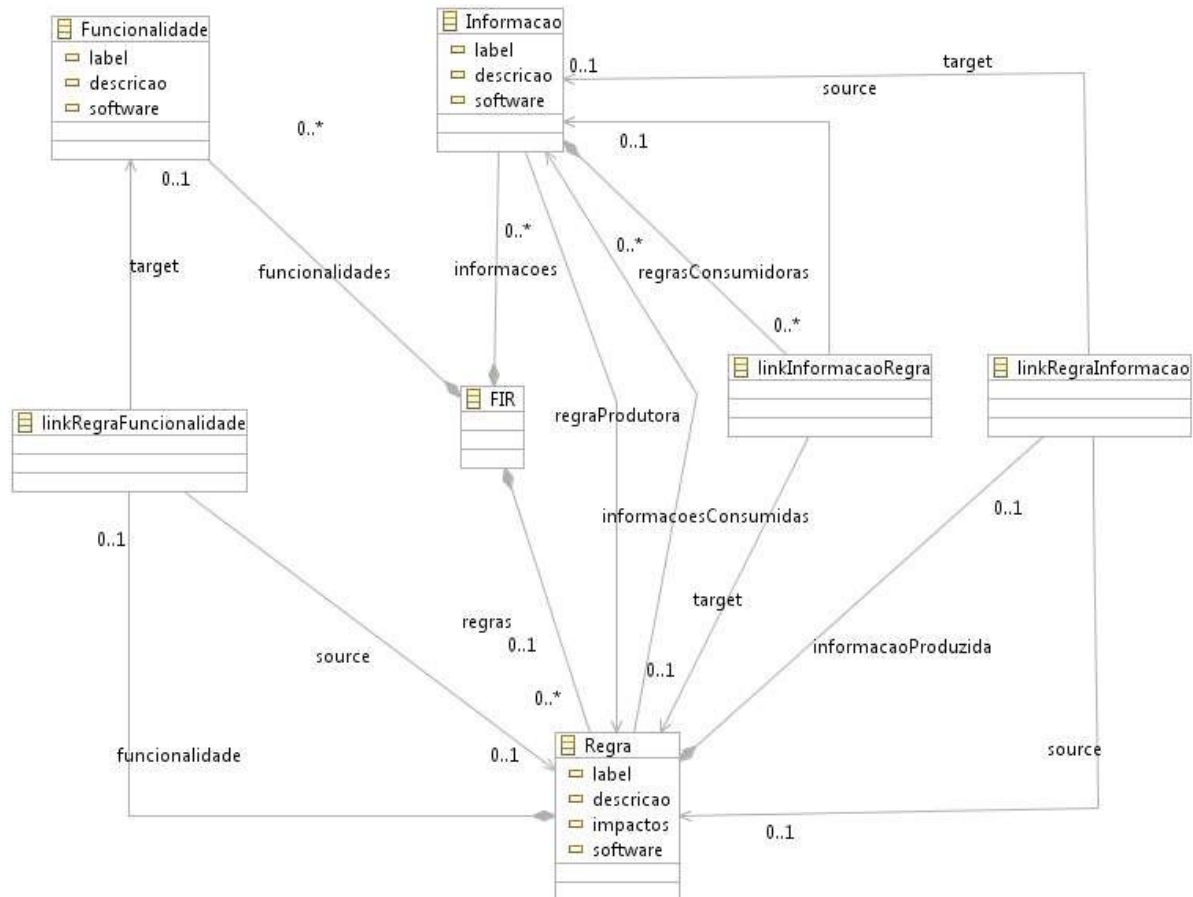


Figura 51 – Diagrama ecore
Nota: Extraído de Foundation (2010)

- a) A classe **FIR**: Essa classe representa o conceito domain do EMF. Uma Classe do tipo domain se torna um container dos demais elementos que compreendem os conceitos node e link;
- b) A classe **Informacao**: Essa classe corresponde ao conceito Informação. O atributo descrição corresponde ao nome por extenso da informação, por exemplo, CPF do Ingresso. O atributo label corresponde ao nome resumido da informação, por exemplo, i2. O atributo software é Boolean. Para o exemplo, se o atributo software for true, significa que a informação i2 pertence ao conjunto I_n^S (Definição

17), pelo contrário se software for false significa que i2 pertence ao conjunto I_s^n (Definição 27).

- c) A classe **Regra**: Essa classe corresponde ao conceito Regra. O atributo descricao corresponde a regra em si, por exemplo, Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado. O atributo label serve para identificar o nó da regra no grafo, por exemplo, r4. O atributo impactos serve para conter as demais regras impactadas, os detalhes estão na Seção 7.4. O atributo software é Boolean. Para o nosso exemplo, se o atributo software for true, significa que a regra r4 pertence ao conjunto B_n^s (Definição 16) das regras requisito do software de Emissão de Ingressos, pelo contrário se software for false significa que r4 pertence ao conjunto B_s^n (Definição 26).
- d) A classe **Funcionalidade**: Essa classe corresponde ao conceito Funcionalidade. O atributo descricao corresponde ao nome por extenso da funcionalidade, por exemplo, Emitir Ingresso. O atributo label corresponde ao nome resumido da funcionalidade, por exemplo, f2. O atributo software é Boolean. Para o nosso exemplo, se o atributo software for true, significa que a funcionalidade f2 pertence ao conjunto F_n^s (Definição 15) das funcionalidades-requisito do software de Emissão de Ingressos, pelo contrário se software for false significa que f2 pertence ao conjunto F_s^n (Definição 25).
- e) A classe **LinkRegraFuncionalidade**: Essa classe representa elementos das relações: Θ_n^s (Definição 18) e Θ_s^n (Definição 28). Por exemplo, se f2.software=true e r4.software=true então o par $(r4,f2) \in \Theta_n^s$; por outro lado, se f2.software=false e r4.software=false, então o par $(r4,f2) \in \Theta_s^n$.
- f) A classe **LinkInformacaoRegra**: Essa classe representa elementos das relações: (a) \check{I}_n^s (Definição 20), para i2.software=true e r4.software=true, então o par $(r4,i2) \in \check{I}_n^s$; (b) \check{I}_s^n (Definição 30), para i2.software=false e r4.software=false, então o par $(r4,i2) \in \check{I}_s^n$; (c) \vec{I}_n^s (Definição 34), para i2.software=true e r4.software=false, então o par

$\$(r4,i2) \in \vec{\Gamma}_n^s$; e (d) $\vec{\Gamma}_s^n$ (Definição 35), para $i2.software=false$ e $r4.software=true$, então o par $(r4,i2) \in \vec{\Gamma}_s^n$.

- g) A classe **LinkRegralInformacao**: Essa classe representa elementos das relações: (a) $\hat{\Gamma}_n^s$ (Definição 19), para $i2.software=true$ e $r4.software=true$, então o par $(r4,i2) \in \hat{\Gamma}_n^s$; (b) $\hat{\Gamma}_s^n$ (Definição 29), para $i2.software=false$ e $r4.software=false$, então o par $(r4,i2) \in \hat{\Gamma}_s^n$; (c) $\vec{\Gamma}_n^s$ (Definição 34), para $i2.software=true$ e $r4.software=false$, então o par $(r4,i2) \in \vec{\Gamma}_n^s$; e (d) $\vec{\Gamma}_s^n$ (Definição 35), para $i2.software=false$ e $r4.software=true$, então o par $(r4,i2) \in \vec{\Gamma}_s^n$.

7.3.2 Construção do *fir.gmfgraph*

Na etapa Definição Gráfica do processo GMF foram definidos os três tipos de elementos gráficos que corresponderão aos elementos do meta-modelo presentes em *fir.ecore*. De acordo com a Figura 52, os elementos gráficos em forma de círculo são do tipo Node; os textos em cada círculo são do tipo Label; e os elos entre os círculos são do tipo Connection. Na definição gráfica de cada elemento podem ser atribuídas propriedades, tais como: cor de fundo, cor da borda, tamanho, forma da figura, fonte dentre outras.

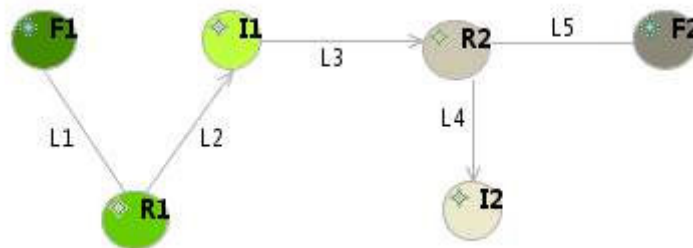


Figura 52 – Elementos de FIR-Diagram
Nota: Autoria Própria

7.3.3 Construção do *fir.gmftool*

Na etapa Definição de Ferramenta do processo GMF, definiu-se a paleta com itens que corresponderão aos elementos gráficos definidos em *fir.gmfgraph* e que serão movidos para a área de edição. A Figura 53 mostra a paleta com seus itens.

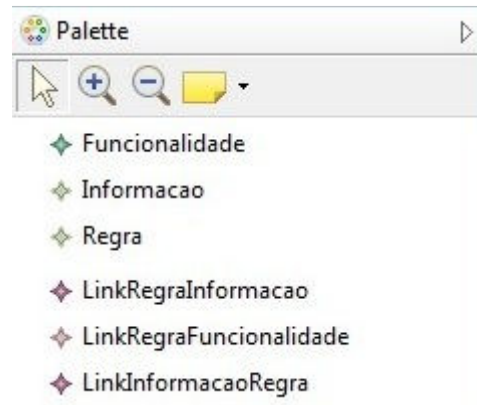


Figura 53 – Paleta de Componentes do FIR-Diagram
Nota: Autoria Própria

7.3.4 Construção do *fir.gmfmap*

Na etapa Mapeamento de Modelo do processo GMF, as classes do domínio definidas em *fir.ecore* são mapeadas para os elementos gráficos que foram definidos em *fir.gmfgraph*. As Figuras 54 e 55 mostram a relação entre os elementos gráficos e as classes do modelo.

Classes	Elementos Gráficos	Cor	Detalhes
Funcionalidade		RGB(69,139,0)	<i>software=true</i>
Regra		RGB(102,205,0)	<i>software=true</i>
Informacao		RGB(192,255,62)	<i>software=true</i>
Funcionalidade		RGB(69,139,0)	<i>software=false</i>
Regra		RGB(102,205,0)	<i>software=false</i>
Informação		RGB(192,255,62)	<i>software=false</i>

Figura 54 – Relacionamentos entre as classes do modelo e os correspondentes elementos gráficos
Nota: Autoria Própria

Classes	Elementos Gráficos	Tipo da Linha	Fonte	Alvo
LinkRegraFuncionalidade		LINE_SOLID	R_1	F_1
LinkRegraInformacao		LINE_SOLID	R_1	I_1
LinkInformacaoRegra		LINE_SOLID	I_1	R_1

Figura 55 – Relacionamentos entre as classes do modelo e os elos no gráfico
Nota: Autoria Própria

7.3.5 Implementação de Restrições no Modelo

As restrições impostas pelas definições do FIR⁹ foram implementadas através de expressões em OCL ou implementadas diretamente em código. Restrições sobre o estado local dos atributos foram implementadas em OCL (Figura 56). Restrições sobre uma instância que consulta o estado de atributos de uma outra instância foram implementadas em Java (Figura 57).

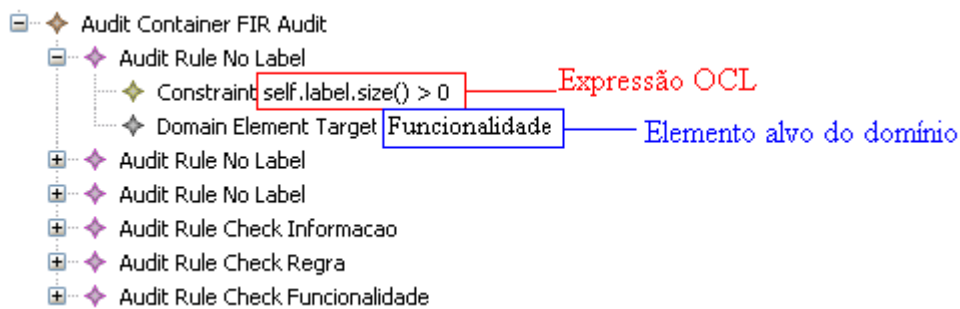


Figura 56 – Expressão em OCL para o atributo label da classe Funcionalidade
Nota: Autoria Própria

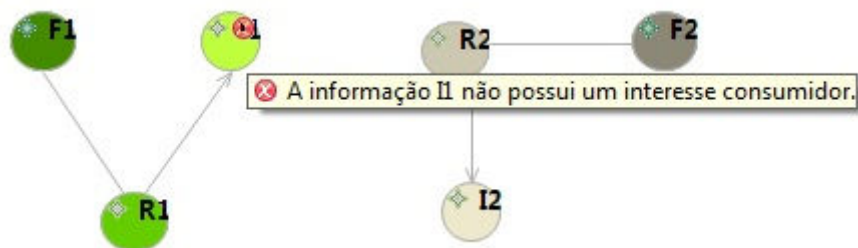


Figura 57 - Restrição implementada em Java
Nota: Autoria Própria

7.4 O EXEMPLO DE EMISSÃO DE INGRESSOS

A Figura 58 apresenta o FIR-Diagram para o exemplo apresentado no Capítulo 1 e a Figura 59 mostra a correspondência de cada instância de regra de negócio.

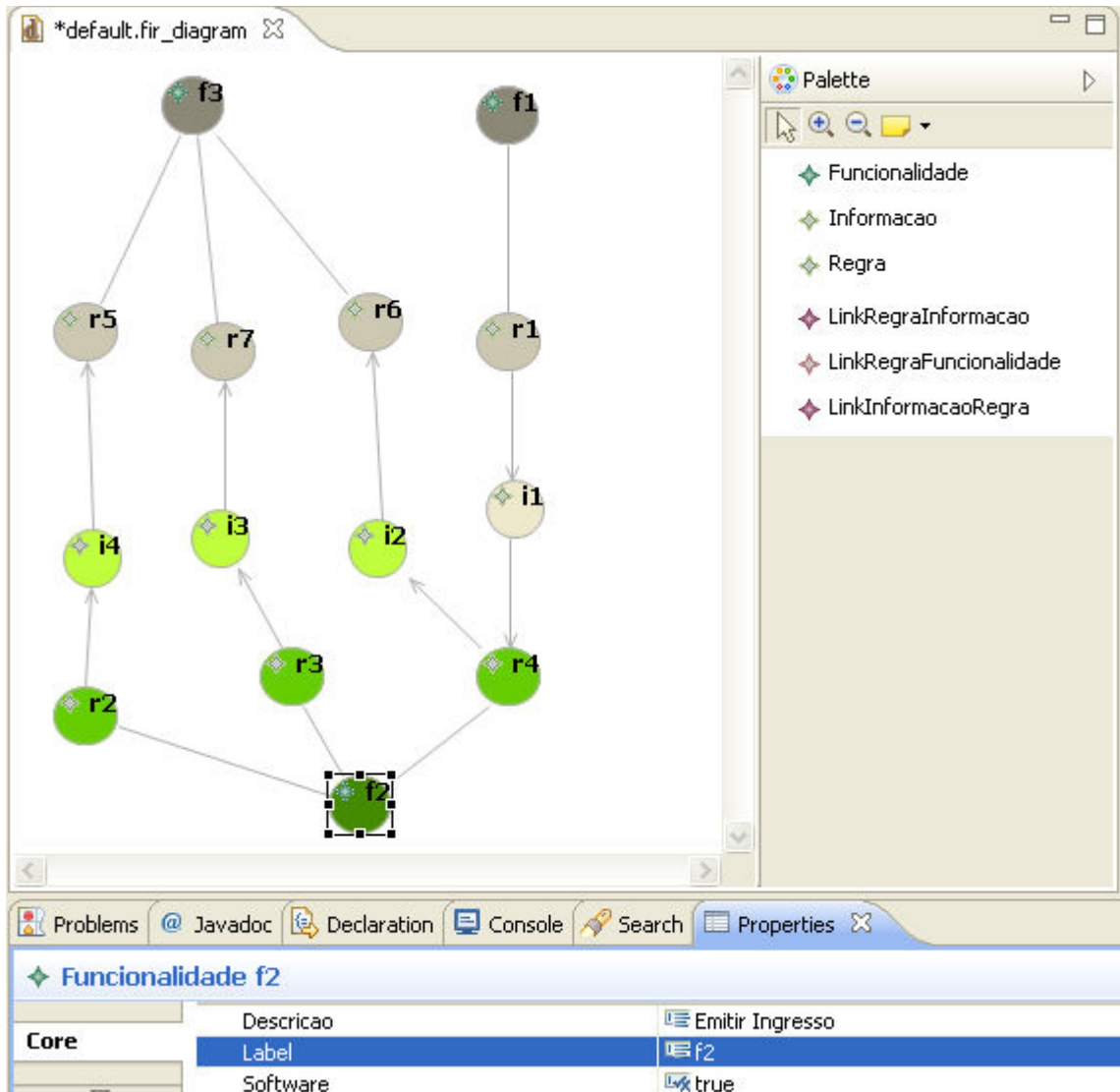


Figura 58 – Diagrama FIR do Exemplo apresentado no Capítulo 1
Nota: Autoria Própria

Funcionalidade f_1	<i>Cadastrar Torcedor</i>
Funcionalidade f_2	<i>Emitir Ingresso</i>
Funcionalidade f_3	<i>Liberar Acesso</i>
Regra r_1	<i>Se o CPF informado for válido, então o Torcedor-CPF é igual ao CPF informado</i>
Regra r_2	<i>Se Nome do Evento informado for válido, então o Ingresso-Nome-do-Evento é igual ao Nome do Evento informado</i>
Regra r_3	<i>Se a Data do Evento informada for válida, então o ingresso-Data-do-Evento é igual à Data do Evento informada</i>
Regra r_4	<i>Se Torcedor-CPF for válido, então o ingresso-Torcedor-CPF é igual à Torcedor-CPF informado</i>
Regra r_5	<i>O Ingresso-Nome-do-Evento deve ser igual ao nome do evento para que o acesso seja liberado</i>
Regra r_6	<i>O Ingresso-CPF deve ser igual ao CPF-do-Torcedor para que o acesso seja liberado</i>
Regra r_7	<i>O Ingresso-Data-do-Evento deve ser igual a Data do Evento para que o acesso seja liberado</i>
Informação i_1	Torcedor-CPF
Informação i_2	Ingresso-CPF
Informação i_3	Ingresso-Data-do-Evento
Informação i_4	Ingresso-Nome-do-Evento

Figura 59 – Revisitando o exemplo do Capítulo 1

Nota: Autoria Própria

A análise de impacto sobre o modelo FIR com o apoio da ferramenta FIR-Diagram é automática. Para realizar a atividade de descoberta de impactos foram considerados os elos de dependência entre os elementos do modelo FIR. O algoritmo de descoberta de impactos determina quais regras serão impactadas caso uma mudança ocorra em uma determinada regra do modelo.

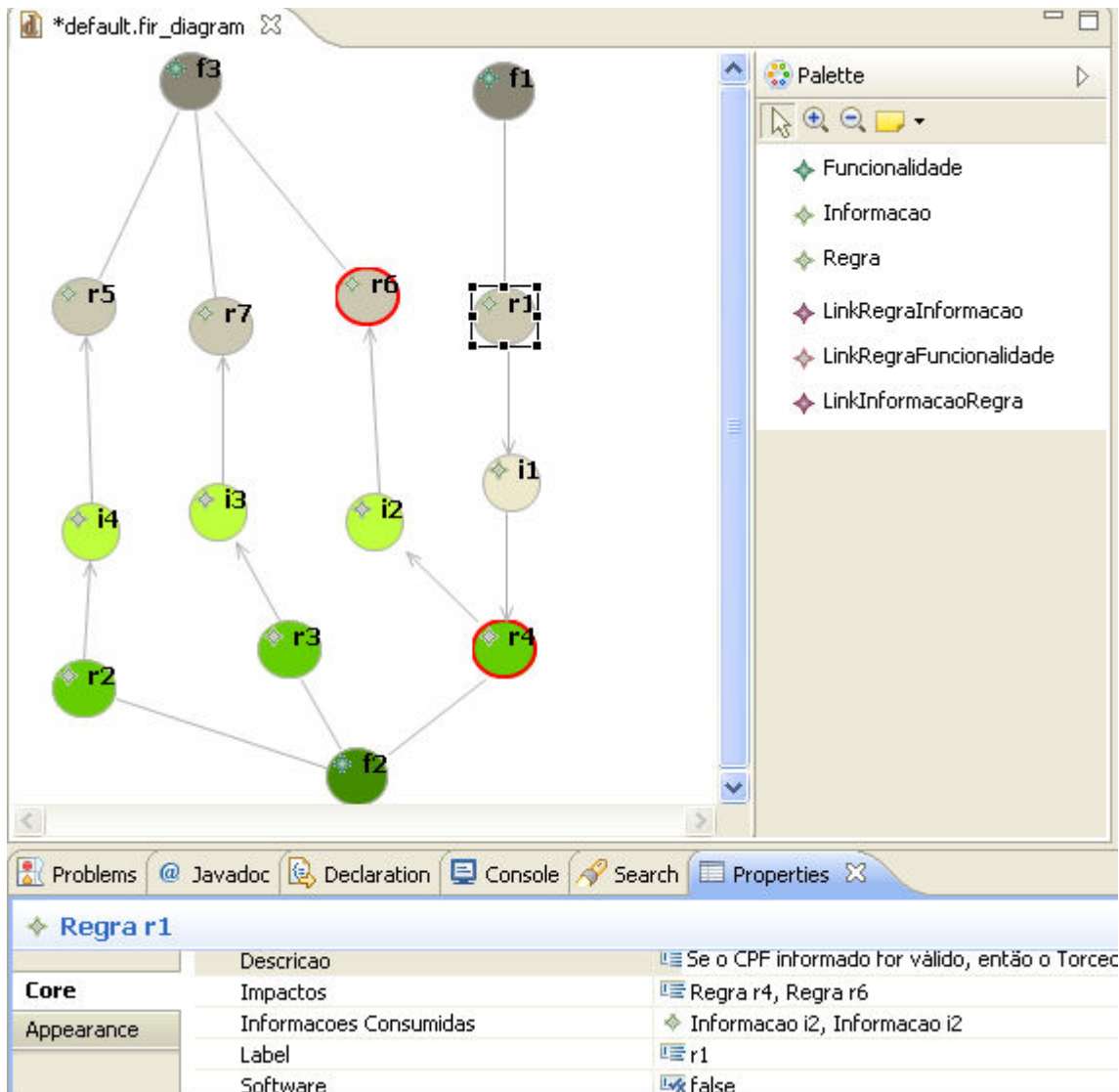


Figura 60 – Descoberta de Impactos através do FIR-Diagram
Nota: Autoria Própria

Quando um vértice do tipo Regra é selecionado no modelo, o algoritmo de descoberta de impactos percorre os demais vértices do grafo em busca dos vértices do tipo Regra que dependem da informação produzida pelo vértice selecionado. Cada vértice encontrado é adicionado na lista de impactos da regra selecionada. O algoritmo é executado recursivamente para cada regra dependente até não haver mais regras. Ao final, as regras impactadas são exibidas com um círculo vermelho ao redor.

A Figura 60 mostra que a atividade de descoberta de impactos para a regra r_1 identifica duas regras dependentes: r_4 e r_6 sinalizadas com um círculo vermelho em torno de cada uma.

8 CONCLUSÃO

Análise de impacto é a identificação das potenciais consequências de uma mudança, ou estimativa do que precisa ser modificado para realizar uma mudança, incluindo estimativas de custos e cronograma. Neste trabalho, várias técnicas de rastreabilidade foram estudadas e foi observado que nenhuma delas apóia a análise de impactos sobre interesses específicos do negócio com valores resultantes razoáveis de precisão e revocação para os impactos descobertos.

Neste trabalho propusemos uma técnica de rastreabilidade denominada FIR, endereçada às deficiências encontradas em quinze técnicas estudadas. Em especial a técnica FIR considera interesses específicos do negócio para apoiar a análise de impacto, utilizando um novo modelo de rastreabilidade definido sobre regras de negócio, com resultados de precisão e de revocação esperados de 100%. Uma ferramenta chamada de FIR-Diagram foi desenvolvida para ilustrar a viabilidade da técnica proposta.

8.1 CONTRIBUIÇÕES

Destacamos as seguintes contribuições deste trabalho:

- a) Um modelo de rastreabilidade denominado FIR que considera interesses de negócio para dar suporte à análise de impacto de mudanças;
- b) Uma interpretação do modelo acima em termos de um modelo de requisitos para dar igual suporte a análise de impacto de mudanças tanto nas regras de negócio quanto nos requisitos de negócio.
- c) Uma nova técnica de rastreabilidade apoiada sobre o modelo FIR projetada para alcançar valores de precisão e revocação iguais a 100%.
- d) Uma ferramenta implementada como um plugin para a plataforma Eclipse que suporta o modelo FIR e automatiza a atividade de descoberta de impactos.

8.2 TRABALHOS FUTUROS

Destacamos os seguintes trabalhos futuros:

- a) **A Técnica FIR é manual:** Alguns trabalhos futuros podem contribuir para automatizar a construção do FIR. Uma abordagem a ser explorada é a utilizada por técnicas de extração de regras de negócio de código fonte.
- b) **A Técnica FIR somente dispõe de elos de dependência:** A extração de regras de negócio de código fonte, em geral evidencia as partes do código onde as regras de negócio estão codificadas; uma estratégia seria estender o FIR para suportar elos de satisfação gerados automaticamente com a extração.
- c) **Necessidade de Experimentação:** Embora a técnica FIR permita a construção de uma rastreabilidade que favorece a descoberta de impactos, ainda não há evidências experimentais da precisão e da revocação da atividade de descoberta de impactos, dentre outras. No Capítulo 4, foram levantadas algumas hipóteses sobre dificuldades relacionadas algumas técnicas de rastreabilidade para orientar a criação de elos de dependência entre requisitos, especialmente, entre requisitos de negócio. Para confirmar ou refutar essas hipóteses, experimentos precisam ser realizados. Até o presente momento, foi feito o planejamento de um conjunto de três experimentos, com relatórios estão disponíveis em: <http://wiki.dcc.ufba.br/Aside>. Os experimentos devem tentar validar o modelo de interesses de negócio como também a precisão e revocação da atividade de descoberta de impactos. A execução dos experimentos será objeto de trabalho futuro.

REFERÊNCIAS

ANTONIOL, Giuliano et al. Recovering traceability links between code and documentation. **Ieee Transactions On Software Engineering**, Piscataway, Nj, Usa, n. , p.970-983, out. 2002.

ARNOLD, Robert S.. **Software change impact analysis**. Los Alamitos, Ca, Usa: Ieee Computer Society Press, 1996.

BUSINESS RULES GROUP (Org.). **Business Rules**. Disponível em: <www.businessrulesgroup.org>. Acesso em: 26 jun. 2010.

CLELAND-HUANG, Jane. Just enough requirements traceability. In: ANNUAL INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE, 30., 2006, Chicago. **Proceedings...** . Washington, Dc, Usa: Ieee Computer Society, 2006. p. 41 - 42.

CLELAND-HUANG, Jane; CHANG, Carl K.; CHRISTENSEN, Mark. Event-based traceability for managing evolutionary change. **Ieee Transactions On Software Engineering**, Piscataway, Nj, Usa, n. , p.796-810, set. 2003.

CLELAND-HUANG, Jane; CHANG, Carl K.; GE, Yujia. Supporting event based traceability through high-level recognition of change events. In: INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE ON PROLONGING SOFTWARE LIFE: DEVELOPMENT AND REDEVELOPMENT, 26., 2002, Oxford. **Proceedings...** . Washington, Dc, Usa: Ieee Computer Society, 2002. p. 595 - 602.

CYSNEIROS, L. M.; LEITE, Julio C. S. do Prado. Using iso 9000 to elicit business rules. In: IEEE INTERNATIONAL SYMPOSIUM AND FORUM ON SOFTWARE ENGINEERING STANDARDS, 4., 1999, Curitiba, Brasil. **Proceedings...** . Washington, Dc, Usa: Ieee Computer Society, 1999. p. 88 - 99.

EGYED, Alexander. A scenario-driven approach to traceability. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 23., 2001, Toronto, Ontario, Ca. **Proceedings...** . Washington, Dc, Usa: Ieee Computer Society, 2001. p. 123 - 132.

FOUNDATION, Eclipse (Org.). GMF, EMF e GEF. Disponível em: <www.eclipse.org>. Acesso em: 29 jun. 2010.

GOTEL, Orlena. **Contribution structures for requirements engineering**. 1996. 177 f. Phd Thesis (Phd) - Imperial College Of Science, London, U.K, 1996.

GOTEL, Orlena; FINKELSTEIN, Anthony. An analysis of the requirements traceability problem. In: INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING, 1., 1994, Colorado Springs, Co, Usa. **Proceedings...** . Colorado Springs, Co, Usa: Ieee Computer Society, 1994. p. 94 - 101. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=292398>. Acesso em: 29 jun. 2010.

HEINDL, Matthias; BIFFL, Stefan. An analysis of the requirements traceability problem. In: INTERNATIONAL SYMPOSIUM ON FOUNDATIONS OF SOFTWARE ENGINEERING, 13., 2005, Lisbon, Portugal. **Proceedings...** . New York, Ny, Usa: Acm, 2005. p. 60 - 69.

KASSE, Tim. **Practical Insight into CMMI**. Norwood, Ma, Usa: Artech House, Inc., 2004.

LEITE, Julio C. S. do Prado; LEONARDI, Maria Carmen. Business rules as organizational policies. In: INTERNATIONAL WORKSHOP ON SOFTWARE SPECIFICATION AND DESIGN, 9., 1998, Ise-shima , Japan. **Proceedings...** . Washington, Dc, Usa: Ieee Computer Society, 1998. p. 68 - 76.

LEITE, Julio C. S. do Prado; OLIVEIRA, Alcione P.. A client oriented requirements baseline. In: IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING, 2., 1995, York, Uk. **Proceedings...** . Washington, Dc, Usa: Ieee Computer Society, 1995. p. 108 - 115.

LEHMAN, Meir Manny. Laws of software evolution revisited. In: EUROPEAN WORKSHOP ON SOFTWARE PROCESS TECHNOLOGY, 5., 1996, Nancy. **Proceedings...** . London, Uk: Springer-verlag, 1996. p. 108 - 124.

MALETIC, Jonathan et al. A client oriented requirements baseline. In: INTERNATIONAL WORKSHOP ON TRACEABILITY IN EMERGING FORMS OF SOFTWARE ENGINEERING, 2., 2003, Montreal, Canada. **Proceedings...** . Montreal, Canada: Center Of Excellence For Software Traceability, 2003. p. 47 - 54.

OLIVEIRA FILHO, Antônio; MENDONÇA NETO, Manoel Gomes de; CHAVEZ, Christina Von Flach G.. Em busca de agilidade na análise de impacto: o artefato FIR. **Latin American Transactions**, Recife, Pe, Brasil, n. , p.275-281, jul. 2008.

Disponível em:

<<http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=4653851&isYear=>>. Acesso em: 29 jun. 2010.

PASHOV, I.; RIEBISCH, M.. Using feature Modeling for program comprehension and software architecture recovery. In: ANNUAL IEEE INTERNATIONAL CONFERENCE AND WORKSHOP, 11., 2004, Brno, Czech Republic. **Proceedings...** . Washington, Dc, Usa: Ieee Computer Society, 2004. p. 406 - 418.

RAJA, Uzair Akbar; KAMRAN, Kashif.. **Framework for requirements traceability: TLFRT supporting pre-rs & post-rs traceability**. 2008. 55 f. Master Thesis (Master) - Blekinge Institute Of Technology, Betul, India, 2008.

RAMESH, B.; JARKE, M.. Toward reference models for requirements traceability. **Ieee Trans. Softw. Eng.**, Piscataway, Nj, Usa, n. , p.58-93, jan. 2001.

RIEBISCH, M.. Supporting evolutionary development by feature models and traceability links. In: ANNUAL IEEE INTERNATIONAL CONFERENCE AND WORKSHOP, 11., 2004, Brno, Czech Republic. **Proceedings...** . Washington, Dc, Usa: Ieee Computer Society, 2004. p. 370 - 377.

ROCHIMAH, Siti; KADIR, Wan M. N. Wan; ABDULLAH, Abdul H.. An evaluation of traceability approaches to support software evolution. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING ADVANCES, 2., 2007, Cap Esterel, French Riviera, France. **Proceedings...** . Washington, Dc, Usa: Ieee Computer Society, 2007. p. 19 - 19.

ROSCA, Daniela; FEBLOWITZ, Mark; WILD, Chris. A decision making methodology in support of the business rules lifecycle. In: IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING, 3., 1997, Annapolis, Md, Usa.

Proceedings... . Washington, Dc, Usa: Ieee Computer Society, 1998. p. 236 - 246. Disponível em: <<http://www.computer.org/portal/web/csdl/proceedings/r#5>>. Acesso em: 29 jun. 2010.

SAYÃO, M.; LEITE, Julio C. S. do Prado. Rastreabilidade de requisitos. **Revista de Informática Teórica e Aplicada**, Porto Alegre, RS, p.57-86, 05 jan. 2006.

SOMMERVILLE, I.. **Engenharia de software**. 7. ed. São Paulo, Sp: Addison Wesley Longman, 2006.

TORANZO, Marco. **Uma proposta para melhorar o rastreamento de requisitos de software**. 2002. 95 f. Phd Thesis (Phd) - Departamento de Centro de Informática, Universidade Federal de Pernambuco, Recife, PE, 2002.

VON HALLE, Barbara. **Business rules applied: building better systems using the business rules approach**. New York, Ny, Usa: John Wiley & Sons, Inc., 2001.

WIEGERS, Karl E.. **Software requirements**. New York, Ny, Usa: Microsoft Press, 1999.

WIERINGA, R.. **An Introduction to requirements traceability**. Faculty Of Mathematics And Computer Science, Amsterdam: Espirit Project, 1995.

YOUNG, Ralph. **Effective requirements practices**. Boston: Addison-wesley Longman Publishing Co., 2001.

ZEMONT, Grant. **Towards value-based requirements traceability**. 2005. 40 f. Master Thesis (Master) - Departamento de Dept. Of Computer Science, Depaul University, Chicago Illinois, 2005.