UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DecKiPO - EXPANSÃO DA EXPRESSIVIDADE SEMÂNTICA NA REPRESENTAÇÃO DE REGRAS DE NEGÓCIO EM CENÁRIOS DE PROCESSOS INTENSIVOS EM CONHECIMENTO

Rodrigo Benits Lyrio

Orientadores

Fernanda Araújo Baião

Gleison dos Santos Souza

RIO DE JANEIRO, RJ – BRASIL

SETEMBRO de 2018

DecKiPO - EXPANSÃO DA EXPRESSIVIDADE SEMÂNTICA NA REPRESENTAÇÃO DE REGRAS DE NEGÓCIO EM CENÁRIOS DE PROCESSOS INTENSIVOS EM CONHECIMENTO

Rodrigo Benits Lyrio

DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO). APROVADA PELA COMISSÃO EXAMINADORA ABAIXO ASSINADA.

Aprovada por:

Prof. Dr. Gleison dos Santos Souza, D.Sc. – UNIRIO

Prof. Dr. Sean Wolfgand Matsui Siqueira – D.Sc. – UNIRIO

Prof. Dr. Ricardo de Almeida Falbo, D.Sc. – Universidade Federal do Espírito Santo - UFES

RIO DE JANEIRO, RJ – BRASIL

SETEMBRO de 2018

Catalogação informatizada pelo(a) autor(a)

Lyrio, Rodrigo Benits
L992 DecKiPO - Expansão da expressividade semântica na representação de regras de negócio em cenários de Processos Intensivos em Conhecimento / Rodrigo Benits Lyrio. -- Rio de Janeiro, 2018.
131f

Orientador: Fernanda Araujo Baião e Gleison dos Santos Souza .
Dissertação (Mestrado) - Universidade Federal do Estado do Rio de Janeiro, Programa de Pós-Graduação em Informática, 2018.

1. Processos intensivos em Conhecimento. 2. Regras de Negócio. 3. LTL Declare. 4. Modelagem Declarativa. 5. KiPO. I. , Fernanda Araujo Baião e Gleison dos Santos Souza, orient. II. Título.

Agradecimentos

Primeiramente, agradeço a todos que compartilharam mais essa etapa na minha vida, em especial as pessoas citadas a seguir.

À Danielle Pereira, minha esposa, que esteve ao meu lado por toda a jornada e demonstrando paciência, compreensão e, principalmente, apoio, ao longo dos vários dias chegando tarde das aulas e reuniões do grupo e que há mais de 12 anos está do meu lado a cada nova empreitada de estudos que inicio.

À Christiani Rocha, minha chefe, cuja relação e confiança no trabalho já duram 10 anos e que prestou todo o apoio permitindo me ausentar por várias tardes ao longo desses anos de mestrado para frequentar as aulas na Universidade.

À Fernanda Baião, minha orientadora, que desde o início acreditou no meu potencial e no trabalho que era capaz de fazer. Sem sua liderança e cobrança, eu jamais teria conseguido chegar a esta etapa dessa jornada. Foi extremamente gentil e companheira revisando e contribuindo para a conclusão do meu trabalho mesmo sem ter, na reta final do mestrado, qualquer obrigação. A ela, especialmente, serei eternamente grato.

Aos meus colegas do grupo de pesquisa ALICE que me ajudaram em diversas discussões e ideias implementadas.

Lyrio, Rodrigo Benits. **DecKiPO – Expansão da expressividade semântica na representação de regras de negócio em cenários de processos intensivos em conhecimento**. UNIRIO, 2018. 131 páginas. Dissertação de Mestrado. Departamento de Informática Aplicada, UNIRIO.

RESUMO

Organizações têm adotado a Gestão de Processos de Negócio (BPM) como principal técnica de entender e gerenciar as atividades realizadas para atingir um objetivo, geralmente descrevendo-as na forma de fluxos de controle bem estruturados. Alguns processos de negócio, entretanto, possuem características dinâmicas em geral por possuírem atividades complexas que dependem do conhecimento de quem as executa, denominados Processos intensivos em Conhecimento, o que dificulta a representação de um fluxo de controle claro e gerenciável. Essa dissertação propõe a representação de KiPs em função das restrições existentes em certo domínio, impostas através de regras de negócio e de como essas restrições são percebidas pelos agentes em termos de necessidade de cumprimento. Esta pesquisa possui um enfoque nas definições teóricas desses elementos e a representação de axiomas, visando definir uma conceituação precisa e bem fundamentada, chamada DecKiPO. A Ontologia de Processos intensivos em Conhecimento (KiPO) e a Ontologia de Regras de Negócio (BRO) são os pontos de partida para caracterizar KiPs de forma independente de domínio, em busca de explorar os conceitos presentes em um KiP e descrever precisamente as restrições impostas por regras de negócio e que norteiam a execução das atividades desses tipos de processos. Cenários de estudo foram identificados de modo a realizar a avaliação ontológica através da representação de cenários reais em comparação com o atual metamodelo existente, formando um conjunto de evidências para a análise e discussão dos novos conceitos e seu impacto na caracterização e compreensão de um KiP.

Palavras-chave: Processos intensivos em Conhecimento, Regras de Negócio, LTL Declare, Modelagem Declarativa, Metamodelo, KiPO.

ABSTRACT

Organizations have adopted Business Process Management (BPM) as the main technique to understand and manage activities in order to achieve their goals, describing them in the form of well-structured control flows. Some business processes, however, have dynamic characteristics mostly because complex activities are executed by its agents and depend on their knowledge to be completed. This kind of processes are called Processes Intensive in Knowledge (KiP), which makes it difficult to represent a clear and manageable control flow.

This work proposes the representation of KiPs in based on constraints in a given domain, that are imposed by business rules that define different restrictions between activities and how these restrictions are perceived by the agents. The research has a focus on the theoretical definitions of these elements and axioms representation, aiming to define a precise and well-founded conceptualization, called DecKiPO. The Knowledge-intensive Process Ontology (KiPO) and the Business Rules Ontology (BRO) are the starting points for characterizing KiPs in order to explore the concepts present in a KiP and to precisely describe the constraints imposed by business rules on activities.

With regard to the scientific methodology for the proposed research, research scenarios were identified in order to perform an ontological evaluation through the representation of real-world scenarios in comparison with the existing KiPO metamodel, providing a set of evidences for analysis and discussion of the new concepts and their impact in the characterization and understanding of a KiP.

Keywords: Knowledge-intensive Process, Business Rules, LTL Declare, Declarative modeling, metamodel, KiPO

Sumário

Agradecimentos iii

Sumário vi

Lista de Figuras viii

Lista de Tabelas xi

1 Introdução 1

1.1 Problema 3

1.2 Hipótese e Proposta de Solução 3

1.3 Objetivo 4

1.4 Metodologia científica 5

1.5 Organização do Texto 6

2 Sobre Paradigmas de modelagem de processos de negócio, Ontologia e Metamodelos 7

2.1 UFO, uma ontologia de fundamentação 10

2.2 Paradigma imperativo e declarativo 16

2.3 A linguagem LTL Declare 18

2.4 Ontologia de Fundamentação Unificada (*Unified Foundational Ontology* – UFO) 24

2.5 Ontologia de Processos Intensivos em Conhecimento (*Knowledge-intensive Processes Ontology* – KiPO) 25

3 Regras de Negócio 32

3.1 Classificação das regras de negócio 33

3.2 Representação de regras de negócio 36

3.3 Limitações da representação de regras de negócio 44

4 DecKiPO – Extensão da ontologia de regras de negócio para cenários de KiP 49

4.1 Proposta dos novos elementos 50

4.2 Restrições com LTL Declare 51

4.3 Níveis de cumprimento com SBVR 69

5 Avaliação e aplicabilidade da proposta 79

5.1 Cenários de estudo 79

5.2 Avaliação 83
5.3 Aplicabilidade da proposta 87
6 Conclusão 106
6.1 Contribuições desse trabalho 107
6.2 Trabalhos futuros 107
Referências Bibliográficas 109
Anexo I – Descrição dos conceitos da DecKiPO 114

Lista de Figuras

Figura 1 - Tipos de ontologia, traduzido de GUARINO (1998) 9
Figura 2 – Exemplo de Relator. Traduzido de (GUIZZARDI, 2005) 12
Figura 3 - Metamodelo da UFO-B 14
Figura 4 - Representação das relações de Allen (1983) 15
Figura 5 - Exemplo de modelo de processo imperativo com BPMN 17
Figura 6 – Exemplos de domínios classificados quanto a previsibilidade e repetitividade de seus processos (PESIC, 2008) 17
Figura 7 – Semânticas de alguns operadores LTL (PESIC, 2008) 19
Figura 8 – Restrições de Existência do LTL Declare e sua representação gráfica (PESIC, 2008) 19
Figura 9 – Restrições de Relação do LTL Declare e sua representação gráfica (PESIC, 2008) 22
Figura 10 – Restrições de Negativa de Relação do LTL Declare e sua representação gráfica (PESIC, 2008) 23
Figura 11 – Restrições de Escolha do LTL Declare e suas respectivas representações gráficas (PESIC, 2008) 24
Figura 12 – Extrato do Metamodelo da BPO na KiPO (FRANÇA, NETTO, *et al.*, 2014) 26
Figura 13 – Metamodelo da DO na KiPO (FRANÇA, NETTO, *et al.*, 2014) 28
Figura 14 – Metamodelo da CO na KiPO (FRANÇA, NETTO, *et al.*, 2014) 29
Figura 15 – Metamodelo da BRO na KiPO (FRANÇA, NETTO, *et al.*, 2014) 30
Figura 16 – Metamodelo da KiPCO na KiPO (FRANÇA, NETTO, *et al.*, 2014) 31
Figura 17 – Conceitos de modelagem de regras em três diferentes níveis de abstração (WAGNER, GIURCA e LUKICHEV, 2006) 33
Figura 18 – Tipos de proposições segundo a SBVR. Fonte: (OMG, 2017) 38
Figura 19 – Tipos de declaração das *Definitional Business Rules.* Fonte: (OMG, 2017) 40

Figura 20 – Estrutura das *Behavioral Business Rules* (OMG, 2017) 41
Figura 21 – A perspectiva da Ontologia de Regras de Negócio na KiPO 42
Figura 22 - Metamodelo da Regras de Reação (*Reaction Rule*) na KiPO 44
Figura 23 - Perspectiva da BRO com os novos conceitos da SBVR 51
Figura 24 – *Os templates* $existence_n(A)$, $absence_n(A)$ e $exactly_n(A)$ do LTL Declare como especializações de *Foundational Existence Rule* na DecKiPO 54
Figura 25 – Exemplo de representação do template e$xistence_n(A)$ na DecKiPO 55
Figura 26 – Modelo LTL Declare de um atendimento helpdesk 57
Figura 27 – Metamodelo da DecKiPO das *Foundational Relation Rules* 59
Figura 28 - Modelo Declare com restrições de escolha 65
Figura 29 - Modelo declarativo em LTL Declare para tratamento de pacientes (VAN DER AALST, PESIC e SCHONENBERG, 2009) 66
Figura 30 – Classes Choice e ExclusiveChoice na KiPO 68
Figura 31 – Modelo do cenário de tratamento médico 69
Figura 32 – Metamodelo dos Severity Levels na DecKiPO 72
Figura 33 - Extrato do processo de tomada de decisão na KiPO (FRANÇA, NETTO, *et al.*, 2014) 74
Figura 34 - Relação entre o Nível de Cumprimento Concebido (*Conceived Enforcement Level*) e os elementos da KiPO 75
Figura 35 - *Perceived Enforcement Level* mediando a consideração da Regra de Negócio Comportamental (*Foundational Behavioral Business Rule*) para tomada de Decisão (*Decision*) pelo Agente (*Agent*) 76
Figura 36 – Modelo do processo de liberação de crédito 82
Figura 37 – Modelagem da Regra-HD1 com a DecKiPO 88
Figura 38 – Modelagem da Regra-AV1 na versão atual da KiPO 90
Figura 39 – Modelagem da Regra-AV1e Regra-AV3 na versão atual da KiPO 91
Figura 40 – Modelo com a extensão DecKiPO da Regra-AV1e Regra-AV3 91
Figura 41 – Modelo da Regra-AV2 na versão atual da KiPO 92
Figura 42 – Modelo da Regra-AV2 com a extensão DecKiPO 93
Figura 43 – Modelo com a Regra-AV1, Regra-AV2 e Regra-AV3 utilizando o metamodelo estendido com a DecKiPO 93
Figura 44 – Modelo com a Regra-AV4 utilizando o metamodelo estendido com a DecKiPO 96

Figura 45 – Modelo em DecKiPO da tomada de decisão e restrições envolvendo a Regra-LC1 97
Figura 46 – Modelo em DecKiPO com a adição do nível de cumprimento concebido para a Regra-LC1 98
Figura 47 – Modelo em DecKiPO com a adição do nível de cumprimento pré-autorizado concebido para a Regra-LC1 99
Figura 48 – Modelo em DecKiPO da Regra-AV7 com o nível de cumprimento derivado da descrição da regra de negócio 100
Figura 49 – Modelo em DecKiPO da instância de execução do processo que viola a Regra-AV7 100
Figura 50 – Modelo LTL Declare gerado após a mineração das instâncias dos processos 102
Figura 51 – Extrato da OWL gerada pela ferramenta da DecKiPO 103
Figura 52 – Exemplo de aplicação da OWL gerada pela ferramenta de transformação da DecKiPO, com um grafo gerado a partir do Neo4J 104

Lista de Tabelas

Tabela 1 – Axiomatização das relações de Allen 15
Tabela 2 – Representação das instâncias de um processo e sua validade segundo as Relações de Restrição 23
Tabela 3 - Comparando linguagens de representação de regras de negócio. Adaptado de (Sull et. al., 2011) 45
Tabela 4 - Níveis de Cumprimento ordenados por gravidade (OMG, 2017) 70
Tabela 5 – Representação da associação dos elementos do LTL Declare com DecKiPO 102

1 Introdução

Um domínio pode ser representado através de seus conceitos básicos, a relação e organização hierárquica entre eles. Essa representação pode ser enriquecida com a exposição dos aspectos comportamentais desse domínio, especificando a dinâmica entre os conceitos e as restrições que regem seus comportamentos. Organizados e encadeados em uma ordem específica no domínio, a especificação destes aspectos comportamentais ganha significado mais amplo e passa a ser vistas como atividades mais complexas que tem o objetivo de produzir um resultado. As atividades, por sua vez, são encadeadas em fluxos de controle ainda mais complexos e extensos que agregam valor ao negócio, que são conhecidos como processos de negócio.

Em domínios complexos onde são produzidos diversos resultados distintos, os processos tornam-se numerosos e normalmente possuem pontos de interseção com outros processos, de modo que passam a existir dependências entre as atividades que os compõem e, dessa forma, a necessidade de se orquestrar e monitorar sua execução para garantir a produção dos resultados esperados.

O Gerenciamento de Processos de Negócio, da expressão em Inglês *Business Process Management* (BPM), adquiriu valor estratégico em organizações na obtenção dos objetivos organizacionais, otimização de recursos e redução de custos (BPTRENDS, 2016). Esses processos de negócio são comumente representados através de um paradigma estruturado, seguindo boas práticas bem conhecidas, e modelados a partir da experiência e maturidade da organização na execução de seus processos. Devido à natureza desses tipos de processos, linguagens tradicionais de modelagem de processos (como BPM) expressam os fluxos de atividade críticas e alternativas, com pouca margem para flexibilidade (DI CICCIO, MARRELLA e RUSSO, 2015).

Mais recentemente, entretanto, organizações têm direcionado seus esforços de gerenciamento para processos de negócio que estão sendo considerados críticos para seu planejamento estratégico. Esses processos envolvem tomadas de decisão complexas, não seguem uma sequência previsível e repetível, envolvem interação e colaboração entre as

partes interessadas e se apoiam fortemente no conhecimento e experiência de seus agentes. Esse tipo de processo se adequa à definição de Hagen (HAGEN, RATZ e POVALEK, 2005), que os descreve como fluxos de atividades imprevisíveis, com mudanças frequentes entre instâncias distintas, sendo também conhecidos como Processos intensivos em Conhecimento (do Inglês, *Knowledge-intensive Processes - KiPs*).

Processos intensivos em Conhecimento englobam um grande número de elementos que envolvem a tomada de decisão e outros aspectos críticos na implementação de processos não-estruturados, abrangendo as intenções e crenças de agentes que executam ações orientadas a objetivos, a colaboração entre os agentes envolvidos no processo e as regras de negócio que refletem as restrições em possíveis fluxos de execução de atividades do processo (HAGEN, RATZ e POVALEK, 2005). Essas regras especificam critérios diversos na execução dos fluxos de negócio e delimitam as possibilidades de ação dos agentes no domínio, restringindo e determinando as escolhas possíveis em situações apresentadas, reações do domínio a eventos ocorridos e consequências à violação de restrições impostas. Esse tipo de processo não é facilmente modelado nas abordagens imperativas tradicionais, visto que a quantidade de desvios necessários para a correta representação do processo, bem como a impossibilidade de representação de desvios não programados, torna a utilização desta abordagem difícil de ser adotada em cenários de Processos Intensivos em Conhecimento.

Na literatura recente, a abordagem baseada em restrições, conhecidas como abordagem declarativa, tem sido reconhecida como mais adequada para a representação de processos não estruturados, uma vez que focam na modelagem de restrições impostas por regras de negócio em detrimento ao sequenciamento dos fluxos de atividade.

Entretanto, Processos intensivos em Conhecimento têm também como características a execução de atividades complexas, que envolvem tomadas de decisão e colaboração entre agentes do processo, elementos estes que normalmente não são representados em modelos de processos de negócio, ou nas linguagens que adotam o paradigma de modelagem declarativa.

Dessa forma, a Ontologia de Processos intensivos em Conhecimento (*Knowledge-intensive Process Ontology - KiPO*) (FRANÇA, NETTO, *et al.*, 2014) foi

proposta com o objetivo de identificar e conceituar os aspectos relevantes envolvidos na descrição de um KiP, abrangendo desde aspectos de fluxo de atividades até a representação de interações entre agentes e os processos mentais que os levaram a tomar decisões em um processo.

Processos intensivos em Conhecimento têm como um de seus pilares a imposição de restrições baseadas em regras de negócio. Uma ontologia que se proponha a representar KiPs precisa ser capaz de prever e relacionar quais regras de negócio restringem a execução do processo e como elas se relacionam entre si e entre os demais elementos do processo.

1.1 Problema

O problema a ser tratado no presente trabalho é a ausência de representatividade na ontologia de Processos intensivos em Conhecimento das relações temporais impostas por regras de negócio a elementos de KiP.

1.2 Hipótese e Proposta de Solução

A hipótese desta pesquisa pode ser descrita como a seguir.

***Hipótese**: Ao estender a Ontologia de Processos intensivos em Conhecimento (KiPO), em particular os conceitos da sub-ontologia de regras de negócio (a BRO), será possível representar as restrições e desvios necessários a um maior entendimento de processos intensivos em conhecimento.*

Em particular, supõe-se que, desenvolvendo os elementos que traduzem e representem restrições temporais entre regras de negócio (aspecto principal do paradigma de modelagem declarativa), em especial as relações descritas pela linguagem LTL Declare, será possível representar ainda mais relacionamentos entre os elementos de KiP e aumentar a capacidade representativa do modelo de um processo intensivo em conhecimento. Como resultado, a KiPO se tornará uma ontologia mais completa e compreendendo semântica mais expressiva para representação de KiPs.

Para prover a semântica necessária à representação dos relacionamentos temporais da lógica temporal linear (LTL) mantendo-se a consistência lógica com as definições formais dos construtos da ontologia de fundamentação UFO, na presente

proposta foram adotadas as relações de intervalo de tempo (ou *Time Interval Relations*) descritas na UFO-B (*before, meets, overlaps, starts, during, finishes, equals*) e representadas pelas chamadas Relações de Allen (1983).

Essas relações não foram exploradas na BRO de LOPES (2010), limitando essa ontologia a descrever regras de negócio sem considerar os intervalos de tempo decorrentes entre os eventos que elas limita. Essa relações, portanto, podem ser utilizadas para prover uma representação semanticamente mais precisa dos aspectos temporais entre regras de negócio, uma vez que as Relações de Allen adicionam a semântica necessária para expressar os estados de pré-execução e de pós-execução de um evento, conforme descrito.

Adicionalmente, na presente proposta foram descritas novas classes que acrescentam à ontologia a semântica para permitir descrever a variabilidade no cumprimento das restrições, estendidas a partir dos conceitos de níveis de cumprimento (ou *enforcement levels*) descritos pela SBVR (OMG, 2017).

Por fim, mas não menos importante, a proposta desta dissertação considerou a adição de uma nova camada de informação a respeito de instâncias de um KiP que se relacionam a conceitos de outras sub-ontologias da KiPO, como crenças e sentimentos de um agente. Níveis de cumprimento são descritos como meta-propriedades de outros conceitos do metamodelo, tornando possível a representação da flexibilidade na execução de instâncias no que diz respeito a desvios de restrições impostas ao domínio.

1.3 Objetivo

O objetivo deste trabalho é propor uma conceituação, na forma de uma ontologia, de regras de negócio em cenários de processos intensivos em conhecimento, de forma a permitir uma maior expressividade semântica da ontologia, bem como uma maior capacidade de representação dos processos de negócio inerentes a esse tipo de domínio, principalmente nos aspectos de relação temporal dos eventos restritos por estas regras de negócio.

Para atingir esse objetivo geral, são considerados como objetivos específicos: (i) a fundamentação dos *templates* LTL Declare na UFO; (ii) a redefinição da taxonomia dos tipos de regras de negócio; (iii) a fundamentação dos níveis de cumprimento prescritos e

percebidos no contexto da execução de um processo intensivo em conhecimento; e (iv) a utilização de cenários de estudo diversos para realização de uma análise qualitativa de critérios de avaliação da ontologia proposta.

Como resultados esperados, pretende-se expandir a capacidade de representação da Ontologia de Regras de Negócio (BRO), bem como da Ontologia de Processos intensivos em Conhecimento (KiPO), tornando-a mais completa e apta a servir como metamodelo para a modelagem de processos intensivos em conhecimento no que diz respeito à representação de regras de negócio.

1.4 Metodologia científica

O método de pesquisa utilizado o *design science*, do qual a partir da avaliação ontológica e da análise qualitativa de cenários de estudo oriundos de três fontes distintas com a aplicação da proposta, são analisadas as extensões de conceitos de regras de negócio propostas neste trabalho e sua contribuição para a hipótese definida. A avaliação qualitativa foi escolhida devido à subjetividade inerente às características de qualidade de uma ontologia.

A variável dependente desse trabalho é a qualidade semântica dos modelos gerados através da proposta, tratando diversos critérios de avaliação ontológica propostos por VRANDEčIć (2010). Utilizando a abordagem de avaliação através de corpus de dados, são desenvolvidos diversos modelos de processos utilizando a linguagem UML para representação dos conceitos propostos nesse trabalho.

Os cenários de estudo avaliados são oriundos de fontes de dados de três domínios distintos: atendimento a *helpdesk*, concessão de empréstimos e locação de veículos. O primeiro cenário de estudo contém regras de negócio e atividades extraídas de uma base de dados contendo o log de eventos dos atendimentos no ano de 2016 a clientes realizados por uma empresa, sediada no Rio de Janeiro, que presta serviços de infraestrutura de TIC para seus clientes; o segundo cenário de estudo contém regras de negócio e atividades oriundas da base de dados fornecida pelo *Business Process Intelligence Challenge* (BPIC) e extraídas dos artigos vencedores do desafio no ano de 2017 (BLEVI, DELPORTE e ROBBRECHT, 2017); o terceiro cenário de estudo contém regras de negócio oriundas da

especificação da *Semantics of Business Vocabulary and Rules* (SBVR) que contém a descrição completa de um domínio de locação de veículos (OMG, 2016).

1.5 Organização do Texto

Este trabalho é divido em seis capítulos. O segundo capítulo apresenta os conceitos sobre metamodelos e ontologias, elencando as bases de fundamentação deste trabalho, seguido do terceiro capítulo que aborda os conceitos relacionados a regras de negócio e suas linguagens de representação. O quarto capítulo apresenta a formalização da proposta, descrevendo todos os conceitos propostos para a extensão da representação de regras de negócio em ontologias. O quinto capítulo discute a avaliação ontológica e aplicabilidade da proposta com base nos critérios de qualidade definidos. Por fim, o sexto e último capítulo apresenta a conclusão do trabalho.

2 Sobre Paradigmas de modelagem de processos de negócio, Ontologia e Metamodelos

MYLOPOULOS (1992) define a disciplina de modelagem conceitual como a atividade de descrever formalmente alguns dos aspectos do mundo físico e social que nos cerca com a finalidade de compreensão e comunicação. Muitas áreas de conhecimento utilizam modelos conceituais para fornecer uma compreensão do domínio estudado e garantir que seus conceitos sejam entendidos por qualquer especialista da área.

No domínio de Sistemas de Informação, por exemplo, modelos conceituais fornecem uma visão estrutural e comportamental de um *software* sendo desenvolvido; em cenários onde processos de negócio precisam ser descritos, modelos de processo fornecem a gestores e agentes do processo os fluxos de atividades que devem ser seguidos e os possíveis desvios. Uma vez que um modelo descreve idealmente e de forma fidedigna o domínio específico para o qual foi desenvolvido, um mesmo modelo de processo de negócio criado para uma área de Recursos Humanos de uma empresa A não necessariamente descreverá os processos de negócio executados pela área de Recursos Humanos da empresa B.

Esses modelos representam conceitos que descrevem cada um dos elementos utilizados para modelar um domínio e como eles se relacionam. MELLOR et al. (2004) afirma que metamodelo é um modelo de uma linguagem de modelagem, que define a estrutura, semântica e restrições para uma família de modelos. Para a OMG (2005), um metamodelo é um tipo especial de modelo que especifica a sintaxe abstrata de uma linguagem de modelagem, podendo ser entendida como a representação das classes de todos os modelos expressos por aquela linguagem. Para GUIZZARDI (2007), um metamodelo é a descrição de uma linguagem que define um conjunto de construtos selecionados com a finalidade de executar uma tarefa específica e um conjunto de regras para combinar esses construtos a fim de criar modelos gramaticalmente válidos na linguagem.

Na área de sistemas de informação, por exemplo, a UML é uma linguagem cujo metamodelo provê os conceitos necessários à criação das representações estruturais e comportamentais de um *software*, como diagramas de classes, casos de uso e sequência. Já o metamodelo subjacente à notação BPMN (*Business Process Model and Notation*) fornece os elementos necessários para a modelagem de processos de negócio, definindo conceitos como atividades, eventos e fluxos de controle. Ambos os metamodelos provem um conjunto de elementos a partir dos quais modelos podem ser construídos. Em geral, metamodelos são desenvolvidos para atender às necessidades de modelagem de suas áreas de interesse. Enquanto metamodelos são uma representação concreta de um domínio sob a ótica de uma linguagem, a conceituação desse domínio é formalizada através de uma estrutura conhecida como ontologia.

Segundo GRUBER (1993), ontologia é "uma especificação explícita de uma conceituação". A definição foi revista por BORST (1997) em sua tese de PhD, que a definiu como uma especificação formal de uma conceituação compartilhada. STUDER et al. (1998) juntaram ambas as definições, chegando a uma mais completa: uma ontologia é uma especificação explícita e formal de uma conceituação compartilhada.

Devido à versatilidade de aplicações de ontologias, sendo utilizadas para descrever diferentes domínios em diferentes áreas de conhecimento (inteligência artificial, banco de dados, engenharia de software, etc), USCHOLD e JASPER (1999) definem que uma ontologia pode assumir uma variedade de formas mas precisa, necessariamente, possuir um vocabulário de termos e alguma especificação de seu significado, o que inclui a definição dos termos e como eles estão inter-relacionados. GUIZZARDI (2007) define uma ontologia como uma teoria sobre os tipos de entidades que devem ser admitidas em um domínio.

GUARINO (1998) apresenta quatro classificações de ontologias, de acordo com o nível de abstração que ela se propõe a descrever:

- **Ontologias de alto nível (ou de fundamentação)** descrevem conceitos muito generalistas, como tempo e espaço, e independem de um domínio em particular;

- **Ontologias de domínio** descrevem o vocabulário relacionado a um domínio de conhecimento, como medicina ou automotivo, especializando os conceitos apresentados na ontologia de alto nível;
- **Ontologias de tarefa** descrevem o vocabulário relacionado a uma atividade ou tarefa genérica, que pode ocorrer em diversos domínios;
- **Ontologias de aplicação** descrevem conceitos inerentes a um determinado domínio e tarefa, especializando conceitos oriundos das duas ontologias.

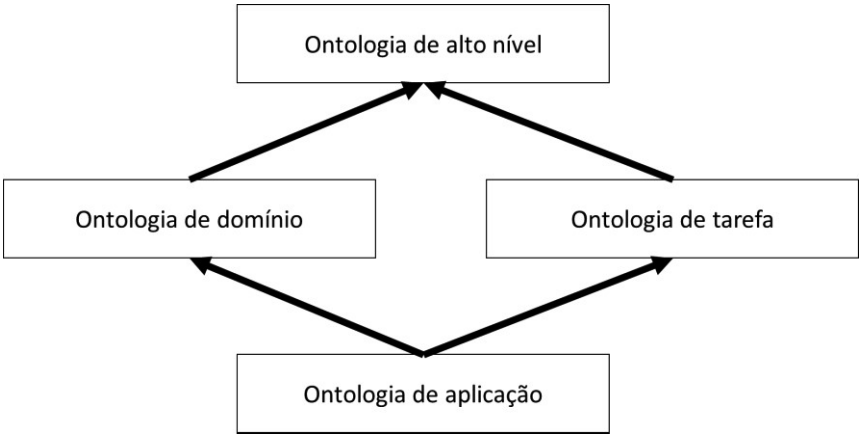

Figura 1 - Tipos de ontologia, traduzido de GUARINO (1998)

A representação de regras de negócio, por compreender um conhecimento rico e amplo sobre um universo de discurso, tanto sob a perspectiva estrutural quanto sob a perspectiva dinâmica, também deve ter as mesmas características de qualidade que qualquer modelo conceitual, e que, portanto, deveria poder representar as regras de forma precisa. Com o uso de uma ontologia de fundamentação (também denominada como ontologia de alto nível na Figura 2), a riqueza semântica proporcionada faz com que a representação conceitual de Regras de Negócio seja mais precisa.

Portanto, a proposta deste trabalho utiliza como base a UFO (GUIZZARDI, 2005), que é uma ontologia de fundamentação que apresenta um conjunto de construtos que serão a base para a proposta de representação de Regras de Negócio em um modelo conceitual que seja mais preciso.

2.1 UFO, uma ontologia de fundamentação

2.1.1 UFO-A, uma ontologia de objetos

Dois dos conceitos principais tratados pela UFO são *Universals* (Tipo) e *Individuals* **(**Particular). *Universals* são padrões de características que podem ser instanciados em qualquer número de *Individuals*, que por sua vez são entidades que possuem identidade única e existem no mundo real, instanciados a partir de um *Universal* (GUIZZARDI, FALBO e GUIZZARDI, 2008).

Universals são compostos de elementos classificados em *endurant* e *perdurant*. *Endurants* mantém sua identidade ao longo do tempo enquanto *perdurants* são caracterizados por um acontecimento em determinado momento, possuindo identidade apenas naquele espaço de tempo definido. Um automóvel é um exemplo de *endurant* dado que sempre manterá suas características ao longo do tempo; Uma viagem de carro é um exemplo de *perdurant* pois é possível identificar um intervalo entre o início e o término do evento, indicando passagem de tempo.

A UFO-A trata de objetos, portanto *endurants*. Deste ponto em diante, os conceitos tratados estão relacionados diretamente a *Individuals*, sendo os *Universals* mencionados apenas quando necessário. Um *endurant* pode ter características de um *substantial* ou de um *moment*.

Substantials são *individuals* existencialmente independentes, como o automóvel do exemplo anterior. Ao contrário, *Moments* são *individuals* que só existem se inerentes a outros *individuals*. GUIZZARDI (2008) apresenta como exemplos de *moments* uma carga elétrica, que é inerente a um condutor, um sintoma, que é inerente a uma doença, ou uma cor que é inerente a um automóvel.

Moments podem ser existencialmente dependentes de um único *individual* ou de vários deles. Quando a dependência existencial é com apenas um *individual*, ela é classificada na UFO-A como sendo um *Intrinsic Moment*. Voltando ao exemplo anterior, a cor de um automóvel é existencialmente dependente apenas do próprio automóvel, não dependendo de nenhum outro indivíduo para existir no mundo real. Essa percepção da cor de um automóvel é uma objetificação de um atributo de um objeto que pode ser projetada diretamente em um certo grupo ou faixa de valores pré-determinados (ALBUQUERQUE e GUIZZARDI, 2013). Essa projeção é classificada na UFO-A como

um *Quality* cujo valor é projetado a partir de um elemento conhecido como *Quale* na UFO-A.

Quales, por sua vez, são percepções ou concepções de um *intrinsic moment* que podem ser representados como um ponto em uma escala de valores pré-definidos. Para entender corretamente o conceito de *Quality* e *Quale*, é necessário primeiro entender o conceito de *quality structures*. GUIZZARDI (2008) explica que uma tentativa de modelar a relação de *Intrinsic Modes* e sua representação em estruturas cognitivas humanas é apresentada na Teoria de Espaços Conceituais introduzida em (GÄRDENFORS, 2000), que se baseia na noção de *quality structures* que delimita os valores possíveis, dentro da cognição humana, de uma propriedade de um objeto.

Dessa forma, podemos dizer que um automóvel possui uma cor (azul, por exemplo) que por sua vez é a percepção de um valor em possível em uma estrutura de qualidade (a escala RGB, por exemplo). A cor, por sua vez, é um *individual* instanciado a partir de um *universal* que descreve as propriedades dessa instância. Assim, podemos dizer que o *quality individual* **cor** é uma instância do *quality universal* **Cor**, que por sua vez está associada com um *quality structure* que possui todos os valores da **escala RGB** e cuja cor **azul** é um ponto percebido (*quale*) nessa escala que caracteriza a *quality individual* **cor** do **automóvel**.

É importante destacar que a escala descrita por um *quality structure* pode ser quaisquer conjuntos de valores unidimensionais ou multidimensionais, desde o conjunto de números reais até uma escala enumeradora de atributos qualitativos que possuam algum significado na cognição humana (ALBUQUERQUE e GUIZZARDI, 2013). Esses conceitos serão relevantes para a formalização dos níveis de cumprimento de regras de negócio discutidos no Capítulo 4.

Até o momento, foram descritas as caraterísticas de um *moment* que é dependente existencialmente de apenas um *individual*. Quando um *moment* depende de uma relação de diversos *individuals* para existir, ela é classificada na UFO-A como um *Relator*. Por exemplo, um aluguel de automóvel só passa a existir como instância do mundo real, ou seja, ser materializada, ao relacionar um automóvel a um locatário; um tratamento médico só passa a existir no mundo real quando relaciona um paciente a uma unidade médica, conforme ilustra a Figura 2.

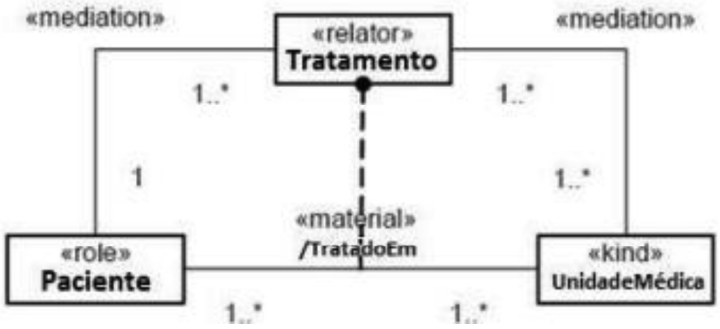

Figura 2 – Exemplo de Relator. Traduzido de (GUIZZARDI, 2005)

Relators, portanto, são entidades que aglutinam outras entidades. Essas relações podem ser formais (*formal relators*) ou materiais (*material relators*). *Formal relators* são o tipo de relação que ocorre entre dois ou mais entidades diretamente, sem outro indivíduo intervindo (GUIZZARDI, FALBO e GUIZZARDI, 2008). Esse tipo de relação normalmente é encontrada na formalização de dependências entre entidades em um metamodelo, como as definições de especialização, agregação e composição entre classes presentes na UML. Já os *material relators* possuem uma materialização que possui características próprias e identifica a relação estabelecida, como é o caso do exemplo descrito na Figura 2. Um tratamento possui atributos próprios e é identificado como uma entidade do mundo real passível de entendimento em um domínio. Novamente, estes conceitos serão importantes na formalização das extensões propostas no Capítulo 4.

Por fim, enquanto *Substantials* e *Moments* caracterizam individuals que são existencialmente independentes e dependentes, respectivamente, uma terceira classificação de *endurant,* a *Situation* é constituída por vários outros objetos que definem uma porção da realidade (estado das coisas ou *state of affairs*) que pode ser compreendida como um todo. Uma *situation* possui uma relação específica nomeada de "estar presente em" entre os objetos e outras *situations* que a compõem. Por exemplo, pode-se dizer que o *substantial* Carro A e seu *moment* azul está presente na *situation* "O Carro A é azul".

2.1.2 UFO-B, uma ontologia de eventos

A UFO-B formaliza conceitos relativos a eventos, ou *perdurants*, cuja principal distinção em relação a *endurants* é que eventos possuem sua existência definida a partir de um intervalo de tempo definido. Em (GUIZZARDI, FALBO e GUIZZARDI, 2008), temos que um evento é um *individual* composto de partes temporais que se acumulam a medida em que acontecem no tempo. Por ser composto de instantes de tempo, *perdurants* não

mantém sua identidade ao longo do tempo sendo definidos apenas em função do tempo em que existem.

Um evento será chamado de *Complex Event* enquanto puder ser decomposto em outros eventos menores, até o ponto em que se atinja o evento mais básico chamado de *Atomic Event*. Eventos, porém, não são entidades isoladas, sendo existencialmente dependentes de objetos (*endurants*) para que possam existir. Esses objetos são considerados participantes do evento de forma que *complex events* sempre serão compostos de dois ou mais participantes. Essa dependência existencial define as propriedades espaciais do evento pois os objetos são materializações no mundo real de conceitos universais.

Em contraste, todas as propriedades temporais de um objeto são definidas em termos dos eventos dos quais eles são participantes (GUIZZARDI *et al.*, 2013). A UFO-B assume que o espaço conceitual de tempo é uma estrutura composta de intervalos temporais (*time intervals*) que, por sua vez, são compostos de instantes (*time point*). Instantes podem ser representados como números reais e intervalos temporais como conjuntos de números reais. O atributo tempo de um evento, portanto, é um *quality* cujo valor é um ponto definido em uma *quality structure* podendo assumir quaisquer escalas de passagem de tempo conhecidas.

A UFO-B formaliza como uma das possíveis estruturas temporais a relação linear entre *time points*, utilizando como base de definição as relações de Allen proposta em (ALLEN, 1983). Nessa estrutura, cada evento é associado a um ponto de início (*begin-point)* outro de término (*end-point*), sendo esses pontos ordenados por uma relação de precedência (*precedes*). A Figura 3 ilustra o metamodelo da UFO-B extraída de (MAURO *apud* MARTINS, 2009)

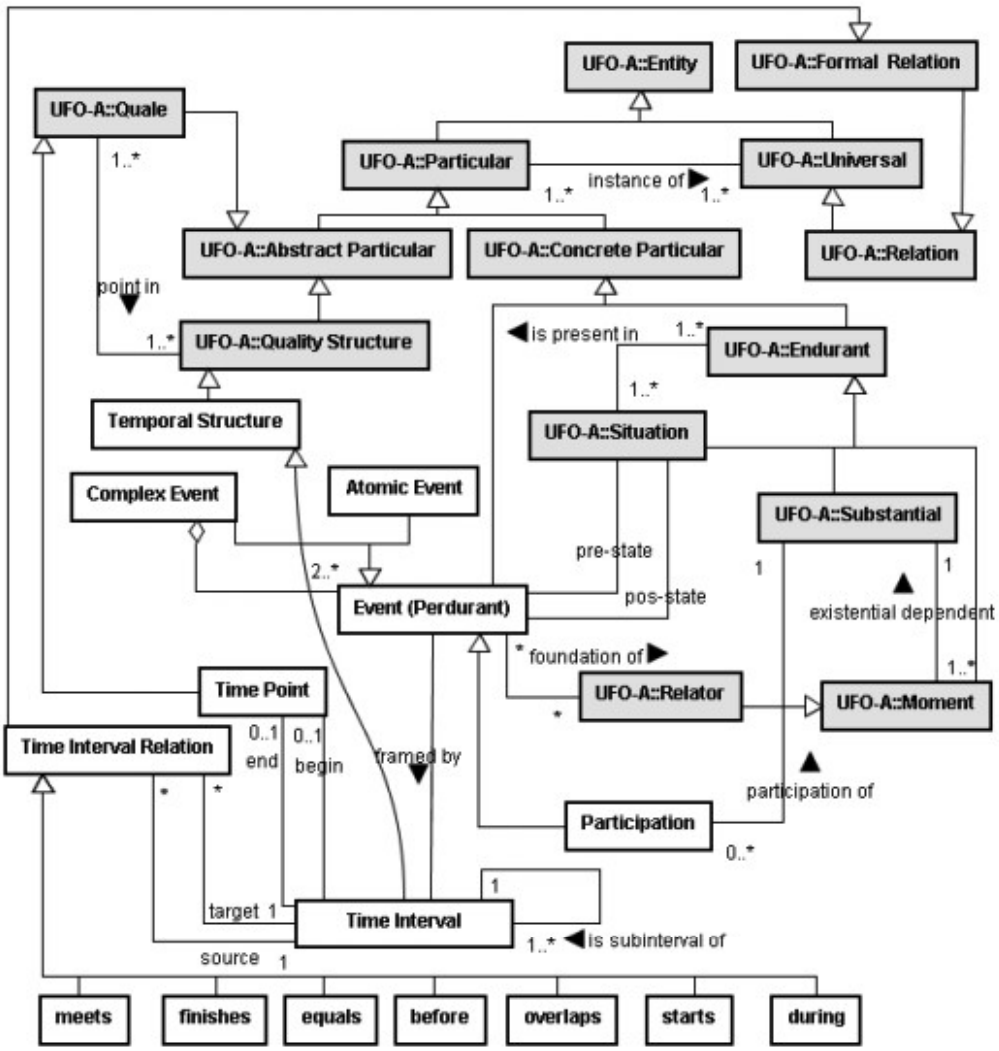

Figura 3 - Metamodelo da UFO-B

Considerando a passagem linear de tempo, podemos considerar que os intervalos de Allen descrevem a relação de um evento **e** em relação ao evento **e'**. Sendo assim, a relação *meets* (encontra) denota que o evento **e'** inicia imediatamente após o evento **e**; a relação *finishes* (termina) denota que o evento **e'** termina juntamente com o evento **e**; a relação *equals* (igual) denota que o início e término do evento **e'** coincidem com o início e o término do evento **e**; a relação *before* (antes) denota que o evento **e'** termina antes do início do evento **e**; a relação *overlaps* (sobrepõe) denota que o início do evento **e'** é anterior ao início do evento **e** e seu término é posterior ao início do evento **e**; a relação *starts* (inicia) denota que o evento **e'** inicia juntamente com o evento **e**; por fim, a relação *during* (durante) denota que o início e o término do evento **e'** se dará durante a realização do evento **e**. A Figura 4 ilustra graficamente cada uma das relações de Allen.

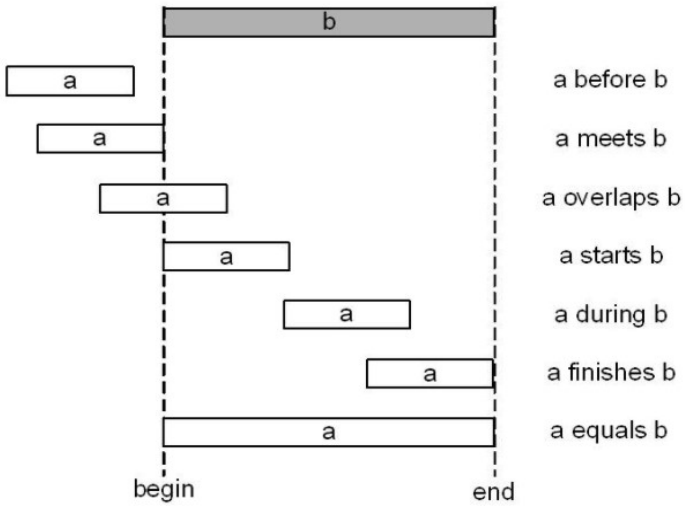

Figura 4 - Representação das relações de Allen (1983)

Em (GUIZZARDI, WAGNER, *et al.*, 2013), são apresentadas as axiomatizações das relações de Allen, que será a base para a formalização das extensões propostas nesse trabalho. Os **Axioma A1** ao **Axioma A8** descritos na Tabela 1 foram transcritos de (GUIZZARDI, WAGNER, *et al.*, 2013):

Tabela 1 – Axiomatização das relações de Allen

Axioma A1	∀e:Event precedes(begin-point(e), end-point(e))
Axioma A2	∀e,e':Event before(e,e') ↔ precedes(end-point(e), begin-point(e'))
Axioma A3	∀e,e':Event meets(e,e') ↔ (end-point(e) = begin-point(e'))
Axioma A4	∀e,e':Event overlaps(e,e') ↔ precedes(begin-point(e), begin-point(e')) ∧ precedes(begin-point(e'), end-point(e)) ∧ precedes(end-point(e), end-point(e'))
Axioma A5	∀e,e':Event starts(e,e') ↔ (begin-point(e) = begin-point(e')) ∧ precedes(end-point(e), begin-point(e'))
Axioma A6	∀e,e':Event during(e,e') ↔ precedes(begin-point(e'), begin-point(e)) ∧ precedes(end-point(e), begin-point(e'))
Axioma A7	∀e,e':Event finishes(e,e') ↔ precedes(begin-point(e'), begin-point(e)) ∧ (end-point(e) = begin-point(e'))
Axioma A8	∀e,e':Event equals(e,e') ↔ (begin-point(e) = begin-point(e')) ∧ (end-point(e) = begin-point(e'))

Eventos alteram uma *Situation*, intervindo no estado das coisas do domínio sendo representado. A relação entre *perdurants* e *endurants* advêm dessa intervenção nas Situações do domínio, onde no mínimo um *endurant* está presente, denotado pelo relacionamento "*is present in*", na Figura 3. Um *perdurant* altera um *pre-state* para um *pos-state*.

2.2 Paradigma imperativo e declarativo

Mapear e descrever processos de negócio em uma organização é uma tarefa difícil e envolve profissionais tanto de áreas de negócio quanto especialistas que dominem as boas práticas de mapeamento desses processos. Por muitos anos, processos de negócio eram modelados utilizando fluxogramas que continham elementos que representavam cada etapa do processo.

Esse tipo de modelagem obrigava os profissionais da área a representar, passo-a-passo, todas as atividades, desvios e saídas de um processo. Essa abordagem de modelagem é conhecida na literatura como **imperativa**. Ela presume que sejam descritos os procedimento passo-a-passo que devem ser seguido durante a execução do processo por um agente (PESIC, SCHONENBERG e VAN DER AALST, 2007) e não permite que sejam tomados caminhos não previstos no modelo.

Em processos altamente estruturados, cuja previsibilidade e repetitividade são muito altas, como por exemplo processos de requisição de viagens, um modelo de processo imperativo pode ser facilmente adotado para representar todos os caminhos possíveis de se seguir no processo, conforme ilustrado no modelo em BPMN na Figura 5. A *Business Process Modeling Notation* (BPMN) é uma das notações mais utilizadas atualmente por executivos de diversas áreas de negócio para modelar seus processos, mas existem outras linguagens e notações como EPC, Redes de Petri e diagramas de atividades.

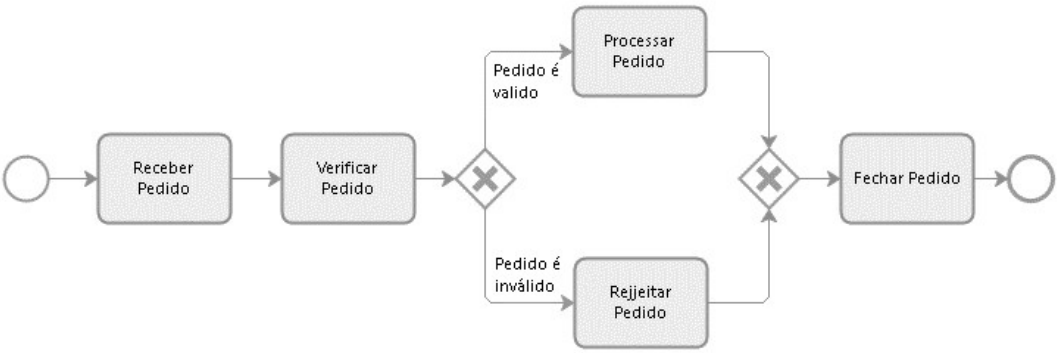

Figura 5 - Exemplo de modelo de processo imperativo com BPMN

Entretanto, em alguns domínios os processos de negócio executados não são previsíveis, ou são pouco repetíveis. Domínios como os que envolvem tratamento médico, resposta a desastres ou controle de tráfego aéreo são guiados por protocolos diversos que governam as restrições e fronteiras de ação de cada agente em cada instância do processo, ao invés de um conjunto de procedimentos passo-a-passo, prevendo todos os cenários possíveis. Isso porque as ações que devem ser tomadas a cada instância do processo variam enormemente de acordo com diversos fatores que não são necessariamente estruturados.

Processos de resposta a desastres naturais, por exemplo, são altamente imprevisíveis quanto à sequência de ações que devem ser tomadas em uma determinada situação. Ainda, um conjunto de ações executadas a partir da ocorrência de um determinado evento não necessariamente poderá ser repetido em outros, tornando esse tipo de processo muito pouco previsível quanto repetível. A Figura 6 ilustra exemplos de processos classificados quanto à previsibilidade e repetitividade, extraído de (PESIC, 2008)

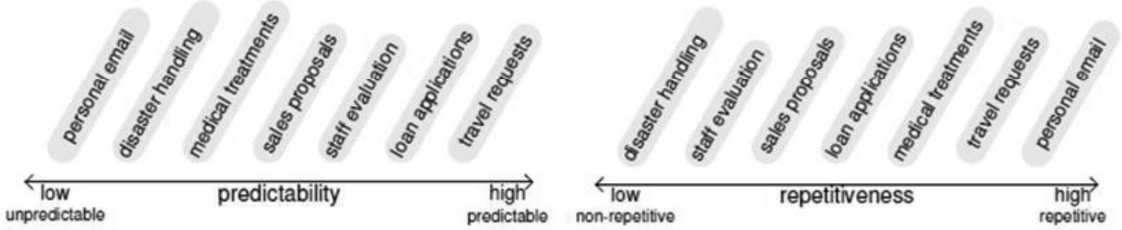

Figura 6 – Exemplos de domínios classificados quanto a previsibilidade e repetitividade de seus processos (PESIC, 2008)

O paradigma de modelagem baseada em restrições é conhecido na literatura como **declarativo** onde, ao invés da modelagem definindo o passo-a-passo de execução de cada

atividade do processo, são definidas as restrições que governam o domínio e os limites e condições para execução das atividades (PESIC, SCHONENBERG e VAN DER AALST, 2007).

2.3 A linguagem LTL Declare

Para criar modelos de processos utilizando essa abordagem, a linguagem LTL Declare é uma das principais ferramentas para modelagem de processos, amplamente adotada pela literatura (VAN DER AALST, PESIC e AND SCHONENBERG, 2009). LTL (sigla para Lógica Temporal Linear) é um tipo especial de lógica utilizada para descrever sequencias de transições entre estados em sistemas reativos, desconsiderando o tempo de maneira explícita, mas referenciando-o como comportamento modal, ou seja, com a expectativa de ocorrência de um evento (*eventually*) ou a certeza de que ele sempre ocorrerá (*always*) (PESIC, 2008).

PESIC (2008) descreve que a aplicação de LTL presume que não existe fim na execução de atividades governadas por esse tipo de lógica, como ocorre por exemplo em sistemas controladores de sinais de trânsito. Para que a LTL funcione em cenários de processos de negócio, onde em algum momento a execução será finalizada, PESIC propôs uma variação na qual chamou de "*business process LTL*" (PESIC, 2008), onde cada elemento da execução do processo corresponde a apenas um evento.

A semântica dos operadores modais da LTL é definida a seguir e ilustrada na Figura 7.

- Os operadores E ($\wedge$), OU ($\vee$) e NÃO (!) são operações da lógica de primeira ordem, representando conjunção, disjunção e negação, respectivamente;
- O operador *always* (sempre) ($\Box p$) especifica que p deve ser verdadeiro durante todo o fluxo do processo;
- O operador *eventually* (eventualmente) ($\Diamond p$) especifica que p irá ser verdadeiro pelo menos uma vez no fluxo do processo;
- O operador *next* (próximo) ($\bigcirc p$) especifica que p deve ser verdadeiro no próximo elemento no fluxo do processo.

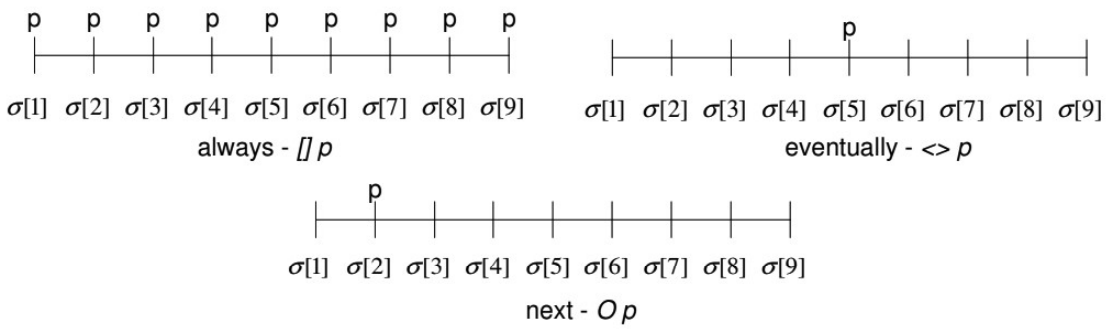

Figura 7 – Semânticas de alguns operadores LTL (PESIC, 2008)

A LTL Declare descreve quatro grupos de restrições que representam relações entre atividades: Restrições de Existência, Restrições de Relação, Restrições de Negação de Relação e Restrições de Escolha (PESIC, 2008). Cada uma dessas relações possui um conjunto de modelos chamados de *templates*, que são fórmulas descritas em Lógica Temporal Linear (LTL) e definem diferentes tipos de restrições a eventos do processo.

As **Restrições de Existência** delimitam a necessidade de existência de instâncias da atividade em um processo de negócio. PESIC (2008) define cinco *templates*: *existence*, *absence*, *exactly*, *init* e *last*. O *template* $existence_n(A)$ caracteriza a necessidade de existência de pelo *n* instâncias do evento A; o *template* $absence_n(A)$ limita a existência de, no máximo, *n* instâncias do evento A; o *template* $exactly_n(A)$ aponta a necessidade de existência de exatamente *n* instâncias do evento A; o *template init(A)* determina o evento pelo qual o processo é iniciado; o *template last(A)* determina o último evento que deve ser executado no processo.

A Figura 8 ilustra as fórmulas de cada um dos *templates* das Restrições de Existência do LTL Declare.

name of formula	LTL expression
$existence(A)$	$\Diamond(A, t_c)$
$existence2(A)$	$\Diamond((A, t_c) \wedge \bigcirc(existence(A)))$
$existence3(A)$	$\Diamond((A, t_c) \wedge \bigcirc(existence2((A))))$
…	…
$existence_N(A)$	$\Diamond((A, t_c) \wedge \bigcirc(existence_{N-1}(A)))$
$absence2(A)$	$!existence2(A)$
$absence3(A)$	$!existence3(A)$
…	…
$absence_N(A)$	$!existence_N(A)$
$exactly1(A)$	$existence(A) \wedge absence2(A)$
$exactly2(A)$	$existence2(A) \wedge absence3(A)$
…	…
$exactly_N(A)$	$existence_N(A) \wedge absence_{N+1}(A)$
$init(A)$	$((A, t_s) \vee (A, t_x))W(A, t_c)$

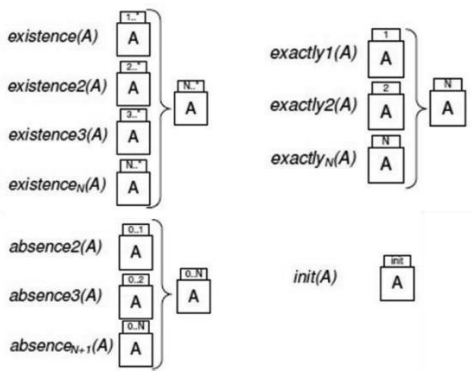

Figura 8 – Restrições de Existência do LTL Declare e sua representação gráfica (PESIC, 2008)

As **Restrições de Relação** formalizam os relacionamentos entre eventos do processo de negócio definindo, principalmente, relações temporais entre eles. Existem dois grupos de *templates* que representam as Restrições de Relação: os *templates* não ordenados e os ordenados.

Os *templates* não ordenados são aqueles que restringem a relação entre dois eventos do processo, mas não se preocupam com a ordem em que eles ocorrem. Dentre eles, o *template responded existence(A,B)* define que, se uma atividade A é executada, então a atividade B também é executada, antes ou depois de A. O *template co-existence(A,B)* descreve que se uma atividade A é executada, a atividade B também é executada e vice-versa.

Já os *templates* ordenados definem uma restrição temporal entre as atividades. Esse grupo de *templates* possui nove fórmulas, das quais seis delas são derivadas de três fórmulas mais fundamentais: *response*, *precedence* e *succession*. O *template response(A,B)* define uma restrição onde uma atividade B sempre vai ocorrer depois da atividade A ter sido executada; o *template precedence(A,B)* define uma restrição onde uma atividade A sempre ocorreu antes de uma atividade B; o *template succession(A, B)* define que, sempre que uma atividade A ocorreu, a atividade B ocorreu eventualmente depois e a atividade A sempre foi executada antes da atividade B.

Para melhor entendimento, considere as atividades (A) correr, (B) beber água e (C) comer. Utilizando essas três atividades simples, serão ilustrados a seguir alguns exemplos dos três *templates* de Restrição de Relação explicados:

- *Template response(A,B)*: É possível somente (B) Beber Água mas, se (A) Correr, eventualmente deverá (B) Beber Água. Esse *template* impede que (A) Correr seja executada isoladamente.
- *Template precedence(A,B)*: Se (B) Beber Água, eventualmente (A) Correr foi realizada antes. Não é possível somente (B) Beber Água sem que tenha sido realizado (A) Correr. Esse *template* impede que (B) Beber Água seja executada isoladamente.
- *Template succession(A,B)*: Sempre que (A) Correr for realizado, então eventualmente (B) Beber Água será realizado depois. Da mesma forma, se (B) Beber Água então (A) Correr foi realizado em algum momento anterior

a (B). Esse *template* impede que ambas as atividades sejam executadas isoladamente.

As Restrições de Relação do LTL Declare ainda apresentam mais seis *templates* que são variações dos três apresentados anteriormente. Eles se dividem em dois grupos: os *templates* de execução alternada e os de execução em cadeia.

Os *templates* de execução alternada definem que, dada uma relação entre atividades A e B, uma vez que A tenha sido executada, ela não poderá ocorrer novamente enquanto B não for executada. Os *templates* desses tipos de regra são *alternate response*, *alternate precedence* e *alternate succession*.

Já os *templates* de execução em cadeia definem que, dada uma relação entre atividades A e B, sempre que a atividade A for executada, B tem que ser executada logo em seguida. Os *templates* desses tipos de regra são *chain response*, *chain precedence* e *chain succession*.

A Figura 9 ilustra as fórmulas de cada um dos *templates* das Restrições de Relação do LTL Declare e suas respectivas representações gráficas.

name of formula	LTL expression
$responded\ existence(A, B)$	$\Diamond(A, t_c) \Rightarrow \Diamond(B, t_c)$
$co\ existence(A, B)$	$\Diamond(A, t_c) \Leftrightarrow \Diamond(B, t_c)$
$response(A, B)$	$\Box((A, t_c) \Rightarrow \Diamond(B, t_c))$
$precedence(A, B)$	$(!((B, t_s) \vee (B, t_c) \vee (B, t_x)))W(A, t_c)$
$succession(A, B)$	$response(A, B) \wedge precedence(A, B)$
$alternate\ response(A, B)$	$response(A, B) \wedge \Box((A, t_c) \Rightarrow \bigcirc(precedence(B, A)))$
$alternate\ precedence(A, B)$	$precedence(A, B) \wedge \Box((B, t_c) \Rightarrow \bigcirc(precedence(A, B)))$
$alternate\ succession(A, B)$	$alternate\ response(A, B) \wedge alternate\ precedence(A, B)$
$chain\ response(A, B)$	$response(A, B) \wedge \Box((A, t_c) \Rightarrow \bigcirc(B, t_s))$
$chain\ precedence(A, B)$	$precedence(A, B) \wedge \Box(\bigcirc(B, t_s) \Rightarrow (A, t_c))$
$chain\ succession(A, B)$	$chain\ response(A, B) \wedge chain\ precedence(A, B)$

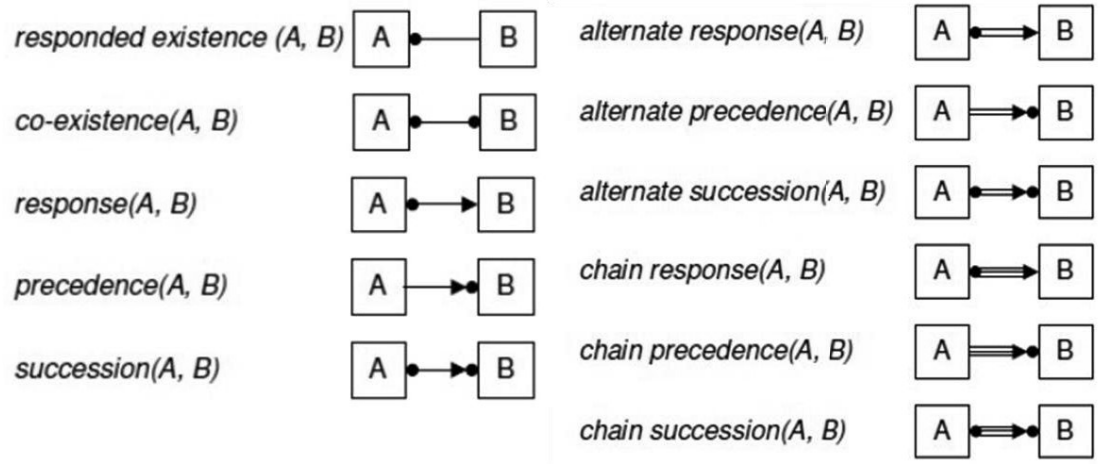

Figura 9 – Restrições de Relação do LTL Declare e sua representação gráfica (PESIC, 2008)

Para exemplificar as três restrições básicas de Relação do LTL Declare, a Tabela 2 descreve um grupo de instâncias de processos e as respectivas validações nos *templates precedence(A,B), response(A,B) e succession(A,B)*, onde os parâmetros A e B representam duas atividades quaisquer e os símbolos ○ e ● representam o início e término do processo, respectivamente.

Tabela 2 – Representação das instâncias de um processo e sua validade segundo as Relações de Restrição

Instância	Precedence	Response	Succession
○→**A**→●	Válida	Violação	Violação
○→**B**→●	Violação	Válida	Violação
○→**A**→**B**→●	Válida	Válida	Válida
○→**B**→**A**→●	Violação	Violação	Violação
○→**A**→**B**→**A**→●	Válida	Violação	Violação
○→**B**→**A**→**B**→●	Violação	Válida	Violação
○→**A**→**A**→**B**→●	Válida	Válida	Válida
○→**A**→**B**→**B**→●	Válida	Válida	Válida
○→**A**→**A**→**B**→**A**→●	Válida	Violação	Violação

As Restrições de Relação também são descritas por PESIC (2008) em função da sua negação, ou seja, relações onde as restrições temporais não devem acontecer. No LTL Declare, as **Restrições de Negativa de Relação** formalizam os relacionamentos entre eventos do processo de negócio definindo, principalmente, as relações temporais não existentes entre eles. A Figura 10 ilustra o conjunto de fórmulas que compõem essa restrição.

name of formula	LTL expression
$not\ responded\ existence(A,B)$ $not\ co\ existence(A,B)$	$\Diamond(A,t_c) \Rightarrow !(\Diamond(B,t_c))$ $not\ responded\ existence(A,B) \wedge$ $not\ responded\ existence(B,A)$
$not\ response(A,B)$ $not\ precedence(A,B)$ $not\ succession(A,B)$	$\Box((A,t_c) \Rightarrow !(\Diamond((B,t_s) \vee (B,t_c))))$ $\Box(\Diamond(B,t_s) \Rightarrow (!(A,t_c)))$ $not\ response(A,B) \wedge$ $not\ precedence(A,B)$
$not\ chain\ response(A,B)$ $not\ chain\ precedence(A,B)$ $not\ chain\ succession(A,B)$	$\Box((A,t_c) \Rightarrow \bigcirc(!(B,t_s)))$ $\Box(\bigcirc(B,t_s) \Rightarrow !(A,t_c))$ $not\ chain\ response(A,B) \wedge$ $not\ chain\ precedence(A,B)$

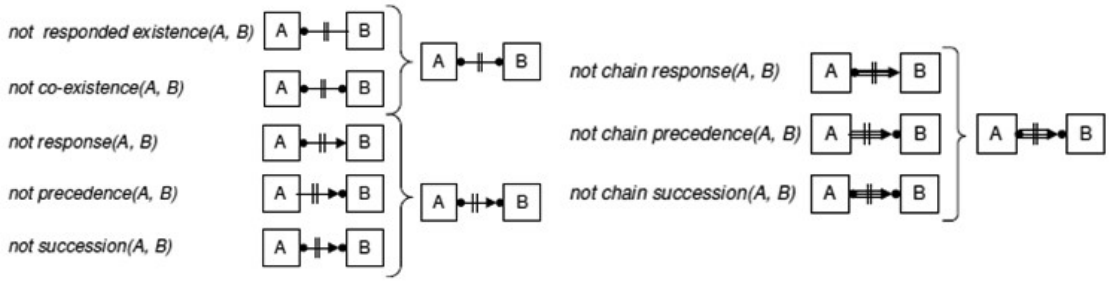

Figura 10 – Restrições de Negativa de Relação do LTL Declare e sua representação gráfica (PESIC, 2008)

As Restrições de Negativa de Relação possuem significado semelhante a seus equivalentes das Restrições de Relação, com a diferença de que esses *templates* restringem a impossibilidade de eventos ocorrem na ordem especificada. PESIC (2008) aponta que nas Restrições de Negativa de Relação não existem os templates referentes a execução alternada, ou seja, *not alternate response (A,B)*, *not alternate precedence(A,B)* e *not alternate sucession(A,B)* pois uma relação restrita pelo *template alternate response(A,B)* tem a mesma semântica de uma relação restrita pelo *template not alternate response(B, A)* sendo, portanto, removidos desse grupo de restrições.

As Restrições de Escolha representam sentenças onde são delimitadas a quantidade de escolhas possíveis dentre eventos disponíveis. Os *templates* de restrição de escolha estão agrupados entre os não exclusivos e os exclusivos. Os *templates* de escolha não exclusivos são *1 of 2*, *1 of 3*, *1 of N* e *2 of 3*. Já os *templates* de escolha exclusivos são *exclusive 1 of 2*, *exclusive 1 of 3* e *exclusive 2 of 3*.

A Figura 11 ilustra as fórmulas e representação gráfica de cada um dos *templates* das Restrições de Relação do LTL Declare.

name of formula	LTL expression
1 of 2(A,B)	$\Diamond(A,t_c) \vee \Diamond(B,t_c)$
1 of 3(A,B,C)	$\Diamond(A,t_c) \vee \Diamond(B,t_c) \vee \Diamond(C,t_c)$
1 of N$(A_1,\ldots,A_N)$	$\Diamond(A_1,t_c) \vee \ldots \vee \Diamond(A_N,t_c)$
2 of 3(A,B,C)	$(\Diamond(A,t_c) \wedge \Diamond(B,t_c))$ $\vee(\Diamond(B,t_c) \wedge \Diamond(C,t_c))$ $\vee(\Diamond(A,t_c) \wedge \Diamond(C,t_c))$
exclusive 1 of 2(A,B)	$(\Diamond(A,t_c) \wedge !\Diamond(B,t_c)) \vee (!\Diamond(A,t_c) \wedge \Diamond(B,t_c))$
exclusive 1 of 3(A,B,C)	$(\Diamond(A,t_c) \wedge !\Diamond(B,t_c) \wedge !\Diamond(C,t_c)) \vee$ $(!\Diamond(A,t_c) \wedge \Diamond(B,t_c) \wedge !\Diamond(C,t_c)) \vee$ $(\Diamond(A,t_c) \wedge !\Diamond(B,t_c) \wedge !\Diamond(C,t_c))$
exclusive 2 of 3(A,B,C)	$(\Diamond(A,t_c) \wedge \Diamond(B,t_c) \wedge !\Diamond(C,t_c))$ $\vee(\Diamond(A,t_c) \wedge !\Diamond(B,t_c) \wedge \Diamond(C,t_c))$ $\vee(!\Diamond(A,t_c) \wedge \Diamond(B,t_c) \wedge \Diamond(C,t_c))$

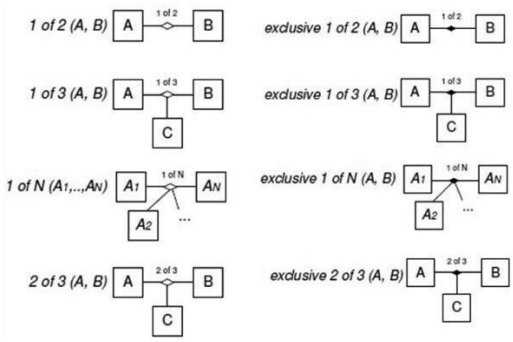

Figura 11 – Restrições de Escolha do LTL Declare e suas respectivas representações gráficas (PESIC, 2008)

2.4 Ontologia de Fundamentação Unificada (*Unified Foundational Ontology* – UFO*)*

A *Unified Foundational Ontology (UFO)* é uma ontologia de fundamentação desenvolvida para suportar atividades de modelagem conceitual e organizacional através de três camadas distintas conhecidas como *compliance sets*: UFO-A, uma ontologia de objetos; UFO-B, uma ontologia de eventos e UFO-C, que descreve entidades sociais,

linguísticas e de intenção. Sendo uma ontologia de alto nível, conforme definição proposta por GUARINO (1998), todos os conceitos de um domínio devem ser especializações de algum tipo de categoria definida na UFO. Como parte deste trabalho é baseado na Ontologia de Regras de Negócio proposta por LOPES (2010), os construtos da UFO-A e UFO-B, principalmente, serão amplamente utilizados na extensão da estrutura e semântica desta ontologia. A seguir, são apresentados os conceitos da UFO-A e UFO-B concernentes aos elementos que utilizados para fundamentar a extensão proposta neste trabalho.

2.5 Ontologia de Processos Intensivos em Conhecimento (*Knowledge-intensive Processes Ontology* – KiPO)

A Ontologia de Processos intensivos em Conhecimento (KiPO) foi proposta por FRANÇA et al. (2014) para permitir uma representação mais precisa de processos intensivos em conhecimento e as relações entre os elementos que o compõem. É uma ontologia de domínio que estrutura os conceitos e relacionamentos relevantes para fornecer um entendimento completo e preciso para entender um KiP e suportar a representação desses tipos de processos.

A KiPO é fundamentada na UFO (FRANÇA, NETTO, *et al.*, 2014) cujos elementos fornecem a semântica precisa para os conceitos da KiPO, impedindo que a ontologia possua problemas como ambiguidade de conceitos e permitindo que seja aplicada na modelagem, descoberta e suporte à execução de processos intensivos em conhecimento.

Diversas perspectivas compõem a KiPO, cada uma descrevendo um aspecto particular característico de processos intensivos em conhecimento: (i) BPO, a Ontologia de Processos que tem como base o metamodelo da BPMN; (ii) DO, a Ontologia de Decisão (PEREIRA e SANTORO, 2010); (iii) CO, a Ontologia de Colaboração (OLIVEIRA, 2009); (iv) BRO, a Ontologia de Regras de Negócio (LOPES, BAIÃO e SIQUEIRA, 2010) e (v) KiPCO, o componente principal (*core*) da KiPO. Através desse conjunto de elementos, a KiPO é capaz de representar diversos aspectos relacionados ao entendimento e execução de um processo intensivo em conhecimento.

2.5.1 Ontologia de Processos (BPO)

A *Business Process Ontology* (BPO) é uma perspectiva da KiPO baseada no metamodelo da BPMN, com o objetivo de representar os conceitos mais convencionais inerentes a um processo intensivo em conhecimento, como atividades e fluxos de controle, principalmente pelo fato de especialistas da área de processos de negócio já estarem habituados aos conceitos e notação descritas pela BPMN (FRANÇA, NETTO, *et al.*, 2014). Além disso, a incorporação do metamodelo da BPMN como uma perspectiva da KiPO permite que qualquer processo de negócio imperativo modelado com a notação possa ser facilmente transcrito como um metamodelo da KiPO. A Figura 12 ilustra uma porção da perspectiva da BPO, destacando alguns dos conceitos que de alguma forma são expandidos pela KiPO em outras perspectivas.

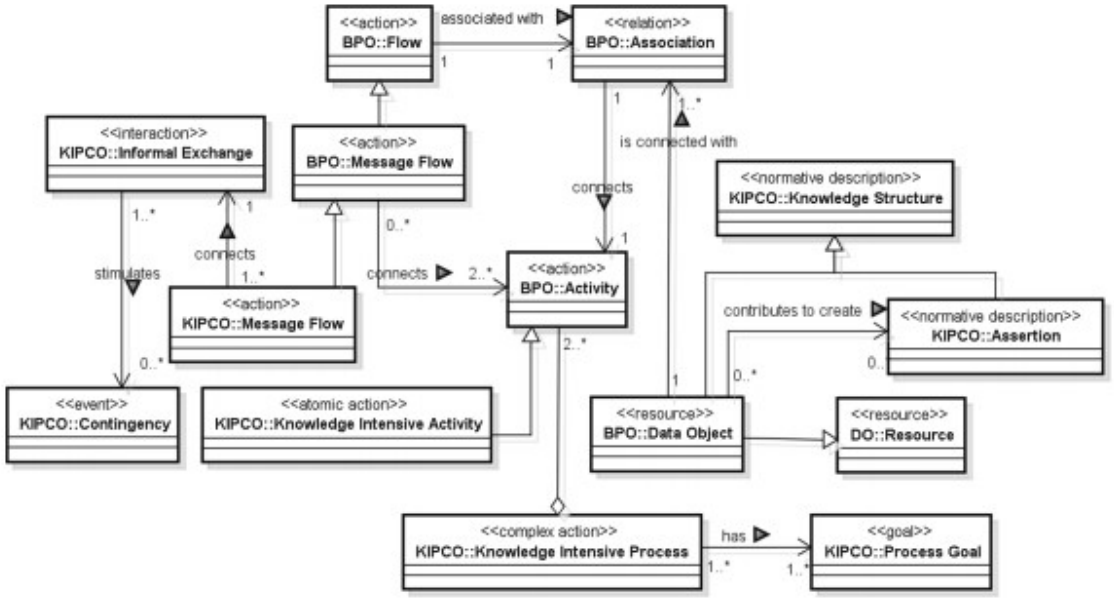

Figura 12 – Extrato do Metamodelo da BPO na KiPO (FRANÇA, NETTO, *et al.*, 2014)

A perspectiva da BPO é relevante na KiPO pois o conceito de Processo intensivo em Conhecimento é uma especialização de um processo de negócio tradicional, acrescentando aspectos que ajudam a entender um processo intensivo em conhecimento (KiP). Em alto nível de abstração, um KiP é tipicamente composto de um conjunto de atividades (*Activity*) que são associadas entre si através de um fluxo de mensagens (*Message Flow*). Em um KiP, esses elementos são associados a conceitos mais complexos de forma a representar cenários que a BPMN não é capaz. Por exemplo, atividades que envolvem decisões são especializações das atividades descritas na BPMN e chamadas de

Atividades intensivas em Conhecimento (*Knowledge-intensive Activity*) que, assim como Fluxos de Mensagens na BPO (*Message Flow*), são especializados em Fluxos de Mensagens (*Message Flow*) na KiPO de forma a poder representar a interação entre troca informal de conhecimento (*Informal Exchange*).

Atividades (*Activity*) podem ser associadas (*Association*) com Objetos de Dados (*Data Object*) que são um tipo especializado de Recurso (*Resource*) que representa dados, documentos ou qualquer outra fonte de informação utilizada por uma Atividade. Esses Objetos de Dados (*Data Objects*) contribuem para criar uma Afirmação (*Assertion*) acerca da Atividade.

Por fim, um Processo intensivo em Conhecimento (*Knowledge Intensive Process*) é o conjunto de atividades realizadas no contexto de um domínio para atingir um determinado Objetivo do Processo (*Process Goal*).

A BPO, portanto, é a perspectiva da KiPO que representa os conceitos inerentes a qualquer processo de negócio, sendo utilizada pelas demais perspectivas da KiPO para representar novos conceitos e relações entre aspectos inerentes a processos intensivos em conhecimento.

2.5.2 Ontologia de Decisão (DO)

Originalmente proposta por PEREIRA E SANTORO (2010), a perspectiva da *Decision Ontology* (DO) foi adicionada à KiPO através da fundamentação de seus conceitos na UFO. A perspectiva permite explicitar a racionalização da tomada de decisão por um agente do processo de negócio, representando os fatores que o levaram a tomar uma decisão em particular. Essa perspectiva é relevante pois, devido a natureza não-estruturada, um KiP frequentemente envolve a necessidade de tomada de diversas decisões.

De acordo com a ontologia, um Agente (*Agent*) é quem intencionalmente se compromete a resolver uma Questão (*Question*) através da tomada de uma Decisão (*Decision*), escolhendo entre Alternativas (*Alternative*) que representam situações potenciais que serão atingidas (*Chosen Alternative*) ou não (*Discarded Alternative*) quando a decisão for tomada. A Decisão (*Decision*) ainda pode considerar Restrições (*Restriction*) do domínio no estado do processo de negócio naquele momento, em

aspectos relacionados como restrições legais ou regras de negócio da organização. A Figura 13 ilustra a perspectiva da DO na KiPO.

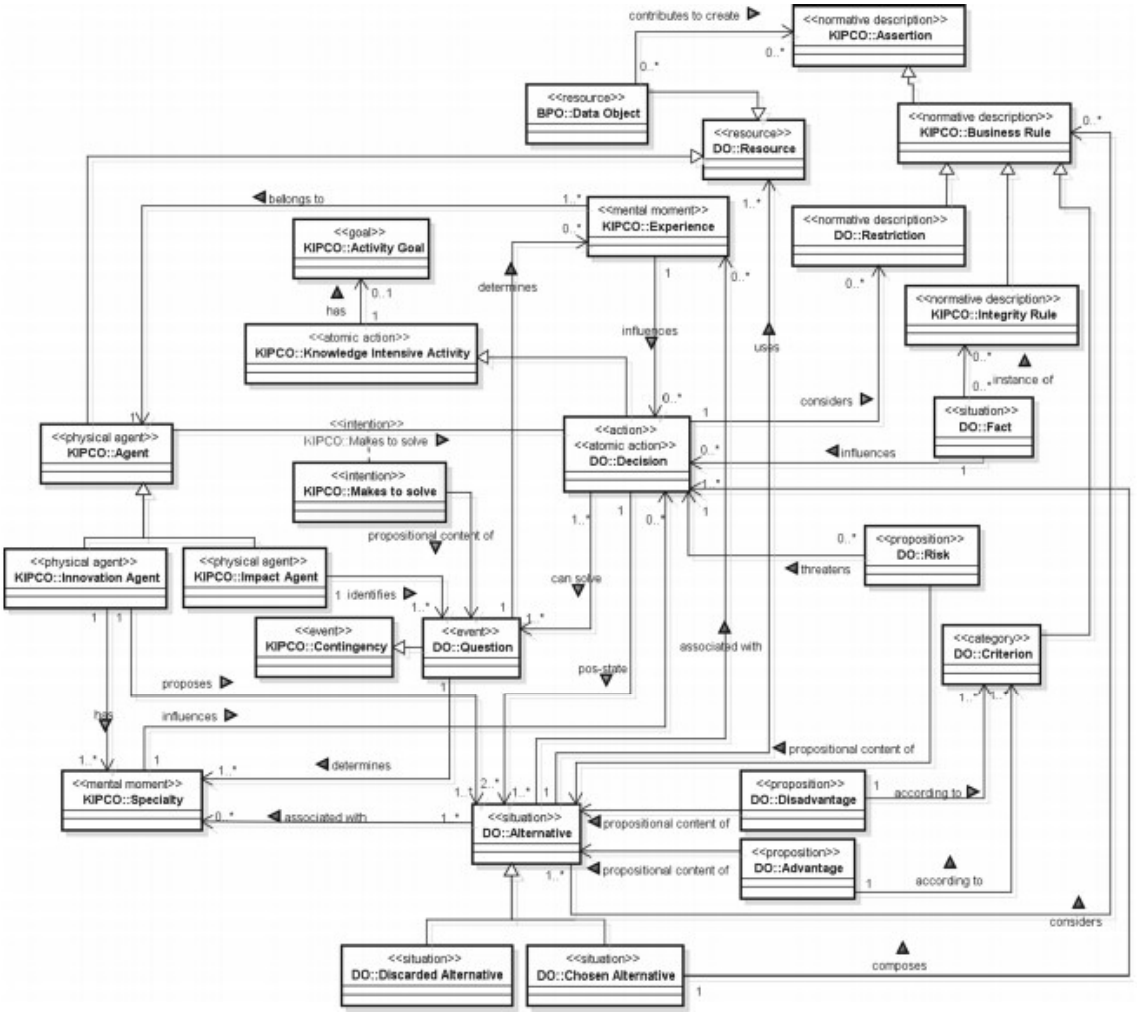

Figura 13 – Metamodelo da DO na KiPO (FRANÇA, NETTO, *et al.*, 2014)

2.5.3 Ontologia de Colaboração (CO)

A *Collaboration Ontology* (CO) foi desenvolvida por OLIVEIRA (2009) para representar a interação entre pessoas sob os aspectos de comunicação, cooperação e coordenação. Os conceitos descritos sob o aspecto de comunicação apoiam a realização de atividades onde há troca de informações entre pessoas e expressam emoções, desejos e ideias (FRANÇA, NETTO, *et al.*, 2014); a cooperação engloba os conceitos relacionados ao trabalho realizado em conjunto pelas pessoas em um espaço compartilhado; a coordenação, por fim, define os conceitos que promovem a organização e harmonia entre os processos de comunicação e cooperação (OLIVEIRA, 2009). A Figura 14 ilustra a perspectiva na KiPO.

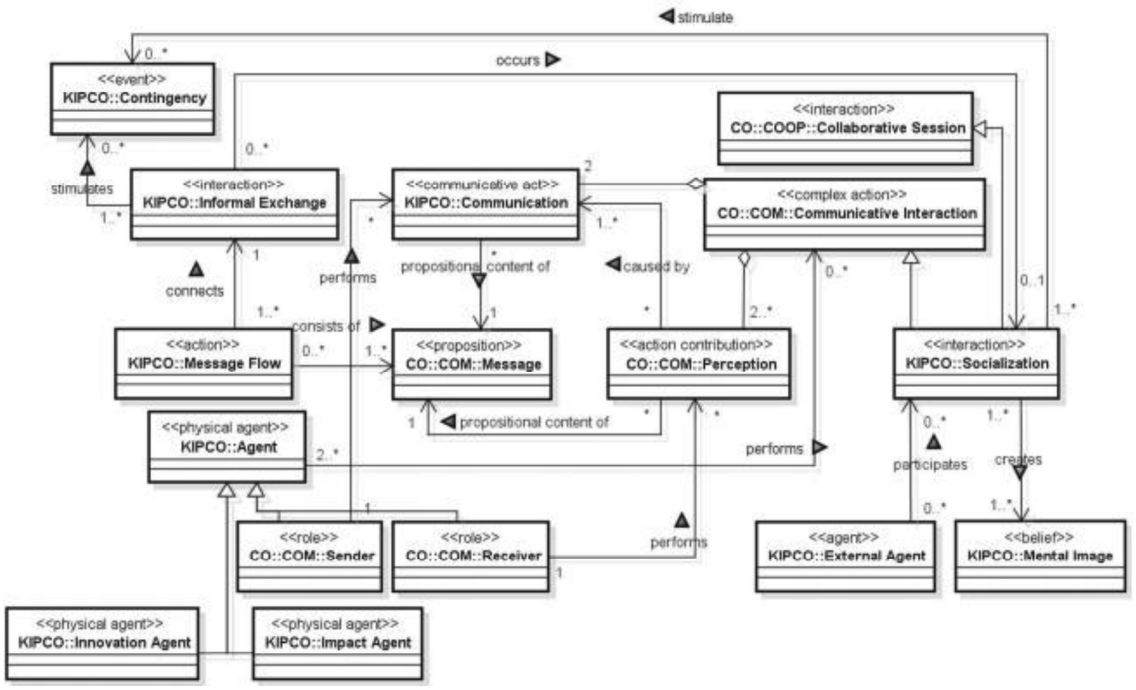

Figura 14 – Metamodelo da CO na KiPO (FRANÇA, NETTO, *et al.*, 2014)

Um Agente (Agent) realiza Interações Comunicativas (*Communicative Interaction*), que são compostas por Comunicações (*Communication*) e Percepções (*Perception*). Essas comunicações são realizadas através da troca de Mensagens (*Message*) entre Agentes (*Agent*), que assumem o papel de Receptores (*Receiver*) ou Emissores (*Sender*) em cada Interação Comunicativa (*Communicative Interaction*). Quando há discussão entre participantes ou uma ação requer um consenso, entre eles ocorre uma Socialização (*Socialization*), um tipo especial de Interação Comunicativa que pode envolver inclusive Agentes Externos (*External Agent*).

2.5.4 Ontologia de Regras de Negócio (BRO)

A *Business Rule Ontology* (BRO) foi inicialmente proposta por LOPES et al. (2010) para permitir a representação de restrições do domínio que serão tipicamente instanciadas durante a execução de um processo intensivo em conhecimento (KiP).

A ontologia descreve as regras de negócio que são derivadas de fontes diversas e que frequentemente são consideradas para a tomada de decisão, sendo estas organizadas conforme a definição de WAGNER (2004) no nível de Domínio de Negócio Independente de Computação (CIM) e sendo especializadas em **Regras de Integridade**, que expressam declarações que devem ser verdade em qualquer estado, **Regras de Derivação**, que definem as condições em que um elemento do domínio pode ser derivado

para outro conceito, e **Regras de Reação**, que especificam a execução de uma ou mais ações quando da ocorrência de um evento disparador ou a satisfação de um conjunto de condições. A Figura 15 ilustra o metamodelo da BRO na KiPO.

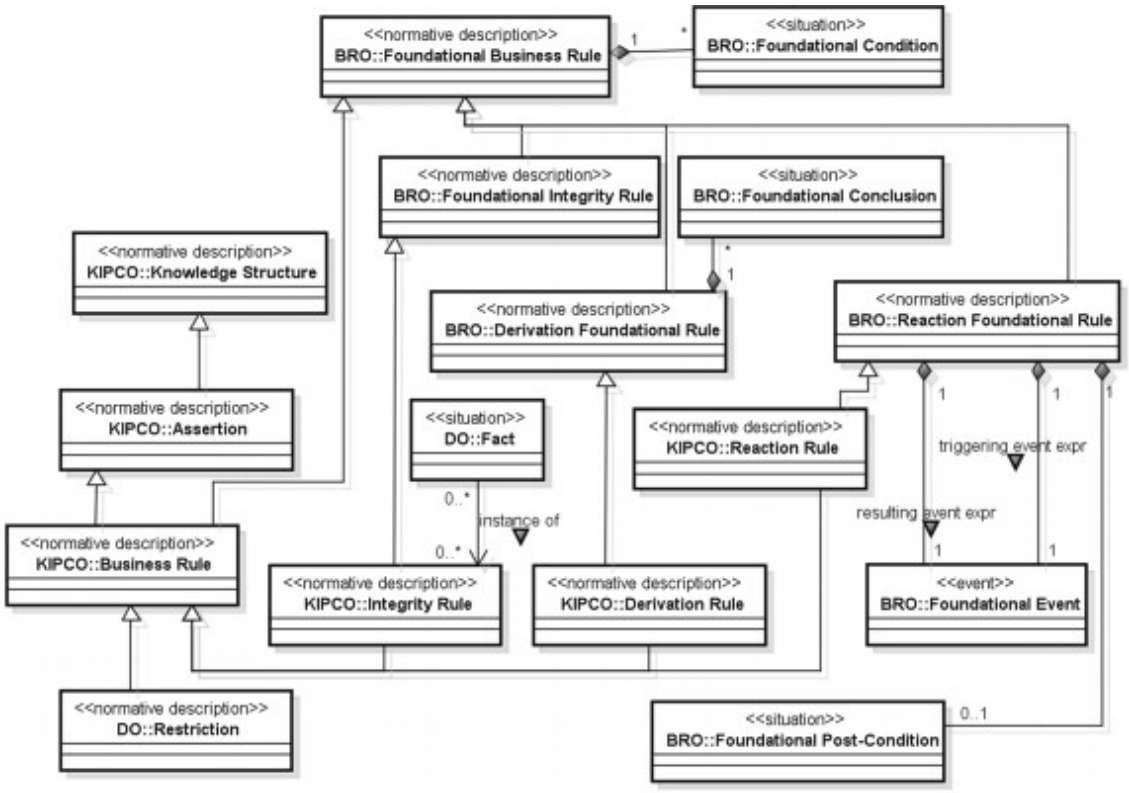

Figura 15 – Metamodelo da BRO na KiPO (FRANÇA, NETTO, *et al.*, 2014)

Como a Ontologia de Regras de Negócio (BRO) é o foco desta proposta, ela será descrita com maiores detalhes na seção 3.2.2.

2.5.5 Ontologia *Core* de Processos intensivos em Conhecimento (KiPCO)

A *KiP Core Ontology* (KiPCO) é a perspectiva cognitiva de um KiP, além de definir o relacionamento entre os conceitos das demais perspectivas da KiPO. Esta ontologia visa associar diversos conceitos para responder a questões sobre os desejos, intenções e crenças dos agentes do processo, sobre como ocorre a integração entre conhecimentos do negócio, quais eventos externos influenciam um processo intensivo em conhecimento ou como decisões acontecem em cenários desse tipo (FRANÇA, NETTO, *et al.*, 2014). A Figura 16 ilustra o metamodelo da KiPCO na KiPO.

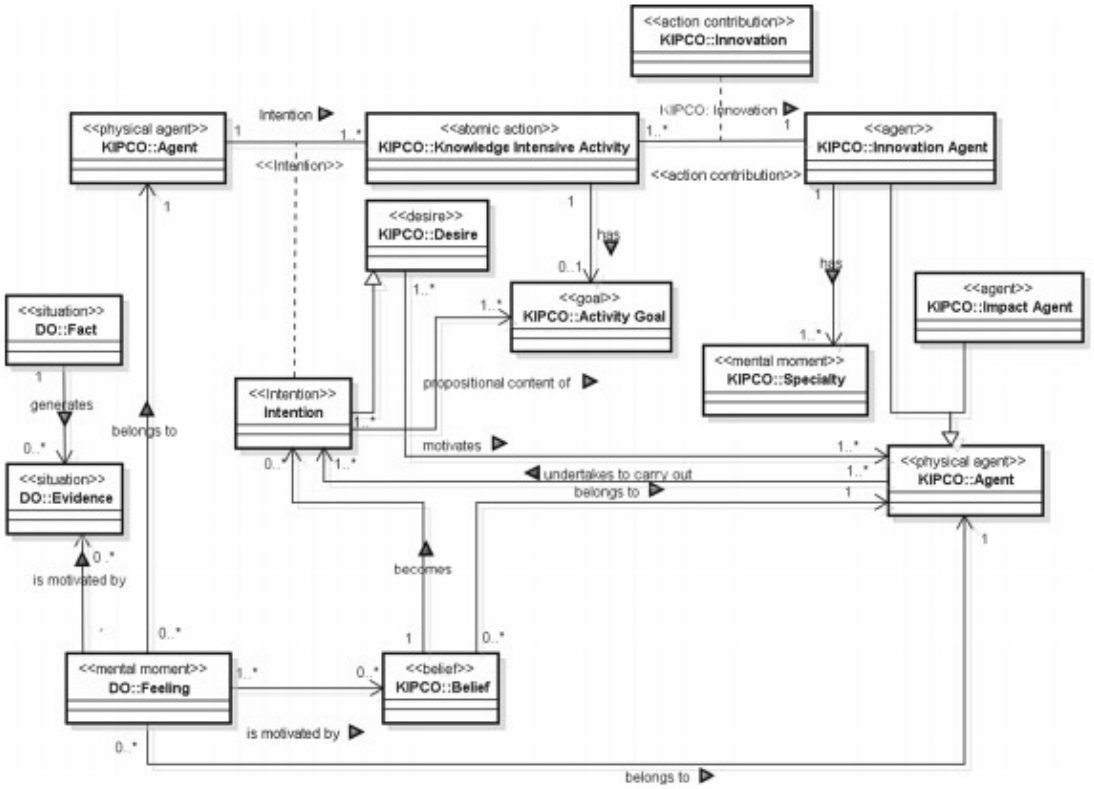

Figura 16 – Metamodelo da KiPCO na KiPO (FRANÇA, NETTO, *et al.*, 2014)

Um Agente (*Agent*) executa intencionalmente Atividades Intensivas em Conhecimento (*Knowledge-Intensive Activity*) para atingir um objetivo desejado (*Activity Goal*), motivado por seus Desejos (*Desire*) e guiado por suas Crenças (*Belief*). Em um KiP, um Agente (Agent) pode assumir o papel de Agente de Impacto (*Impact Agent*) que executa uma Atividade Intensiva em Conhecimento (*Knowledge-Intensive Activity*) e identifica Questões (*Question*) que levarão à necessidade de tomada de Decisão (*Decision*). Um Agente (*Agent*) também é especializado em Agente de Inovação (*Innovation Agent*), que contribui para a execução do processo com Inovação (*Innovation*).

3 Regras de Negócio

As ações realizadas em situações diversas do mundo real são delimitadas por certas restrições impostas pelo ambiente. Essas restrições definem o nível de liberdade que uma pessoa ou entidade tem para agir em determinada situação como, por exemplo, esperar a vez na fila para pagar uma conta no banco ou respeitar o limite de velocidade em uma via expressa. Sem regras de negócio, todas as tomadas de decisão seriam realizadas sem qualquer preparação, escolhendo-se entre alternativas caso-a-caso.

Segundo o BRG (2000), regra de negócio é uma declaração que define ou restringe algum aspecto de uma organização, sendo atômica, de forma que não pode ser dividida. Tem como objetivo afirmar a estrutura de um negócio ou controlar ou influenciar o comportamento deste. Dessa forma, é possível afirmar que regras de negócio podem ser usadas em conjunto aos modelos de processos, ajudando a defini-los. Segundo Halle (2002), as regras de negócio são as decisões que regem uma organização, compreendidas por políticas recomendadas e obrigatórias que governam a interação entre empregados, clientes, fornecedores e sistemas automatizados. Estas são as condições que governam os eventos do negócio de maneira que eles ocorram de forma aceitável para a organização.

Em modelagem de processos de negócio, regras de negócio podem estar atreladas à definição de atividades ou permeadas pelos processos, apoiando a execução destes, ao mesmo tempo em que os restringe, a partir das regras definidas no negócio da organização.

Essas restrições são interpretadas como regras que governam a forma de condução ou delimitam procedimentos em um ramo de atividade em particular (OMG, 2017). No ambiente de negócios, diversas restrições são criadas de forma a moldar o funcionamento dos processos definidos e garantir que os agentes, sejam humanos ou automatizados, executem suas atividades seguindo um comportamento esperado. Nesse sentido, regras que estão sob a jurisdição de um determinado negócio são chamadas de regras de negócio (OMG, 2017). Um aspecto importante sobre regras de negócio é que, estando sob a jurisdição de um negócio, significa que podem ser acrescentadas,

modificadas e descartadas conforme a necessidade da autoridade que as governa. Em contraste, legislações que se apliquem ao negócio, regulações específicas ou outras restrições externas não são consideradas regras de negócio sob a perspectiva da organização, apesar de afetarem-no.

Apesar disso, essas regras externas ainda devem ser consideradas para delimitar o escopo de ações realizadas pela organização. Regulações ou legislação específicas, por exemplo, são obrigatórias sob um ponto de vista legal e devem de alguma forma ser refletidas nos processos internos.

As regras de negócio, na maior parte dos casos, são descritas em linguagem natural nas organizações. Porém, existem diversas linguagens que definem um método formal para explicitar tais regras de maneira estruturada, possibilitando tanto um entendimento maior de humanos quanto a interpretação através de máquinas de inferência e outras tecnologias.

3.1 Classificação das regras de negócio

O *Object Management Group* (OMG) definiu diferentes níveis de abstração de regras de negócio segundo o *framework* conhecido como *Model Driven Architecture* (MDA) (OMG, 2005), conforme ilustra a Figura 17: *Computation-independent business domain level* (CIM), *platform-independent operational design level* (PIM) e *platform-specific implementation level* (PSM).

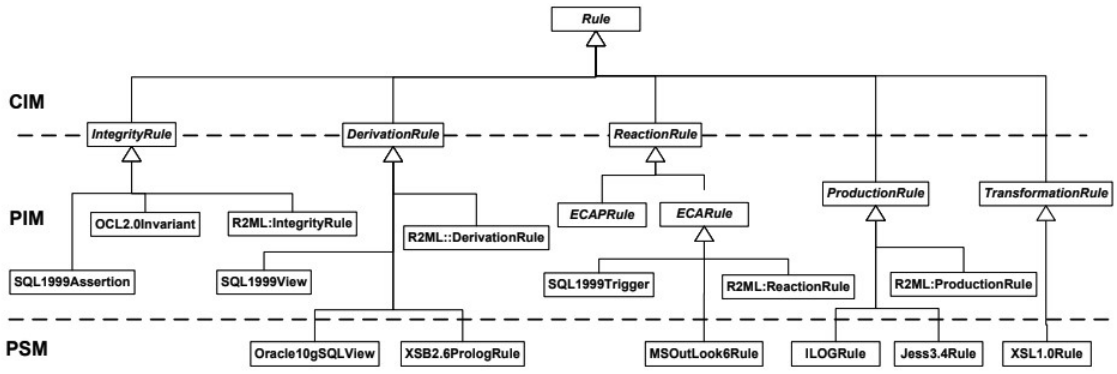

Figura 17 – Conceitos de modelagem de regras em três diferentes níveis de abstração (WAGNER, GIURCA e LUKICHEV, 2006)

No nível Independente de Computação (CIM), as regras são declarações que expressam partes das políticas do negócio ou domínio de maneira declarativa, normalmente

utilizando linguagem natural ou visual (WAGNER, GIURCA e LUKICHEV, 2006). Exemplos de regras de negócio neste nível são "*O motorista que aluga o veículo deve ter, no mínimo, 25 anos*" ou "*Os atendimentos a problemas informados por clientes devem ser solucionados em até 72 horas*".

No nível Independente de Plataforma (PIM), as regras são expressas de maneira formal através de algum paradigma computacional ou formalismo, de forma que possam ser diretamente mapeadas para a execução em sistemas de informação (WAGNER, GIURCA e LUKICHEV, 2006). A descrição das regras em nível PIM pode ser realizada através de *Structured Query Language* (SQL) e a *Object Constraint Language* (OCL), por exemplo, que definem formalismos claros e que se aplicam a quaisquer plataformas computacionais que se disponham a avaliar regras de negócio.

Por fim, no nível Específico de Plataforma (PSM), as regras são declaradas em uma linguagem específica do ambiente para o qual estão sendo implementadas (WAGNER, GIURCA e LUKICHEV, 2006). Neste caso, as regras são intercambiadas entre a descrição fornecida em nível Independente de Plataforma (PIM) e as linguagens específicas do ambiente. Exemplos de aplicação dessas regras são a descrição específica para o banco de dados Oracle ou implementadas na linguagem Java.

A OMG define que as regras em nível Independente de Computação (CIM) são assim descritas pois não consideram os detalhes da estrutura computacional (BRG, 2000). Isso implica que o especialista do domínio que determina as regras de negócio não conhece (ou não precisa conhecer) modelos e artefatos específicos para compreender quais requisitos governam seu negócio. Essa característica é evidenciada pelo fato das regras nesse nível de abstração serem normalmente descritas utilizando linguagem natural ou uma representação gráfica de alto nível.

Como a KiPO é uma Ontologia de Domínio, faz sentido que somente as regras modeladas nesse nível de abstração sejam consideradas. Além disso, como esse trabalho é baseado na proposta de LOPES et al. (2010), somente serão avaliadas as regras de negócio modeladas em nível Independente de Computação (CIM). Neste nível, WAGNER (2004) define a existência de três tipos de regras: Integridade (*Integrity Rules*), Derivação (*Derivation Rules*) e Reação (*Reaction Rules*).

Regra de Integridade consiste em uma sentença de restrição que pode ser expressa como uma sentença alética, como "*é necessariamente o caso que*" ou como uma sentença deôntica (*deontic*), como "*é obrigatório que*" (WAGNER, 2004). Este tipo de regra é estrutural, não altera o domínio, não cria novo evento ou ação que o altere. Esta regra somente restringe algo já existente entre os conceitos já previstos. Também relaciona-se apenas a estrutura do domínio e não se preocupa com o comportamento dos eventos em um processo de negócio. Regras de Negócio como "*Um carro deve ter um número de registro*" e "*Se o aluguel não for uma locação de ida, a filial de locação de retorno deve ser o mesmo que a locação de locação de locação*" são exemplos de regras de integridade.

Em seu trabalho, WAGNER (2004) descreve que as regras de integridade podem ser de dois tipos: as restrições do tipo estrutural (voltadas a delimitar a necessidade de atributos de elementos do domínio, como a necessidade de uma identificação para um carro) e as restrições de processos (que se referem à integridade dinâmica do sistema e restringem as transições admissíveis entre os estados do sistema). As regras de integridade estrutural são mais comuns e as que são frequentemente traduzidas para as linguagens independentes de plataforma. Com linguagem SQL, por exemplo, é possível descrever restrições de integridade estrutural como atributos e relacionamentos obrigatórios. O segundo tipo, apesar de relevante à proposta deste trabalho, não é descrito em detalhes em nenhum dos trabalhos de referência consultados para esta proposta. Na seção 4.2.1 será realizada uma discussão acerca desse tipo de regra de integridade e como ela foi considerada na proposta da DecKiPO.

As **Regras de Derivação** definem como o conhecimento pode ser transformado (LOPES, BAIÃO e SIQUEIRA, 2010). Regra que possui condição e conclusão e que explica como um elemento do modelo pode ser derivado. Ela representa a derivação de novos conceitos no domínio a partir de conhecimento já existente no domínio sendo modelado, e apresenta uma condição anterior à derivação. Quando o estado do domínio satisfizer essa condição, uma conclusão acontecerá, acrescentando um novo elemento ao domínio. A afirmação "*Um carro está disponível para locação, se não for atribuído a nenhuma locação e não está agendada para serviço*" é um exemplo de uma regra de derivação que deriva o conceito "Carro Disponível Para Locação".

As **Regras de Reação**, também conhecidas como regras de evento-condição-ação, consistem em uma afirmação de restrição que, no caso da ocorrência de um evento disparador, uma ou mais ações são executadas como resultado. A afirmação "*Se um carro alugado não for devolvido até a data final do aluguel, em seguida, cobrar do cliente pelo retorno atrasado*" define uma condição que, se for satisfeita, aciona a execução de outro evento.

3.2 Representação de regras de negócio

Diversas linguagens de representação de regras de negócio foram descritas nos últimos anos, cada uma delas possuindo finalidades e características distintas que atendem, em maior ou menor escala, as necessidades de representação de regras.

As linguagens de representação a seguir elencadas são focadas na modelagem formal de regras de negócio com o objetivo de interoperabilidade entre sistemas de informação. Estas linguagens têm a finalidade de prover a troca de regras entre sistemas de informação, descrevendo suas relações usando sintaxe e semântica específicas escritas em um formato aberto como XML, capazes de serem interpretadas por qualquer mecanismo de regras de negócio que conheça sua estrutura.

A URML é uma linguagem de modelagem de regras que utiliza elementos gráficos do diagrama de classes da UML (Serjik, 2006).

A RuleML é uma representação de regras para publicação e compartilhamento na Web (RuleML, 2011). É uma iniciativa global aberta com o objetivo de desenvolver a especificação de padrões de design relacionados à regras de negócio com ênfase na semântica e interoperabilidade entre áreas. A iniciativa RuleML inclui sintaxes de apresentação e serialização, linguagens configuráveis, transformações, modelos teóricos e motores de inferência.

A SWRL (*Semantic Web Rule Language*) é uma linguagem proposta submetida à OMG em 2004 cujo objetivo é expressar tanto regras quanto a lógica envolvida utilizando os conceitos combinados das linguagens OWL (*Ontology Web Language*) e RuleML.

A OWL é uma linguagem desenvolvida para definir e instanciar ontologias na web, permitindo especificar classes, propriedades e suas instâncias utilizando

principalmente a sintaxe da especificação RDF/XML. Em conjunto com as definições da RuleML, a SWRL estende a OWL acrescentando elementos que permitam a representação das regras de negócio.

3.2.1 SBVR

Existem linguagens cujo propósito é descrever regras em linguagem natural, facilitando a compreensão da representação de regras de negócio por especialistas em negócios e outras partes interessadas. Como a proposta deste trabalho é voltada para representação de regras de negócio em nível Independente de Computação (CIM), esses tipos de linguagem são mais adequados a serem analisados de forma a aumentar a compreensão e fundamentar claramente os elementos propostos na extensão DecKiPO.

A especificação *Semantics of Business Vocabulary and Business Rule* (SBVR) foca a definição de vocabulários e regras para documentação semântica de regras de negócio, com uma grande ênfase na descrição de regras usando linguagem natural e formalismo sintático e semântico, capaz de ser interpretado pela lógica modal (OMG, 2017). Ela suporta a especificação de regras de negócio com as seguintes premissas:

- Descrição a partir da perspectiva da organização e não de um sistema de informação;
- Utilização de vocabulário do negócio e não de modelos computacionais;
- Não tem como premissa que as regras de negócio são ou serão automatizadas.

Para a SBVR, a especificação das sentenças é descrita na forma de proposições, que são sentenças declarativas, que não são consideradas paradoxos, e não variam devido a traduções (OMG, 2017). Essas proposições podem ou não remover algum grau de liberdade das ações, não sendo necessariamente restrições que delimitam um comportamento. Quando removem algum grau de liberdade de ações, essas proposições são entendidas como Regras (*Rules*) enquanto aquelas onde a liberdade de ação nunca é removida são entendidas como Orientações (*Advices*).

Por exemplo, a proposição "*É proibido que um aluguel de veículo seja pago com cheque*" indica que os agentes não são completamente livres para agir quando se trata de pagamentos de aluguel de veículos e, por isso, essa proposição pode ser classificada como uma Regra (*Rule*). As regras, dentro do cenário de organizações, são normalmente

implementadas sob a forma de Regras de Negócio (*Business Rule*), derivadas a partir de um conjunto de políticas de negócio (*Business Policy*) que guiam a organização.

Enquanto Regras removem certa liberdade, Orientações (*Advices*) são o exato oposto: esse tipo de proposição sempre confirma ou lembra que algum grau de liberdade existe ou é permitido em um domínio, como na proposição "*É permitido que um aluguel de veículo seja pago como dinheiro*". Assim como as Regras de Negócio, as Orientações (*Advices*) são derivadas a partir de políticas de negócio (*Business Policy*) e guiam o comportamento dos agentes dentro da organização. A Figura 18 ilustra os tipos de proposições definidas pela SBVR.

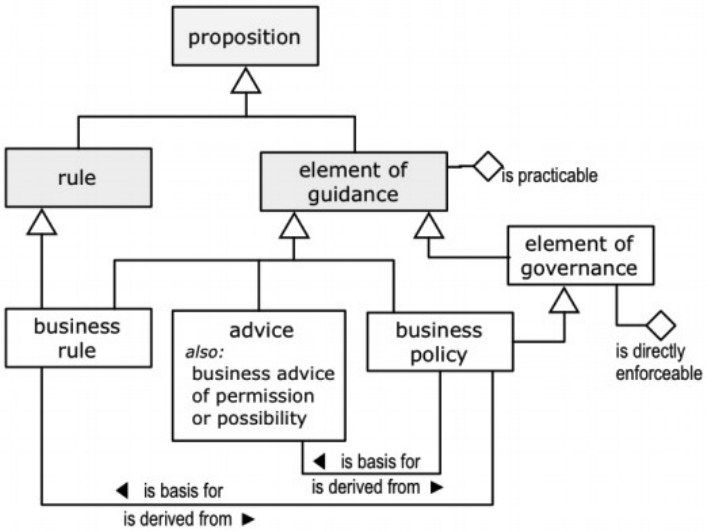

Figura 18 – Tipos de proposições segundo a SBVR. Fonte: (OMG, 2017)

Regras de Negócio (*Business Rules*) e Orientações (*Advices*) são derivados de Políticas de Negócio (*Business Policy*) pois estes não são diretamente executáveis por agentes da organização, normalmente possuindo as seguintes características (OMG, 2017):

- Menos estruturados;
- Mais abrangentes.
- Menos cuidadosos em termos de padronização de vocabulário.

Por exemplo, as políticas "*A frota de veículos disponíveis para aluguel deve sempre ser nova*" ou "*Clientes que reclamarem devem ser contatados pessoalmente por um representante da organização*" não são diretamente aplicáveis aos processos da organização, cabendo a regras de negócio definirem mais precisamente o que são "frota

nova" e "reclamação" ou uma atividade ser explicitamente criada para executar garantir a execução da política.

As proposições de Orientação (*Advices*) não são o foco específico deste trabalho, visto que a ontologia que serve como base para a extensão proposta possui elementos que têm características muito semelhantes aos descritos pela SBVR. Utilizando o exemplo acima, a declaração da política "A frota de veículos disponíveis para aluguel deve sempre ser nova" indica uma prática que deve ser seguida pela Organização e garantida através da execução de atividades do processo. Dessa forma, os *Advices*, de modo geral, podem ser modelados na KiPO como Objetivos do Processo (*Process Goals*) ou Objetivos da Atividade (*Activity Goals*).

Sobre as Regras de Negócio (*Business Rules*), a SBVR define as divide em dois grupos: as Regras de Negócio de Definição (*Definitional Business Rules*), que descrevem conceitos e as regras de negócios necessárias em um determinado estado de coisas e representam regras aléticas, ou seja, que não podem ser violadas; e as Regras de Negócio Comportamentais (*Behavioral Business Rules*), que descrevem regras de negócios que obrigam um determinado estado de coisas, representando regras deônticas, ou seja, que denotam um conceito de obrigatoriedade mas podem pode ser violadas dadas condições ou critérios específicos. As definições das *Definitional Business Rules* e *Behavioral Business Rules* são disjuntas, ou seja, nenhuma regra de definição pode ser também uma regra de comportamento, e vice-versa.

Um aspecto importante de ambos os tipos de regras é baseado na percepção do agente no que tange ao entendimento dos conceitos definidos (no caso das *Definitional Business Rules*) ou ao comportamento das pessoas (no caso das *Behavioral Business Rules*). No primeiro caso, a restrição é imposta por definição, ou seja, a própria estrutura da regra não permite que ela seja violada. Já no seguindo caso, mesmo conhecendo as restrições impostas, os agentes podem potencialmente violá-las ou ignorá-las (OMG, 2017).

As *Definitional Business Rules* podem ser expressas em termos de necessidade ou impossibilidade, introduzindo ou removendo liberdade no negócio, conforme ilustra a Figura 19.

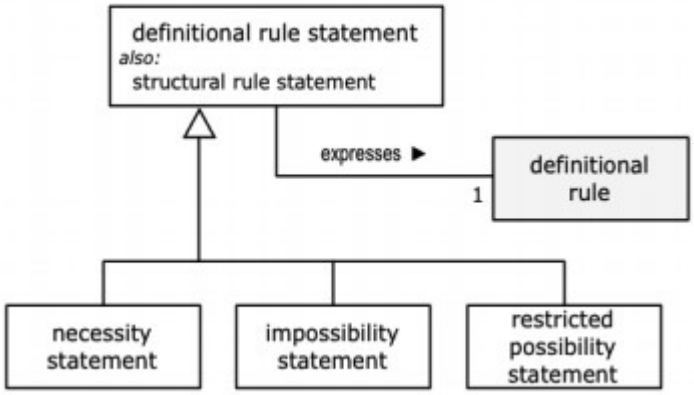

Figura 19 – Tipos de declaração das *Definitional Business Rules.* Fonte: (OMG, 2017)

As regras de definição descritas em termos da necessidade denotam casos onde estado das coisas (*state of affairs*) é sempre verdade. Por exemplo as sentenças "*É necessário que cada aluguel de veículo tenha exatamente um carro associado*" e "*Todo atendimento a problemas de clientes sempre deve ser identificado por um único Ticket*" são regras de negócio de definição, descritas sob a perspectiva de necessidade.

As mesmas regras de negócio podem ser descritas em função da impossibilidade onde, nesse caso, descrevem-se estados das coisas que nunca devem ser verdadeiros. Assim, as regras de negócio supracitadas poderiam ser descritas na forma "*É impossível que um aluguel de carro não tenha um carro associado*" e "*Nunca um atendimento a problemas de clientes não será identificado por um único Ticket*".

As Regras de Negócio classificadas como *Behavioral Business Rules* são aquelas que obrigam um determinado estado das coisas (*state of affairs*). Conforme definido pela SBVR (OMG, 2017), são regras que "prescrevem, sugerem ou impõem a condução ou ação" de agentes do processo. Ao contrário das *Definitional Business Rules*, elas descrevem um comportamento inerente ao domínio, delegando ao agente a capacidade de entendimento e ação sobre a operação da regra como, por exemplo, a regra de negócio "É obrigatório que uma locação incorra em uma taxa se a filial de devolução da locação for diferente daquela em que o automóvel foi alugado". A Figura 20 ilustra a estrutura das *Behavioral Business Rules* na SBVR.

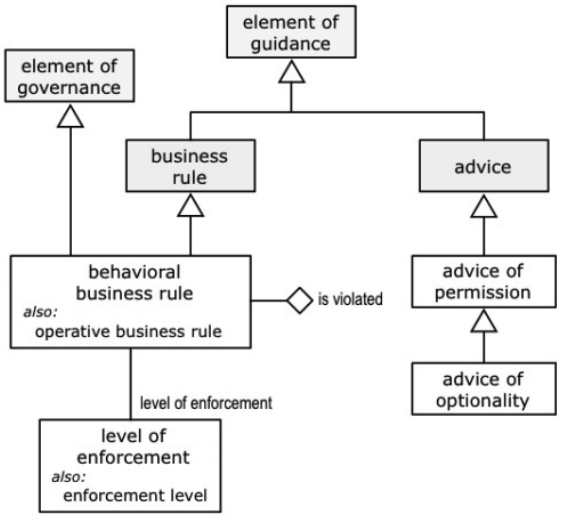

Figura 20 – Estrutura das *Behavioral Business Rules* (OMG, 2017)

Por causa de seu comportamento modal, as *Behavioral Business Rules* têm um atributo exclusivo que define o seu nível de cumprimento. Este nível é uma posição em uma escala definida que determina a severidade da restrição imposta pela regra de negócio, prescrevendo penalidades se for violada (OMG, 2017). Caso sejam descumpridas pelo agente, as regras de negócio são postas em estado de violação para que sejam tratadas.

Esses conceitos serão utilizados amplamente para fundamentar a extensão proposta neste trabalho e as adições de elementos descritas ao longo dos próximos capítulos.

3.2.2 Ontologia de Regras de Negócio (BRO)

LOPES et al. (2010) propuseram a *Business Rules Ontology (BRO)*, uma ontologia de regras de negócio bem fundamentada na UFO. A BRO compreende três subtipos de regras de negócio: regras de integridade, regras de derivação e regras de reação. Esses subtipos são originalmente baseados na taxonomia de regras de negócio proposta pela *Business Rule Group*, cuja definição é especializada dos grupos de *Structural Assertions, Action Assertions* e *Derivations* (BRG, 2000), respectivamente. De acordo com WAGNER et al. (2004), essas regras estão no nível de domínio do negócio, tendo importante papel na redução do espaço de entendimento entre os especialistas do domínio e os especialistas em modelagens de artefatos.

No KiPO, a perspectiva da Ontologia da Regra de Negócios, demonstrada na Figura 21, descreve os conceitos de regras de negócio baseados nas definições propostas

por WAGNER (2004), classificando as regras de negócios em Regras de Integridade, Regras de Derivação e Regras de Reação.

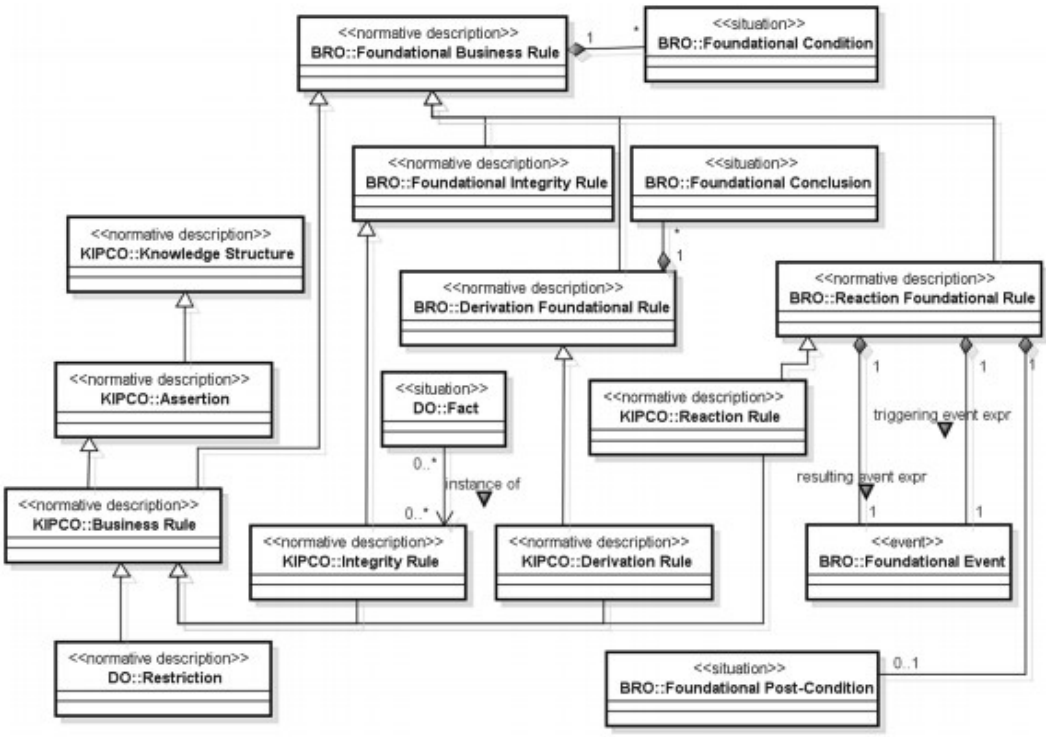

Figura 21 – A perspectiva da Ontologia de Regras de Negócio na KiPO

Sendo regras de negócio no nível Independente de Computação (CIM) declarações em linguagem natural das restrições de um domínio, os conceitos relativos a esses elementos são subtipos de *normative descriptions*, fundamentados pela UFO, cuja representação foi realizada na forma de estereótipos para efeitos de simplificação da visualização.

As regras de negócio na KiPO são especializações de Assertivas (*Assertion*) que apresentam o formalismo do conhecimento construído no processo, que por sua vez contribuem para a formação de uma Estrutura de Conhecimento (*Knowledge Structure*) sobre todo o domínio. As Regras de Negócio de Fundamentação (*Foundational Business Rules*) são afirmações que definem ou restringem aspectos de um domínio e se especializam nos três tipos de regras definidas por WAGNER (2004): Regras de Fundamentação de Integridade (*Foundational Integrity Rule*), Regras de Fundamentação de Derivação (*Derivation Foundational Rule*) e Regras de Fundamentação de Reação (*Foundational Reaction Rule*).

A execução de uma regra de negócio pode estar atrelada a zero ou mais Condições (*Foundational Condition*) que sempre são avaliadas para determinar se uma regra de negócio deve ou não ser considerada.

As Regras de Integridade (*Integrity Rule*) são representações na KiPCO das Regras de Fundamentação de Integridade (*Foundational Integrity Rules*) e representam elementos do domínio que restringem algo do mesmo, como cardinalidade ou relacionamento (LOPES, BAIÃO e SIQUEIRA, 2010). As Regras de Integridade descritas na KiPO são do tipo *alethic*, ou seja, não podem ser violadas, assemelhando-se aos conceitos de *Definitional Business Rules* descritos pela SBVR.

As Regras de Derivação (*Derivation Rule*) representam a derivação de novos conceitos no domínio a partir de um conhecimento já existente. Se as Condições (*Foundational Condition*) da Regra de Derivação forem satisfeitas, uma Conclusão (*Foundational Conclusion*) acontece e um novo conceito derivado do original é adicionado ao domínio. As Regras de Derivação também se assemelham ao conceito de *Definitional Business Rules* da SBVR, visto que não altera o comportamento do domínio, não podem ser violadas e sua integridade é garantida pela própria definição da regra de negócio.

As Regras de Reação (*Reaction Rule*) representam uma restrição no comportamento do domínio. As Regras de Reação são executadas sempre que as Condições (*Foundational Condition*) da regra são satisfeitas. Essas Condições somente são avaliadas a partir de um Evento (*Foundational Event*) disparador que gera um Evento resultante e uma Pós-Condição (*Foundational Post-Condition*) que representa uma Situação que altera o estado das coisas. As Regras de Reação se comportam como as *Behavioral Business Rules* da SBVR.

Na ontologia de regras de negócio (BRO) todos os *templates* LTL Declare das restrições de relação são elementos estendidos da classe *Reaction Rule* uma vez que esse elemento possui a semântica de um elemento ativador (*triggers*) e de um elemento resultante (*resulting*). A semântica da *Reaction Rule* impõe necessariamente que uma atividade resultante B seja consequência do estímulo de uma atividade ativadora A. O metamodelo da *Reaction Rule* é apresentado na Figura 22.

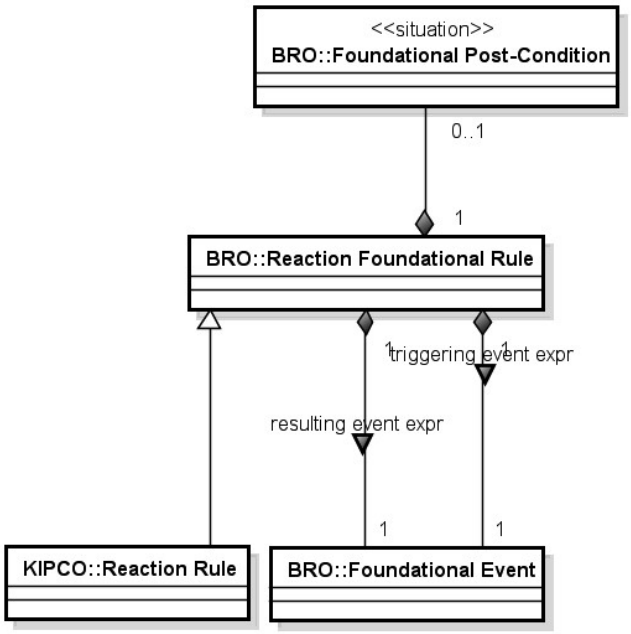

Figura 22 - Metamodelo da Regras de Reação (*Reaction Rule*) na KiPO

Apesar da literatura não descrever com precisão o tempo decorrido entre a ativação e o resultado, é plausível afirmar que a atividade resultante B será realizada eventualmente após a ativação da atividade A.

A Ontologia da Regra de Negócios (BRO) é essencial para representar as relações entre as atividades de um KiP, pois essas relações podem ser definidas em termos de suas restrições, em vez do controle de fluxo dos processos imperativos tradicionais.

Entretanto, esses três tipos de regras de negócio não são suficientes para representar relacionamentos e restrições mais complexas de um domínio visto que suas conceituações, explicadas na seção 3.1, não permitem determinar relações de causalidade entre eventos considerando a passagem de tempo na execução do processo.

3.3 Limitações da representação de regras de negócio

Conforme foi introduzido, as linguagens possuem características distintas e podem ser aplicadas em contextos diversos, algumas delas não tendo sido criadas necessariamente para representação de regras de negócio. Nesse sentido, SULL *et al.* (2011) realizaram uma comparação entre as linguagens considerando alguns critérios para relacionamento da linguagem com seu contexto de uso, apresentadas na Tabela 3.

Tabela 3 - Comparando linguagens de representação de regras de negócio. Adaptado de (Sull et. al., 2011)

	URML	**RuleML**	**SWRL**	**SBVR**
Qual seu objetivo?	Modelagem de regras utilizando elementos gráficos do diagrama de classes da UML.	Representação de regras, incluindo regras de negócio.	Representação de regras, incluindo regras de negócio.	Representação de regras de negócio.
Foi construído para modelagem de regras de negócio?	Sim	Não, mas contempla as regras de negócio.	Não, mas contempla as regras de negócio	Sim
Fácil entendimento pelo negócio?	Sim	Não	Não	Sim
Trata ambiguidade?	Sim	Sim	Sim	Sim
Nível de abstração	PIM	PIM	PIM	CIM
Observações	Permite uma modelagem gráfica de regras baseadas no modelo de classes UML.	Linguagem baseada em XML para representação e intercâmbio de regras com foco na web.	Propõe a combinação de RuleML com OWL, para acrescentar semântica a regras e ontologias.	Padrão OMG. Definida pelo "Business Rules Group"

As linguagens de representação de regras de negócio, URML, RuleML e SWRL são bastante robustas na representação de regras de negócio, mas são focados para uso computacional, exigindo competências técnicas que especialistas do domínio podem não possuir. Porém, como se tratam de linguagens para modelagem em nível Independente de Plataforma (PIM), foram elencadas neste trabalho apenas para efeito de estudo das referências bibliográficas, não é objeto de estudo desta proposta.

Quanto à representação no nível de abstração Independente de Computação (CIM), foram apresentadas as taxonomias da SBVR, que define uma estrutura de descrição de regras de negócio sem a preocupação de automatização e a BRO, que se propõe a modelar regras de negócio sob a perspectivas de processos intensivos em conhecimento (KiP).

Analisando a SBVR, seus conceitos permitem uma descrição estruturada e precisa de regras de negócio, através da definição de uma sintaxe formal e semântica baseada em lógica modal. Entretanto, como tratamos de processos de negócio em cenários

intensivos em conhecimento, a SBVR não possui os conceitos necessários para delinear todas as interações que ocorrem durante processos complexos como de uma tomada de decisão.

Sobre a Ontologia de Regras de Negócio (BRO), seus conceitos são baseados amplamente nas definições de WAGNER (2006) e da BRG (2000) e são bem fundamentadas na UFO, garantindo uma semântica mais precisa dos elementos. A BRO também se beneficia da relação com a KiPO, pois, através da KiPCO, define relações com conceitos de tomada de decisão, interação entre agentes e avaliação de alternativas de processos.

Entretanto, em cenários de processos intensivos em conhecimento, algumas restrições inerentes a estes tipos de processo não são cobertas pelas regras de negócio descritas na ontologia. Dada a característica não-estruturada e imprevisível de KiPs, as regras de negócio são imprescindíveis para a modelagem das restrições do domínio tanto sob o aspecto estrutural quanto sob o aspecto comportamental.

Analisando cada um dos tipos de regras de negócio da BRO individualmente, podemos chegar a algumas conclusões que corroboram um dos problemas abordados por este trabalho: as ontologias de regras de negócio não conseguem expressar por completo todas as restrições impostas em cenários de processos intensivos em conhecimento.

Começando pelas Regras de Integridade (*Integrity Rules*), podemos avaliar que somente são cobertas as regras que descrevem a estrutura de elementos do domínio e não a integridade do comportamento das atividades executadas. Por exemplo, uma regra de negócio *r* **"***Todo atendimento a problemas de clientes sempre deve ser identificado por um único Ticket***"** é uma regra de integridade que cria uma restrição estrutural no domínio. Em contrapartida, uma regra de negócio *r'* **"***Ao menos 3 tentativas de ligação devem ser feitas para clientes antes do cancelamento da proposta de empréstimo***"** também representa uma restrição de integridade, mas dessa vez relacionada ao processo como um todo e não somente a um elemento do domínio.

Conforme a definição da SBVR, uma regra de negócio estrutural é restrita por definição, não sendo possível violá-la em qualquer estado das coisas. No caso de *r'*, o agente pode ativamente optar por ignorá-la ou outra situação no processo pode

impossibilitar sua execução. Assim, *r'* impõe um aspecto de integridade no domínio que não pode ser representada pela BRO.

Outra limitação na representação está relacionada às Regras de Reação. Também formalmente conhecidas como Evento-Condição-Ação (*Event-Condition-Action*), esse tipo de regra de negócio dispara um evento resultante sempre que a ativação por outro evento satisfaz as condições previstas. Esse comportamento denota restrições onde há uma certa progressão de ações e atende a maior parte dos cenários onde eventos devem ser disparados automaticamente após algumas condições, como por exemplo envio de e-mails de confirmação de transações financeiras, criação de log de eventos após uma atualização de dados, entre outros.

Finalmente, os diversos trabalhos na literatura (BRG, 2000), (OMG, 2005), (WAGNER, GIURCA e LUKICHEV, 2006) que fundamentam as regras de reação não estabelecem o tempo decorrido entre a avaliação das condições e o disparo do evento resultante, levando a conclusão que ou o processo é instantâneo ou o tempo não é fator relevante. Porém, existem restrições onde há a necessidade de se estabelecer estritamente um aspecto temporal na relação entre os eventos que são restringidos pela regra de negócio. Por exemplo, considere as regras de negócio "*É obrigatório que uma confirmação por e-mail com o resultado da análise seja enviada ao cliente imediatamente após a conclusão da verificação pelo gerente.*" e "*É obrigatório que um pedido de prorrogação do período de locação em aberto seja analisado antes que uma nova solicitação possa ser feita;*". Enquanto a primeira regra de negócio poderia ser representa como uma Regra de Reação, o aspecto temporal explicitado pela palavra "imediatamente" na regra de negócio seria perdido em sua avaliação. Já na segunda regra de negócio, há uma relação de exclusividade entre a execução de ambas as atividades que a Regra de Reação não consegue avaliar, principalmente porque a atividade "Realizar pedido de prorrogação do período de locação" não é necessariamente o evento disparador da atividade "Analisar solicitação". Esse tipo de restrição também não pode ser modelado como uma Regra de Integridade, pois ela está delineando o comportamento dos eventos do domínio e não da estrutura de elementos.

Os exemplos fornecidos neste trabalho são focados em eventos, especialmente Atividades, pois em cenários de processos intensivos em conhecimento todas as interações, socializações e tomadas de decisão realizadas por agentes do processo são

guiadas para que se atinja um objetivo, que só é alcançado com a realização de Atividades Intensivas em Conhecimento (*KiA*) que levem à essa conclusão.

Outro ponto analisado é que, com exceção de extensões da RuleML e SWRL, nenhuma outra linguagem tem expressividade suficiente para representar relações temporais entre regras de negócio e eventos, sendo restritas a relações de obrigatoriedade e possibilidade. Entretanto, os aspectos temporais da RuleML e SWRL são ancorados principalmente sob a perspectiva de um tempo absoluto *t*, focando na relação temporal entre instantes de tempo e não entre eventos propriamente ditos, além do que essas linguagens são utilizadas para representação das regras de negócio sob o nível Independente de Plataforma (PIM), visando um formalismo para interoperabilidade entre sistemas de informação.

Elencadas as limitações das taxonomias de representação de regras de negócio, esta dissertação visa definir conceitos que preencham essa lacuna de representação através da extensão da ontologia de regras de negócio, apresentada no capítulo a seguir.

4 DecKiPO – Extensão da ontologia de regras de negócio para cenários de KiP

Conforme apresentado no Capítulo 3, existem diversas linguagens de representação de regras de negócio, cada uma delas possuindo uma ênfase específica para descrição das restrições de um domínio. Apesar disso, nenhuma das linguagens atuais possuem uma representação precisa, especialmente no que se refere à perspectiva comportamental. Com isso, as regras de negócio podem não ser representadas como esperado, descrevendo comportamentos nem sempre condizentes com o esperado para o domínio, possibilitando a ocorrência de eventos que violam essas regras que regem a dinâmica do negócio.

As limitações de representação de aspectos temporais em relações entre eventos e a condição de inviolabilidade das regras de negócio na ontologia de regras de negócio são algumas das limitações apresentadas no capítulo anterior que trazem problemas em relação à modelagem de Processos Intensivos em Conhecimento (KiPs), especialmente quando se tratada da especificação de uma ontologia desse tipo de processo onde, ainda em tempo de modelagem, não se conhece ou não se pode determinar o tempo preciso em que um evento ocorreu ou deve ocorrer. Como o objetivo é permitir que a representação das regras de negócio no nível Independente de Computação (CIM) seja mais precisa, é necessário estender a ontologia de representação de regras de negócio.

Desta forma, este capítulo descreve a proposta extensão da Ontologia de Processos de Negócio (BRO) nomeada de DecKiPO, utilizando os diversos conceitos descritos até o presente capítulo, em especial os *templates* LTL Declare e os níveis de cumprimento da SBVR. Na Seção 4.1 são apresentados os conceitos incorporados à KiPO que definem e explicam em alto nível os diversos elementos estendidos a partir do LTL Declare e SBVR; a Seção 4.2 apresenta em detalhes os *templates* do LTL Declare de Existência, Relação e Escolha, que adicionam conceitos semânticos muito mais abrangentes principalmente em relação às restrições impostas a execução de atividades, permitindo que o fluxo de atividades em processos intensivos em conhecimento passem poder ser descritos em função das restrições do domínio ao invés do sequenciamento passo-a-passo inerentes a processos imperativos; por fim, na Seção 4.3 é apresentada a

extensão que adiciona os elementos e a semântica dos níveis de cumprimento das regras de negócio, considerando as situações onde essas regras são violadas durante a execução de instâncias de KiPs.

4.1 Proposta dos novos elementos

Conforme descrito na Seção 2.5, uma das perspectivas da ontologia KiPO é a de regra de negócio, estruturada como a sub-ontologia BRO. Esta perspectiva descreve três tipos de regras de negócio definidas por WAGNER (2004): *regras de integridade*, *regras de derivação* e *regras de reação*. Esses três tipos de regra se enquadram em diferentes classificações da SBVR (2017), onde regras de negócio podem ser definidas como *Definitional Business Rule*, quando restringe um aspecto estrutural do domínio ou como *Behavioral Business Rule*, quando restringe um aspecto comportamental do domínio.

As Regras de Integridade possuem as mesmas características observadas nas Regras de Derivação, segundo a definição de WAGNER (2004). Elas restringem aspectos estruturais de elementos do domínio, onde é possível concluir que ambas as regras podem ser classificadas como uma *Definitional Business Rul*e da SBVR.

As Regras de Reação, por outro lado, determinam que um novo evento seja disparado a partir de pré-condições alcançadas, influenciando diretamente o comportamento de um processo de negócios e alterando o estado das coisas (WAGNER, 2004). Essa definição se enquadra no contexto abrangido pelas *Behavioral Business Rule* da SBVR, cujas regras criam uma situação de obrigação durante a execução de instâncias do processo de negócio.

Para expandir a expressividade semântica da KiPO mantendo a sua compatibilidade com as definições e semântica descritas pelos elementos da SBVR, os conceitos apresentados anteriormente são adicionados à Ontologia de Regras de Negócio, explicitando que os conceitos *Foundational Integrity Rule* e *Derivation Foundational Rule* são de fato especializações de *Definitional Business Rule* da SBVR e que a *Foundational Reaction Rule* é especialização de *Behavioral Business Rule*. A Figura 23 ilustra as classes adicionadas à BRO.

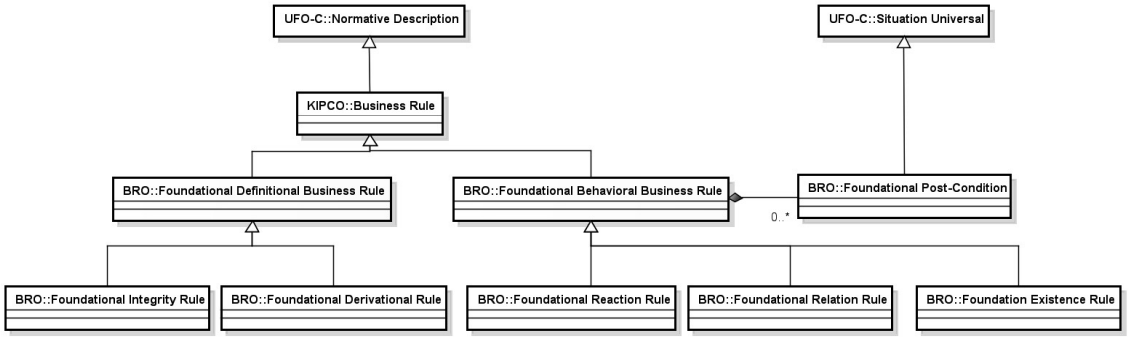

Figura 23 - Perspectiva da BRO com os novos conceitos da SBVR

A adição desses conceitos permite que a KiPO represente um conjunto de regras comportamentais mais abrangente, conforme descrito ao longo da dissertação.

4.2 Restrições com LTL Declare

Conforme apresentado na Seção 2.2 - Paradigma imperativo e declarativo, o metamodelo do LTL Declare permite a representação das relações entre atividades utilizando o formalismo da lógica temporal linear para representar restrições de existência, de relação e de escolha, adotando o paradigma declarativo para modelar o domínio. O LTL Declare, entretanto, relaciona apenas as atividades possíveis de serem executadas no domínio, sem relacioná-las com outros elementos inerentes a Processos intensivos em Conhecimento (como os agentes, as interações realizadas entre eles e demais aspectos que envolvem a tomada de decisão em um processo dessa natureza).

Esse potencial representativo de LTL Declare foi utilizado como base para a criação de novos elementos na KiPO, de forma a tornar a ontologia de Processos intensivos em Conhecimento capaz de representar um espectro maior de relações entre atividades e regras de negócio que não eram possíveis com os atuais elementos existentes na Ontologia de Regras de Negócio.

Com o objetivo de manter a organização disposta pela LTL Declare, os novos elementos foram organizados conforme a natureza das suas relações, sendo então classificados como **Regras de Restrição de Existência**, **Regras de Restrição de Relação** e **Regras de Restrição de Escolha**.

Como descrito em (PESIC, SCHONENBERG e VAN DER AALST, 2007), as relações propostas pelos *templates* LTL Declare são compostas por um ou mais elementos

atribuídos a uma semântica específica descrita em Lógica Temporal Linear e uma representação gráfica. Nesta dissertação, a representação gráfica dos elementos descritos não será abordada. A KiPO pode ser utilizada como metamodelo para o desenvolvimento de uma linguagem para representação de KiPs e, portanto, trabalhos futuros podem ser desenvolvidos para a criação de uma sintaxe concreta dessa linguagem. A semântica do relacionamento dos templates LTL Declare com atividades, bem como os axiomas descritos em lógica temporal linear, foram considerados para a extensão proposta na DecKiPO.

Dada a natureza da proposta apresentada neste trabalho, pode haver confusão acerca dos elementos citados ao longo das seções seguintes, pois a nomenclatura utilizada na definição dos novos elementos dessa proposta são as mesmas utilizadas pelos *templates* LTL Declare e pela SBVR. Assim, com o intuito de facilitar o entendimento, o seguinte padrão é adotado nas próximas seções:

- *Templates* LTL Declare – Quando for necessário realizar citações às restrições propostas por PESIC, SCHONENBERG e VAN DER AALST (2007), descritas na Seção 2.2, a referência a esses elementos adotará o padrão gráfico *em itálico* e respeitará a notação das fórmulas apresentadas na seção supracitada. Exemplo: O *template response(A,B)*;
- Elementos da DecKiPO – Quando for necessário realizar citações aos novos elementos desenvolvidos nessa proposta, será adotado o padrão gráfico **em negrito**. Como esses elementos tratam-se de conceitos adicionados à ontologia, eles serão sempre referenciados como Classes da Ontologia. Exemplo: **Classe Existence**.

Alinhados os conceitos, a seguir apresentada cada uma das novas classes desenvolvidas na extensão da KiPO.

4.2.1 Restrições de Existência

As restrições de existência são aquelas que delimitam a necessidade da existência (ou inexistência) da execução de um evento durante a instância de um processo de negócio. Estruturalmente, esse grupo de restrições remete a conceitos de integridade mas, diferentemente das Regras de Integridade descritas por WAGNER (2004) e utilizadas por

LOPES (2010) para fundamentação de sua Ontologia de Regras de Negócio (BRO), as restrições de Existência restringem o comportamento do domínio.

Em LTL Declare, as restrições de Existência são representadas por cinco *templates*: *init(A), existence$_n$(A)*, *absence$_n$(A), exactly$_n$(A) e last(A)*. Para que possam ser utilizados na modelagem de um Processo intensivo em Conhecimento, é necessário adicionar à KiPO elementos que definam as características principais desses *templates*. Porém, neste trabalho, somente os *templates existence$_n$(A)*, *absence$_n$(A)* e *exactly$_n$(A)* são considerados para a extensão DecKiPO.

Os templates *init(A)* e *last(A)* não são considerados pois possuem a mesma semântica dos elementos *StartEvent* e *EndEvent* da *Business Process Ontology* (BPO) que indicam, respectivamente, o evento que inicia e o evento que encerra o processo de negócio. Portanto, considerá-los nesta extensão causaria uma redundância de conceitos na ontologia. Outro problema ao incluir os templates *init(A)* e *last(A)* do LTL Declare seria a contradição que poderia haver em modelos de processos que utilizam a KiPO. Enquanto podem haver diversos *StartEvents* em um modelo de processo com condições de disparo distintos (OMG, 2011), só pode haver uma única atividade com o template *init(A)* em um modelo LTL Declare (PESIC, 2008). Esse tipo de contradição adiciona inconsistência ao metamodelo da KiPO, trazendo problemas para modelagem e inferência de modelos de processos de negócio representados pela DecKiPO.

Desta forma, os *templates existence$_n$(A)*, *absence$_n$(A)* e *exactly$_n$(A)* são adicionados à ontologia DecKiPO como especializações do conceito *Foundational Existence Rule,* que abrange as definições e características dos elementos da restrição de Existência do LTL Declare, conforme ilustra a Figura 24.

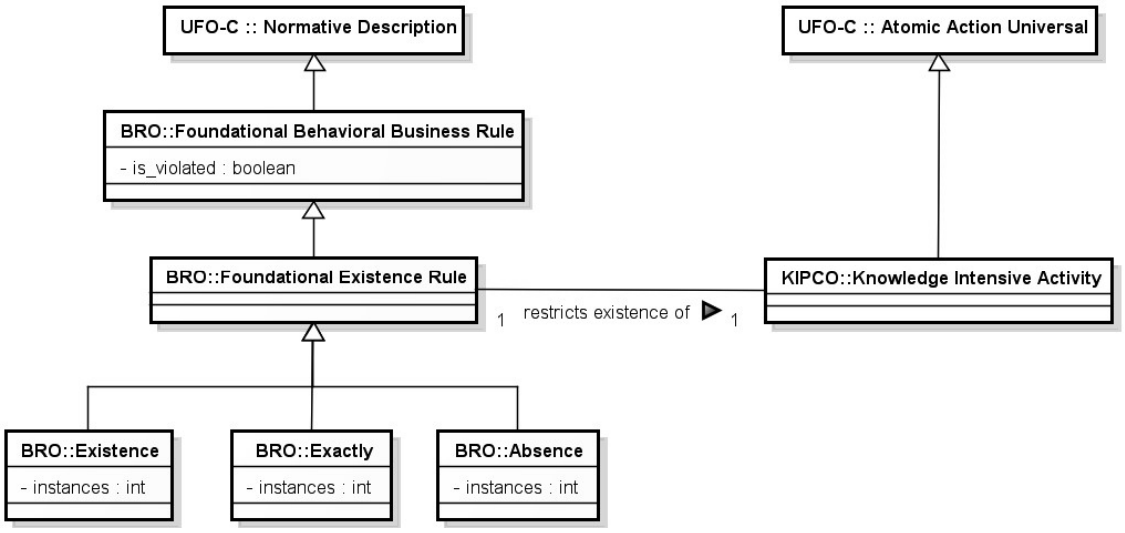

Figura 24 – *Os templates existence$_n$(A), absence$_n$(A)* e *exactly$_n$(A)* do LTL Declare como especializações de *Foundational Existence Rule* na DecKiPO

As restrições de Existência do LTL Declare definem, de modo geral, a necessidade de execução ou não das atividades em uma instância de processo. Esse comportamento pode se assemelhar, em um primeiro momento, à restrição imposta pelas Regras de Integridade definidas por WAGNER (2004) pois não alteram o estado das coisas em um processo. De fato, WAGNER propõe a existência de dois tipos de regras de integridade em seu trabalho: as restrições do tipo estrutural (voltadas à delimitar a necessidade de atributos de elementos do domínio, como a necessidade de uma identificação para um carro) e as restrições de processos (que se referem à integridade dinâmica do sistema e restringem as transições admissíveis entre os estados do sistema) (WAGNER, 2004).

Entretanto, a revisão bibliográfica realizada para este trabalho não encontrou referências mais extensas ao segundo tipo de regra de integridade (de processos). LOPES (2010) também não faz menção ao segundo tipo de regra, focando os exemplos em restrições do primeiro tipo (estruturais). Portanto, para efeitos deste trabalho, foi considerado que as regras de integridade presentes na KiPO se referem aos aspectos puramente estruturais do domínio.

Todos os conjuntos de restrições descritas pelo LTL Declare buscam impor uma relação de integridade ao domínio, seja por exigir a execução de certas atividades, seja por definir a relação existente entre elas. Porém, diferentemente das Regras de Integridade da BPO e descritas por WAGNER (2004), as regras de negócio modeladas a partir do

LTL Declare restringem o **comportamento** do domínio, ao invés de um aspecto **estrutural** de algum elemento do domínio.

Devido a essa característica de restringir o comportamento do domínio, a DecKiPO representa as Restrições de Existência do LTL Declare como especializações de uma regra de negócio fundamental, a *Foundational Existence Rule* que, além de definir características de possiblidade de violação da regra (como descrito no início deste capítulo), especifica atributos e a relação que todas as demais regras de negócio criadas a partir do LTL Declare terão.

A seguir é descrita em detalhes cada uma das restrições de Existência do LTL Declare e suas respectivas representações na DecKiPO.

4.2.1.1 Template existence

A relação descrita pelo *template existence$_n$(A)* do LTL Declare, presume que, na execução de uma instância de um processo, existirá pelo menos uma quantidade **n** de instâncias da atividade A. Na DecKiPO, cada *Foundational Existence Rule* restringe a existência de (*restricts existence of*) exatamente uma Atividade intensiva em Conhecimento (KiA). Por exemplo, considere a **Regra-LC1** modelada na Figura 25.

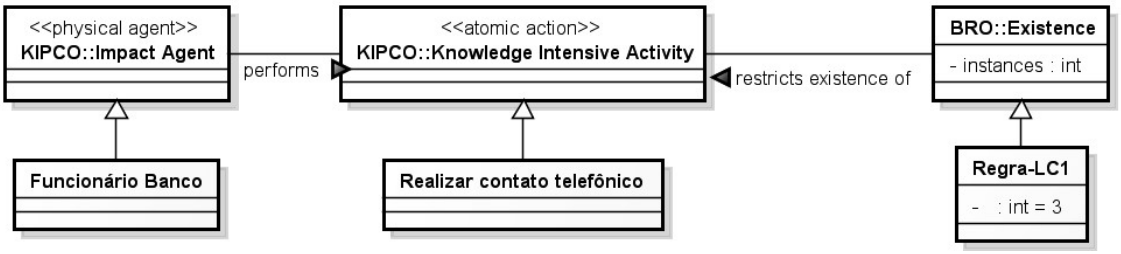

Figura 25 – Exemplo de representação do template e*xistence$_n$(A)* na DecKiPO

De acordo com o *template existence$_n$(A)* no LTL Declare, as restrições de existência podem ter diversas cardinalidades, sendo apresentados nos diversos trabalhos sobre LTL Declare (PESIC, SCHONENBERG e VAN DER AALST, 2007), (VAN DER AALST, PESIC e SCHONENBERG, 2009) os *templates existence(A),existence$_2$(A),existence$_3$(A) e existence$_n$(A)*, cuja formalização foi discutida no Capítulo 2.2 deste trabalho.

Como o *template existence$_n$(A)* representa quaisquer cardinalidades da relação de existência, todos esses *templates* estão representados na DecKiPO pela **Classe**

Existence, cuja propriedade *instances* determina o número mínimo de instâncias esperadas da KiA associada no âmbito da instância do processo.

A restrição de existência é definida formalmente através dos seguintes axiomas:

Axioma A9: $Existence(a, p, 1) \leftrightarrow \forall_{ip}(\Diamond \exists_{ia} KiA(a) \wedge KiP(p) \wedge instanceof(ia, a) \wedge instanceof(ip, p) \wedge partof(ia, ip))$

Axioma A10: $Existence(a, p, n) \leftrightarrow \forall_{ip}(\Diamond \exists_{ia} KiA(a) \wedge KiP(p) \wedge instanceof(ia, a) \wedge instanceof(ip, p) \wedge partof(ia, ip) \wedge \circ Existence(a, p, n-1))$

4.2.1.2 Template absence

O *template absence$_n$(A)* proposto pelo LTL Declare descreve um tipo de restrição de existência onde a atividade A não precisa ser executada ou pode ser executada, no máximo, **n** vezes ao longo da execução da instância de um processo P. Assim como o *template existence$_n$(A)*, o *template absence$_n$(A)* possui um meta-atributo que determina a quantidade máxima de instâncias esperadas de uma determinada atividade. Os *templates absence$_2$(A)* e *absence$_3$(A)* são descritos em diversos trabalhos relacionados (PESIC, SCHONENBERG e VAN DER AALST, 2007) mas, como o *template absence$_n$(A)* representa quaisquer números de instâncias possíveis, todos esses *templates* estão representados na KiPO pela **Classe Absence**, formalmente descrita através do seguinte axioma:

Axioma A11: $Absence(a, p, n) \leftrightarrow !\, Existence(a, p, n) \wedge n > 1$

4.2.1.3 Template exaclty

Ambas as restrições anteriores delimitam um limite inferior ou superior de instâncias de uma determinada atividade que deve ser atingido ou não pode ser ultrapassado. Para delimitar situações onde existem regras de negócio que especifiquem um número exato de instâncias que devem ser observadas, o LTL Declare descreve o *template exactly$_n$(A)*, onde A é uma atividade que deve ser instanciada exatamente **n** vezes ao longo da execução da instância do processo P.

A **Classe Exactly** foi adicionada à DecKiPO como uma especialização das *Foundational Existence Rule* e é definida formalmente através do seguinte axioma:

Axioma A12: $Exaclty(a, p, n) \leftrightarrow Existence(a, p, n) \wedge Absence(a, p, n+1)$

4.2.2 Restrições de Relação

As restrições de relação são aquelas que definem um tipo de relacionamento entre dois eventos. São um conjunto de *templates* LTL Declare de relacionamento binário, onde um evento A estará ligada a um evento B através de uma relação que contém uma semântica especificada em lógica temporal linear (PESIC, SCHONENBERG e VAN DER AALST, 2007).

Conforme descrito no Seção 2.2, essas restrições estão segmentadas em *responded existence, co-existence, response, precedence, succession, alternate response, alternate precedence, alternate succession, chain response, chain precedence e chain succession.* Além disso, as relações também possuem sua forma negada, conforme descrito na Seção 2.2.

Os *templates* de restrição de relação possuem semântica semelhante à da Regra de Reação (*Reaction Rule*) da KiPCO. Conforme descrito na Seção 2.2, esses *templates* representam restrições onde um evento A é executado antes ou depois de um evento B, dependendo da restrição adotada.

Porém, a Regra de Reação (*Reaction Rule*) da KIPCO não consegue representar por completo a semântica das restrições de relação do LTL Declare pois não considera o aspecto temporal presente nos *templates* de relação do LTL Declare. Por exemplo, considere o metamodelo em LTL Declare representado na Figura 26, extraído do log de eventos de atendimentos de uma empresa de *helpdesk*:

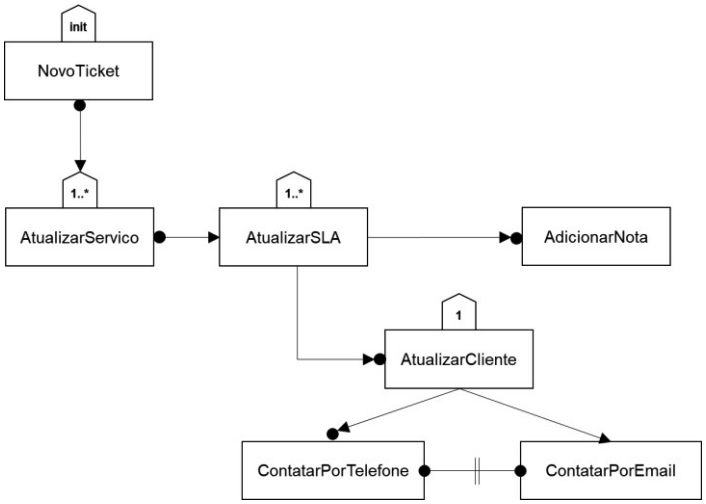

Figura 26 – Modelo LTL Declare de um atendimento helpdesk

Com base nos conceitos já discutidos na Seção 2.2, o modelo declarativo apresentado na Figura 26 descreve o processo iniciando pela atividade *NovoTicket*, sendo eventualmente sucedida pela execução da atividade *AtualizarServico* conforme representado pela restrição *response*. Em seguida, o *template chain response(A,B)* denota que a atividade *AtualziarSLA* deve ser executada logo após o término da execução da atividade *AtualizarServico*.

Considerando que este trabalho visa estender ontologias de regras de negócio fundamentando seus elementos na UFO e que o LTL Declare formaliza suas relações utilizando Lógica Temporal Linear (LTL), as relações de Allen descritas na UFO-B são os elementos necessários para fundamentar e prover a semântica necessária para representação dessas relações na ontologia de regras de negócio.

Conforme descrito na Seção 2.1.2, as relações de Allen provém a semântica de relações entre duas atividades em instantes de tempo definidos, abrangendo conceitos como as de antecedência (*before*), concorrência (*overlaps*) e encontro (*meets*). Os axiomas que definem as relações de Allen foram descritos por GUIZZARDI (2013) e são adotadas amplamente para construção dos axiomas dos elementos propostos neste trabalho. Ainda assim, as relações de Allen por si só não satisfazem as restrições de relação, pois existe uma relação condicional de existência entre as atividades, conforme apresentado na Seção 2.1.2 que não são consideradas nas relações de Allen.

Dessa forma, a adição de novas entidades à ontologia de regras de negócio, com suas respectivas axiomatizações, é necessária para representar a semântica dos *templates* LTL Declare.

Com exceção dos *templates co-existence* e *responded existence*, todos os *templates* de relação de restrição do LTL Declare são especializações dos *templates precedence*, *response* e *succession*.

Para representar essas três Restrições de Relação, foram adicionados à Ontologia de Regras de Negócio (BRO) as **Classes Precedence**, **Response** e **Succession** e suas formas negativas, **Not Precedence**, **Not Response** e **Not Succession**, que mantêm a relação binária entre duas *Knowledge-intensive Activities (KiA)* através dos relacionamentos *triggering event* e *resulting event* herdados. A Figura 27 ilustra a representação de todos os *templates* de relação na DecKiPO.

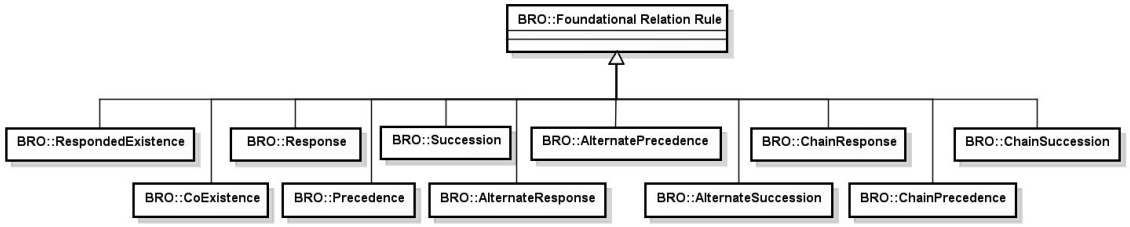

Figura 27 – Metamodelo da DecKiPO das *Foundational Relation Rules*

4.2.2.1 *Templates response, alternate response e chain response*

O *template response(A,B)* caracteriza uma relação onde sempre que uma atividade A for executada, uma atividade B eventualmente será executada em seguida, sendo importante perceber que nunca haverá uma instância da atividade A sem uma instância da atividade B ou isso seria uma violação do domínio. Neste tipo de relação, porém, uma atividade B pode ser instanciada livremente sem que exista uma instância da Atividade A.

Na UFO-B a relação de Allen *before* é a que melhor representa a relação temporal definido pelo *template response(A,B)*. De forma a se manter a ontologia bem fundamentada, a classe foi estendida da relação de Allen *before* da UFO-B, herdando as características de um *Time Interval Relation*, permitindo que sejam definidas atividades de origem (*source*) e destino (*target*) descritas na ontologia de fundamentação e mantendo a relação binária do *template* LTL Declare.

A formalização lógica da relação de Allen *before* foi apresentada na seção 2.1.2 e identificada pelo **Axioma A2**.

O axioma das **Classes Response** e **Not Response** estão definidos a seguir:

Axioma A13: $Response(a,b,p) \leftrightarrow \forall_{ip}(\Box\forall_{ia} KIA(a) \wedge KIP(p) \wedge instanceof(ia,a) \wedge instanceof(ip,p) \wedge partof(ia,ip) \rightarrow \Diamond\exists_{ib,b} KIA(b) \wedge instanceof(ib,b) \wedge partof(ib,ip) \wedge before(ia,ib))$

Axioma A14: $NotResponse(a,b,p) \leftrightarrow \forall_{ip}(\Box\forall_{ia} KIA(a) \wedge KIP(p) \wedge instanceof(ia,a) \wedge instanceof(ip,p) \wedge partof(ia,ip) \rightarrow \neg(\Diamond\exists_{ib,b} KIA(b) \wedge instanceof(ib,b) \wedge partof(ib,ip) \wedge before(ia,ib)))$

O *template alternate response(A,B)* caracteriza uma relação onde, considerando duas atividades, A e B, haverá uma restrição do tipo *response(A,B)* entre elas mas a atividade A não poderá ser executada novamente enquanto a atividade B não foi

executada (PESIC, 2008). Esse template será representado na DecKiPO através da **Classe AlternateResponse**, cujo axioma está definido a seguir:

Axioma A15: $AlternateResponse(a,b,p) \leftrightarrow \forall_{ip}(Response(a,b,p) \wedge \square(\forall_{ia} KIA(a) \wedge KIP(p) \wedge instanceof(ia,a) \wedge instanceof(ip,p) \wedge partof(ia,ip) \rightarrow \circ\,(Precedence(b,a,p)))$

O *template chain response(A,B)* caracteriza uma relação onde, considerando duas atividades, A e B, haverá uma restrição do tipo *response(A,B)* entre elas mas a atividade B é executada imediatamente após a atividade A ser concluída (PESIC, 2008). Assim como no *template response(A,B)*, para fundamentar na UFO os *templates* do LTL Declare adicionados à ontologia, é necessário recorrer à semântica descrita por Allen (1983). Como o *template* c*hain response(A,B)* condiciona o término de uma Atividade A ao início da Atividade B, o intervalo de tempo *meets*, formalizada pelo **Axioma A3** na seção 2.1.2, pode ser utilizado na formalização da relação para garantir que a semântica do elemento na LTL Declare seja mantido na DecKiPO.

Sendo assim, o *template chain response(A,B)* e *not chain response(A,B)* do LTL Declare serão representados na DecKiPO através das **Classes Chain Response** e **Not Chain Response**, cujos axiomas estão definidos a seguir:

Axioma A16: $ChainResponse(a,b,p) \leftrightarrow \forall_{ip}(Response(a,b,p) \wedge \square(\forall_{ia} KIA(a) \wedge KIP(p) \wedge instanceof(ia,a) \wedge instanceof(ip,p) \wedge partof(ia,ip) \rightarrow \circ\,(\exists_{ib} KIA(b) \wedge instance(ib,b) \wedge partof(ib,ip) \wedge meets(ia,ib))$

Axioma A17: $NotChainResponse(a,b,p) \leftrightarrow \forall_{ip}(\square(\forall_{ia} KIA(a) \wedge KIP(p) \wedge instanceof(ia,a) \wedge instanceof(ip,p) \wedge partof(ia,ip) \rightarrow \circ\,(\neg\exists_{ib} KIA(b) \wedge instance(ib,b) \wedge partof(ib,ip) \wedge meets(ia,ib))$

4.2.2.2 *Templates precedence, alternate precedence e chain precedence*

Enquanto o *template response(A,B)* representa a dependência de uma Atividade A com a Atividade B, o *template precedence(A,B)*, denota a independência da Atividade A em relação à Atividade B, pois esta relação caracteriza-se por relacionar a existência de uma instância da Atividade A sempre que uma Atividade B é executada (PESIC, SCHONENBERG e VAN DER AALST, 2007).

Para representar esse *template* e sua forma negativa, foram adicionadas à DecKiPO as **Classes Precedence** e **Not Precedence**, utilizando como fundamentação a relação temporal *before* da UFO-B e estendendo sua semântica para incluir a dependência existencial proposta por este *template* LTL Declare. Para fornecer a semântica necessária à correta representação das **Classes Precedence** e **Not Precedence**, os axiomas a seguir foram adicionados à ontologia:

Axioma A18: $Precedence(a,b,p) \leftrightarrow \forall_{ip}(\square\forall_{ib} KIA(b) \wedge KIP(p) \wedge instanceof(ib,b) \wedge instanceof(ip,p) \wedge partof(ib,ip) \rightarrow \Diamond\exists_{ia} KIA(a) \wedge instanceof(ia,a) \wedge partof(ia,ip) \wedge before(ia,ib))$

Axioma A19: $NotPrecedence(a,b,p) \leftrightarrow \forall_{ip}(\square(\Diamond\forall_{ib} KIA(b) \wedge KIP(p) \wedge instanceof(ib,b) \wedge instanceof(ip,p) \wedge partof(ib,ip) \rightarrow \neg\exists_{ia} KIA(a) \wedge instanceof(ia,a) \wedge partof(ia,ip) \wedge before(ia,ib))$

O *template alternate precedence(A,B)* caracteriza uma relação onde, considerando duas atividades, A e B, haverá uma restrição do tipo *precedence(A,B)* entre elas mas a atividade B não poderá ser executada novamente enquanto a atividade A não foi executada (PESIC, 2008). Esse template será representado na DecKiPO através da **Classe Alternate Precedence**, cujo axioma está definido a seguir:

Axioma A20: $AlternatePrecedence(a,b,p) \leftrightarrow \forall_{ip}(Precedence(a,b,p) \wedge \square(\forall_{ib} KiA(b) \wedge KiP(p) \wedge instanceof(ib,b) \wedge instanceof(ip,p) \wedge partof(ib,ip) \rightarrow \circ\,(Precedence(a,b,p)))$

O *template chain precedence(A,B)* caracteriza uma relação onde, considerando duas atividades, A e B, haverá uma restrição do tipo *precedence(A,B)* entre elas mas a atividade A é executada imediatamente antes da atividade B ser iniciada (PESIC, 2008).

Da mesma forma como o *template chain response(A,B)* recorre à relação temporal *meets* da UFO-B, as **Classes Chain Precedence** e **Not Chain Precedence**, que representam respectivamente os *templates chain precedence(A,B)* e *not chain Precedence(A,B)* na DecKiPO, também pode ser fundamentado utilizando essa relação de Allen, sendo representado através do axioma definido a seguir:

Axioma A21: $ChainPrecedence(a,b,p) \leftrightarrow \forall_{ip}(Precedence(a,b,p) \wedge \Box(\forall_{ib} KiA(b) \wedge KiP(p) \wedge instanceof(ib,b) \wedge instanceof(ip,p) \wedge partof(ib,ip) \rightarrow \exists_{ia} KiA(a) \wedge instanceof(ia,a) \wedge partof(ia,ip) \wedge meets(ia,ib))$

Axioma A22: $NotChainPrecedence(a,b,p) \leftrightarrow \forall_{ip}(\Box(\circ \forall_{ib} KiA(b) \wedge KiP(p) \wedge instanceof(ib,b) \wedge instanceof(ip,p) \wedge partof(ib,ip) \rightarrow \neg\exists_{ia} KiA(a) \wedge instanceof(ia,a) \wedge partof(ia,ip) \wedge meets(ia,ib))$

4.2.2.3 *Templates succession, alternate sucession e chain sucession*

O *template succession(A,B)* descrito pelo LTL Declare caracteriza uma relação de interdependência entre as Atividades A e B, ou seja, a execução da Atividade A sempre obrigará a execução posterior de ao menos uma instância da Atividade B.

Pela restrição imposta por este *template*, a Atividade B nunca poderá ser executada de forma independente, nem antes, nem depois da Atividade A. Isso implica que, depois de toda Atividade A, deve existir ao menos uma Atividade B e toda Atividade B deve der precedida por uma Atividade A, conforme descrito na Seção 2.2.

O *template succession(A,B)* pode ser representado como a conjunção dos *templates precedence(A,B)* e *response(A,B)*, bem como o *template not sucession(A,B)* pode ser representado com a conjunção dos templates *not precedence(A,B)* e *not response(A,B)*. Para representar esse *template* na Ontologia de Processos intensivos em Conhecimento, as **Classes Succession** e **Not Succession** foram incluídas na DecKiPO para representar essa restrição, sendo formalizadas pelos seguintes axiomas:

Axioma A23: $Succession(a,b,p) \leftrightarrow Response(a,b,p) \wedge Precedence(a,b,p)$

Axioma A24: $NotSuccession(a,b,p) \leftrightarrow NotResponse(a,b,p) \wedge NotPrecedence(a,b,p)$

O *template alternate succession(A,B)* caracteriza uma relação onde, considerando duas atividades, A e B, haverá uma restrição do tipo *succession(A,B)* entre elas mas a atividade A não poderá ser executada novamente enquanto a atividade B não foi executada (PESIC, 2008). Esse template será representado pela DecKiPO através da **Classe Alternate Succession**, cujo axioma está definido a seguir:

Axioma A25: $AlternateSucession(a,b,p) \leftrightarrow AlternateResponse(a,b,p) \wedge AlternatePrecedence(a,b,p)$

O *template chain succession(A,B)* caracteriza uma relação onde, considerando duas atividades, A e B, haverá uma restrição do tipo *succession(A,B)* entre elas mas a atividade B é executada imediatamente depois da atividade A ser iniciada (PESIC, 2008).

O *template chain succession(A,B)* é uma conjunção dos *templates chain response(A,B)* e *chain precedence(A,B)*, assim como o *template not chain succession(A,B)* é uma conjunção dos *templates not chain precedence (A,B)* e *not chain response(A,B)*, sendo representado pelas **Classes Chain Succession** e **Not Chain Succession** na DecKiPO e formalizados através dos axiomas definidos a seguir:

Axioma A26: $ChainSuccession(a, b, p) \leftrightarrow ChainResponse(a, b, p) \wedge ChainPrecedence(a, b, p)$

Axioma A27: $NotChainSuccession(a, b, p) \leftrightarrow NotChainResponse(a, b, p) \wedge NotChainPrecedence(a, b, p)$

4.2.2.4 *Template responded existence*

Algumas situações em processos de negócio demandam que um determinado recurso exista para que possa ser utilizado ao longo da execução da instância. Um exemplo é a criação de um *ticket* de *helpdesk* para que um atendimento de suporte seja realizado. Em alguns casos, o atendimento pode ser iniciado por uma conversa um técnico ou por telefone e o *ticket* ser criado somente depois do início do atendimento. O *ticket* é essencial para os objetivos de controle de negócio e sua existência é obrigatória e o processo deveria ser iniciado por ele, mas sua criação pode ser atrasada sem causar danos ao processo.

Essa situação é representada pelo *template responded existence(A,B)* onde se uma Atividade A for executada então uma Atividade B deve ter sido executada antes ou deve executar eventualmente depois. Essa relação dá flexibilidade à execução do processo enquanto restringe a dependência entre as atividades.

A definição do axioma para a **Classe Responded Existence** e de sua negativa, **Not Responded Existence,** não depende da utilização dos conceitos das relações de Allen da UFO-B visto que a restrição imposta exige apenas que a atividade de destino (*target*) seja executada eventualmente. O axioma básico definido pelo LTL Declare é $\Diamond A \rightarrow \Diamond B$ mas com se deseja manter a fundamentação dos novos elementos na UFO e associar as restrições LTL Declare às demais classes da KiPO, o axioma desta relação foi definida como:

Axioma A28: $RespondedExistence(a, b, p) \leftrightarrow \forall_{ip}(\Diamond\forall_{ia} KiA(a) \wedge KiP(p) \wedge instanceof(ia, a) \wedge instanceof(ip, p) \wedge partof(ia, ip) \rightarrow \Diamond\exists_{ib} KiA(b) \wedge instanceof(ib, b) \wedge partof(ib, ip))$

Axioma A29: $NotRespondedExistence(a, b, p) \leftrightarrow \forall_{ip}(\Diamond\forall_{ia} KiA(a) \wedge KiP(p) \wedge instanceof(ia, a) \wedge instanceof(ip, p) \wedge partof(ia, ip) \rightarrow \neg\Diamond\exists_{ib} KiA(b) \wedge instanceof(ib, b) \wedge partof(ib, ip))$

4.2.2.5 *Template coexistence*

O *template coexistence(A,B)* é a restrição de relação do LTL Declare que cria uma relação de dependência mútua entre dois eventos (ou atividades). Enquanto que no *template responded existence(A,B)* a Atividade B é independente, no *template coexistence(A,B)* sempre que uma Atividade A for executada, a Atividade B já deve ter sido executada ou tem que ser eventualmente executada, e vice-versa.

Para que essa relação seja possível de ser representada na Ontologia de Regras de Negócio, foi criada a **Classe Coexistence**, que herda suas características da relação temporal *before* da UFO-B, vinculando duas atividades através dos relacionamentos origem (*source*) e destino (*target*). Já sua negativa, a **Classe Not Coexistence** é uma conjunção do template *not responded existence(A,B)* e *not responded existence(B,A)*, uma vez que se há a restrição onde nenhuma das atividades podem ocorrer na mesma instância, uma delas não pode ter ocorrido nem antes nem depois da outra. Ambas as classes são formalizadas pelos axiomas a seguir:

Axioma A30: $Coexistence(a, b, p) \leftrightarrow RespondedExistence(a, b, p) \wedge RespondedExistence(b, a, p)$

Axioma A31: $NotCoexistence(a, b, p) \leftrightarrow NotRespondedExistence(a, b, p) \wedge NotRespondedExistence(b, a, p)$

4.2.3 Restrições de Escolha

As restrições de escolha são aquelas que limitam o número de alternativas que podem ser selecionadas pelo Agente dentre as atividades existentes no processo. Por exemplo, no processo de atendimento *helpdesk*, ao avaliar um *ticket* o técnico pode optar por realizar o chamado ou passar para outro técnico, dependendo de sua demanda. Esse tipo de

decisão é representado em LTL Declare como uma ação de escolha de uma dentre duas opções, conforme apresentado na Figura 28.

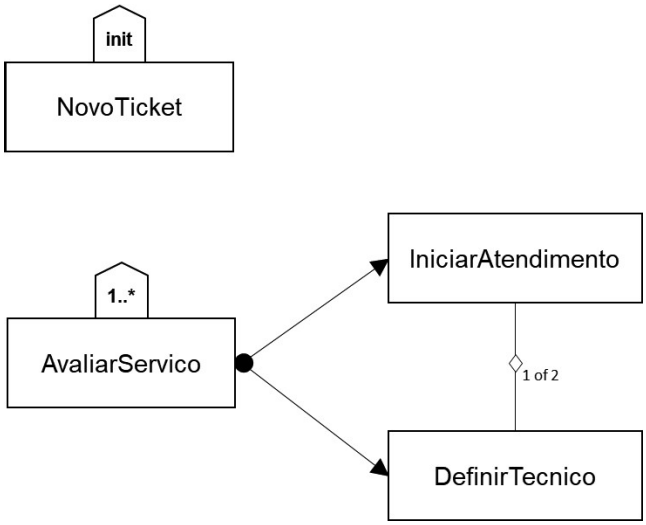

Figura 28 - Modelo Declare com restrições de escolha

As restrições de escolha são segmentadas em grupos, sendo as relações simples aquelas onde pelo menos uma das atividades dentre as opções apresentadas deve ser executada uma vez. É importante destacar que essa relação restringe apenas que deverá haver pelo menos uma instância dentre as opções disponíveis, não havendo qualquer restrição de que a outra opção de atividade também não possa ser executada. Essa relação será mais bem explicada posteriormente.

Já nas relações exclusivas, as atividades relacionadas são mutuamente exclusivas, ou seja, quando uma das atividades dentre as opções disponíveis for executada, não poderá haver instâncias das demais no processo.

Essas relações são representadas através dos *templates choice* e *exclusive choice*. Cada relação possui uma cardinalidade que define o número de escolhas possíveis dentre as opções disponíveis. Portanto, uma relação de escolha *choice 1 of 3 (A,B,C)* caracteriza que pelo menos uma dentre as três atividades deve ser executada ao menos uma vez.

4.2.3.1 Templates Choice e Exclusive Choice

O *template choice 1 of N* ($A_1..A_N$) denota a possibilidade de execução de pelo menos uma dentre as atividades relacionadas. Ela permite ao especialista visualizar as opções disponíveis em determinado instante do processo e escolher uma ou mais opções dentre as elencadas pela relação.

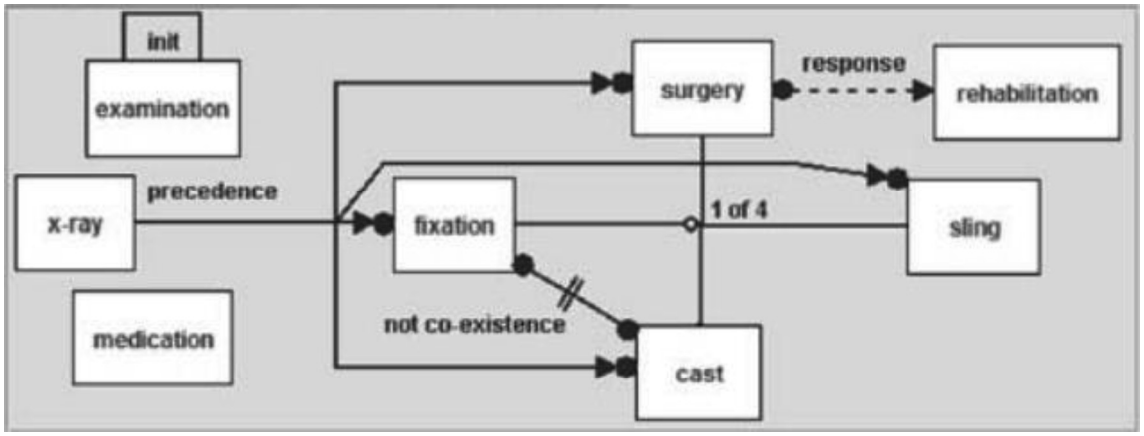

Figura 29 - Modelo declarativo em LTL Declare para tratamento de pacientes (VAN DER AALST, PESIC e SCHONENBERG, 2009)

A Figura 29 apresenta um modelo declarativo para tratamento de pacientes onde as atividades Tipóia (*sling*), Fixação (*fixation*), Cirurgia (*surgery*) e Gesso (*cast*) se apresentam como escolhas após a execução de Raio-X (*x-ray*). Assim, dependendo da avaliação do Raio-X (*x-ray*), existem disponíveis diversos tratamentos que podem ser selecionados de acordo com a gravidade do problema. A relação de escolha denota que ao menos uma das quatro possibilidades deve ser adotada, mas não exclui a possibilidade de adoção das outras. A única exceção no modelo ocorre entre as atividades Fixação (*fixation)* e Gesso (*cast*), que possuem uma restrição de *not-coexistence* conforme visto anteriormente, tornando-as mutuamente exclusivas.

Na literatura, escolha envolve tomada de decisão a partir do julgamento e escolha dentre um conjunto de alternativas apresentadas (SCHWARTZ, 2004). Essa definição se aproxima do conceito de Decisão (*Decision*) adotado pela KiPO onde decisão é o conjunto de ações que leva à escolha da melhor alternativa dentre as apresentadas (FRANÇA, NETTO, *et al.*, 2014).

Porém, para o LTL Declare, a relação de escolha denota apenas alternativas disponíveis e não envolve qualquer julgamento de valor. A avaliação e tomada de decisão ocorre a qualquer momento ao longo da execução do processo, em uma característica do modelo declarativo chamada de *decide to decide later* (PESIC, SCHONENBERG e VAN DER AALST, 2007). Dessa forma, a representação das atividades envolvidas na alternativa denota as possibilidades existentes e não o processo de decisão que leva o Agente a escolhê-las. Apesar disso, ao ser selecionada pelo Agente, o metamodelo LTL Declare considera que a atividade é executada imediatamente, avaliando quaisquer outras restrições que sejam aplicáveis a partir daquela atividade executada.

Entretanto na KiPO esse comportamento não é desejado, uma vez que a ontologia busca identificar todos os processos que levaram um Agente a optar por uma ou outra ação. Dessa forma os template c*hoice* e *exclusive choice* devem ser consideradas apenas como componentes de restrição do domínio e não como elementos de tomada de decisão.

Essa abordagem é adotada na KiPO de forma que as **Classes Choice** e **ExclusiveChoice** são adicionados à BRO através da DecKiPO, podendo ser consideradas em qualquer evento onde uma regra de negócio restrinja as escolhas possíveis em um determinado estado do processo.

Para deixar clara a diferença entre as **Classes Choice** e **ExclusiveChoice** da Ontologia de Regras de Negócio e a classe *Alternative* da Ontologia de Decisão (DO), é importante lembrar que o elemento da DO representa o conjunto de opções apresentadas pelo Agente de Inovação (*Innovation Agent*) analisando Vantagens (*Advantages*) e Desvantagens (*Disadvantages*) com base em Critérios (*Criterions*) estabelecidos possuindo, portanto, uma semântica muito mais abrangente do que a relação *choice*.

O *template choice 1 of N(A_1...A_N)*, por sua vez, representa as opções que podem ser adotadas em dado instante na execução da instância do processo, uma vez que Alternativas (*Alternative*) são *situations* abordadas pelo Agente (*Agent*) como possíveis linhas de ação para a tomada de decisão, que são efetivadas através da execução de uma ou mais Atividades (*Activity*) possíveis de serem escolhidas, restritas pela regra de negócio.

Dessa forma, na Ontologia de Regras de Negócio (BRO), as **Classes Choice** e **ExclusiveChoice** são entendidas como uma extensão das regras de negócio com a propriedade adicional de ser composta pelas atividades que podem ser consideradas como alternativas em uma tomada de decisão, permitindo que a ontologia seja bem mais clara quanto à representação de uma regra de negócio que restrinja tais opções. A Figura 30 representa a relação dos novos elementos do *template* de relações de escolha do LTL Declare com os demais elementos da KiPO.

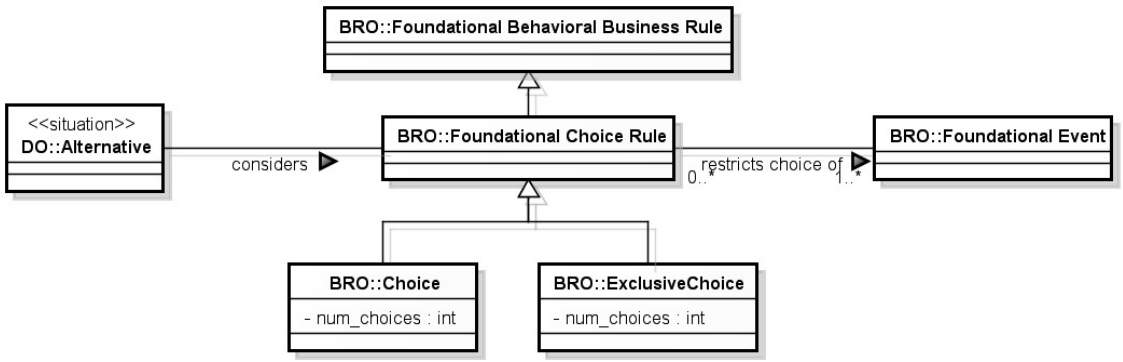

Figura 30 – Classes Choice e ExclusiveChoice na KiPO

Com a utilização de LTL Declare, podemos interpretar que uma restrição de escolha entre duas ou mais Atividades intensivas em Conhecimento (KiA) denota que, eventualmente, pelo menos uma das KiAs será executada no processo, ou seja, eventualmente uma instância de uma das atividades deverá existir na instância do processo.

Assim sendo, ambas Restrições de Escolha podem ser definidas em função das Restrições de Existência, em especial o *template* $existence_n(A)$. Na seção 2.2 foi apresentada a formalização das Restrições de Escolha conforme definido por PESIC (2008) mas os *templates* c*hoice* e e*xclusive choice* podem ser reescritas em função do **Axioma A9** das Restrições de Existência, já formalizada na seção 4.2.1.

Os axiomas que definem essas duas restrições estão descritos a seguir:

Axioma A32: $Choice(a, b, p) \leftrightarrow Existence(a, p, 1) \vee Existence(b, p, 1)$

Axioma A33: $ExclusiveChoice(a, b, p) \leftrightarrow not_coexistence(a, b, p)$

Exemplificando, a Figura 31 apresenta um cenário onde suponhamos que a unidade de tratamento somente realize cirurgia às quartas-feiras. Considerando a ontologia, o Agente poderia considerar as quatro opções de tratamento disponíveis na unidade mas também deveria considerar a regra de negócio que restringe a realização de cirurgias às quartas-feiras. Dessa forma, o Agente teria como alternativas apenas 3 das 4 opções disponíveis apesar da ontologia apresentá-las como opções de escolha.

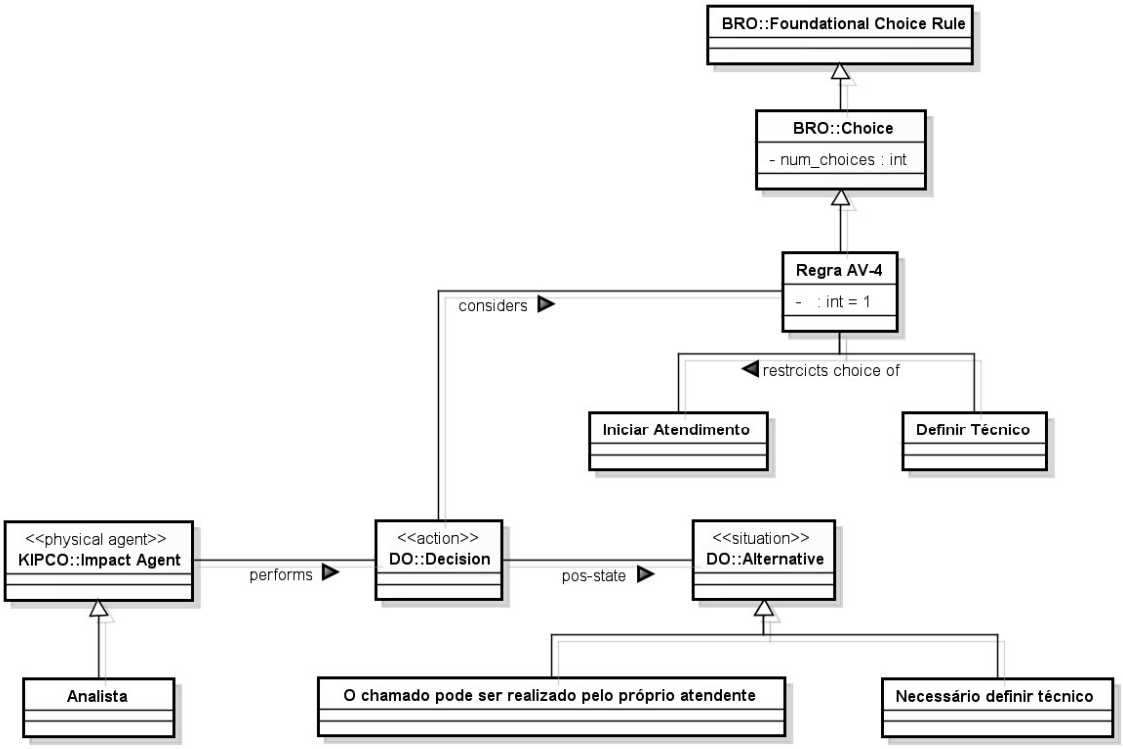

Figura 31 – Modelo do cenário de tratamento médico

Todas as relações descritas pelos *templates* LTL Declare e incorporadas à KiPO como extensões de Regras de Negócio presumem que as restrições propostas pelos elementos serão obedecidas nas instâncias dos processos. Porém regras de negócio podem ser violadas por diversos fatores e tais exceções precisam ser representadas de alguma forma na KiPO.

4.3 Níveis de cumprimento com SBVR

Especialmente em ambientes de Processos Intensivos em Conhecimento, a presença apenas de regras invioláveis diminui a capacidade representativa de qualquer modelo de processo em domínios dessa natureza sendo imprescindível, portanto, que o metamodelo que se propõe a descrever KiPs seja capaz de representar condições onde uma regra de negócio descrita possa, eventualmente, não ser cumprida.

Em SBVR, o conceito relativo às *Behavioral Business Rules* (OMG, 2017), presume que uma regra de negócio pode ser violada pelos agentes do domínio cabendo uma punição à essa transgressão dependendo do grau de violação que é atribuído à regra. Esse grau de violação é chamado de nível de cumprimento (*enforcement level*), um conceito originado na *Business Model Motivation* (BMM) que provê estruturas para o

desenvolvimento, comunicação e gerenciamento de planos de negócio, identificando elementos de um plano de negócio e como eles se relacionam (OMG, 2015).

Para que a Ontologia de Regras de Negócio (BRO) seja capaz de descrever regras de negócio que podem ser violadas, é necessário adicionar os conceitos de nível de cumprimento à ontologia e realizar as modificações necessárias para que regras deônticas sejam representadas.

4.3.1 Níveis de cumprimento

De acordo com SBVR (2017), níveis de cumprimento são valores em uma escala ordenada que especificam a severidade de consequência imposta de forma garantir o cumprimento de uma *Behavioral Business Rule*. A definição desses níveis, portanto, presume que o domínio mantenha a conformidade com a regra de negócio estabelecida e que sua violação implica em sanções baseadas no nível imposto.

Os níveis de cumprimento propostos pelo SBVR são dispostos por ordem crescente de gravidade na Tabela 4 e foram utilizados como referência para a definição das classes na Ontologia de Regras de Negócio (BRO).

Tabela 4 - Níveis de Cumprimento ordenados por gravidade (OMG, 2017)

Nível de Cumprimento	Definição
Strictly enforced	Se a regra for violada, uma penalidade será aplicada necessariamente.
Deferred enforcement	Se a regra for violada, uma penalidade será aplicada necessariamente, porém, o cumprimento da regra pode ser atrasado.
Pre-authorized override	Violações são permitidas desde que sejam previamente autorizadas.
Post-justified override	Violações são permitidas e devem ser aprovadas depois do fato ou penalidades serão aplicadas.
Override with explanation	Violações são permitidas sendo necessário apenas um comentário sobre o motivo da violação.
Guideline	Apenas uma sugestão, não sendo necessário o cumprimento.

Como apenas regras de negócio de negócio comportamentais (*behavioral business rule)* possuem a característica de obrigação ou possibilidade e tendo em vista que as regras fundamentais propostas por LOPES (2010) e utilizados pela Ontologia de Regras de Negócio (BRO) são do tipo alética, ou seja, não podem ser descumpridas, é necessário realizar modificações nos elementos propostos para permitir que a KiPO represente regras de negócio do tipo deôntica.

A solução proposta por este trabalho é modificar a atual estrutura da Ontologia de Regras de Negócio e reorganizar as *Foundational Business Rules* da BRO de forma que seja possível refletir melhor a gama de possibilidades de restrições possíveis em um domínio.

Para fundamentar na UFO o conceito de níveis de cumprimento é preciso analisar sua definição e relação com regras de negócio comportamentais. No âmbito da relação com regras de negócio, uma regra de negócio comportamental possui um e somente um nível de cumprimento associado (*behavioral business rule has enforcement level*) (OMG, 2017), sendo existencialmente dependente da regra de negócio ao qual está vinculada.

Sendo assim, é correto afirmar que o nível de cumprimento é uma propriedade inerente a regras de negócio comportamentais onde toda regra dessa natureza possui necessariamente um nível de cumprimento. Apesar da definição proposta pela SBVR descrever o relacionamento entre uma regra de negócio comportamental e o nível de cumprimento com o verbo *has*, consideramos que o termo é muito abrangente e, portanto, neste trabalho, utilizamos a relação *enforced by* para representar esse termo, cujo axioma é definido a seguir:

Axioma A34: $\forall br(FoundationalBehavioralBusinessRule(br) \rightarrow \exists! e\ EnforcementLevel(e) \wedge enforcedBy(br, e) \wedge SeverityLevel(sl) \wedge qualeOf(e, sl))$

Quanto à definição, níveis de cumprimento são valores em uma escala ordenada que especificam a severidade de consequência imposta de forma garantir o cumprimento de uma regra de negócio (OMG, 2017). Como o valor de um nível de cumprimento pode ser posicionado dentre um conjunto de valores pré-definidos, é possível vincular o nível de cumprimento ao conceito de *Quality Universal* da UFO-A.

Neste ponto temos definido que, na Ontologia de Regras de Negócio, o elemento *Enforcement Level* é um *Quality Universal* que está associado a uma *Behavioral Business Rule* através da relação *enforced by*. Um *Quality Universal* sempre está associado com estruturas de qualidade (*Quality Structures*) que podem ser definidas como uma base para a representação conceitual do conjunto de todos os valores possíveis que uma instância de *Quality Universal* pode assumir (ALBUQUERQUE e GUIZZARDI, 2013). Na DecKiPO, foi criada a **Classe Severity Level**, que representa as estruturas de qualidade definidas para cada um dos níveis de severidade descritos na SBVR: *strict*, *deferred*, *pre-authorized*, *post-justified*, *override* e *guideline* como demonstrado na Figura 32.

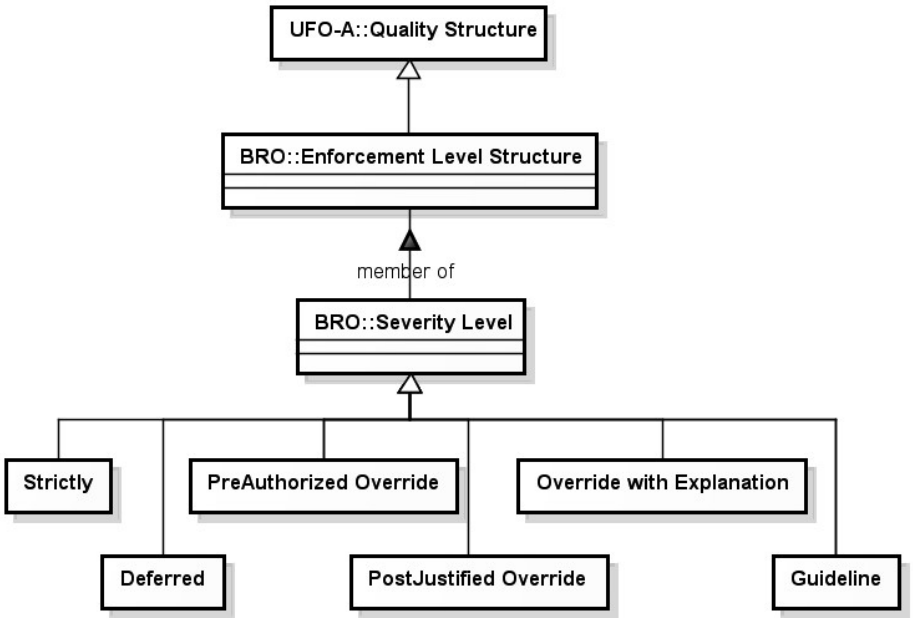

Figura 32 – Metamodelo dos Severity Levels na DecKiPO

É importante notar que um nível de cumprimento, a priori, é definido ou prescrito por um especialista do domínio, de modo que o grau de severidade de descumprimento de uma restrição é conhecido no âmbito do modelo de um processo intensivo em conhecimento. Entretanto, ao longo da execução das instâncias dos processos, uma regra pode vir a ser violada seja deliberadamente ou por desconhecimento da regra pelo Agente ou por este entender que a regra não era obrigatória. Nota-se, neste momento, que o nível de cumprimento associado a uma regra de negócio possui naturezas distintas quando se fala de modelo e de instância.

Normalmente, as regras de negócio e seus níveis de cumprimento descritos no modelo serão definidos manualmente por um especialista ou a partir da mineração de

processos e regras de negócio. Já o nível de cumprimento percebido pelo Agente na instância não é o mesmo daquele prescrito pelo especialista do domínio e, portanto, a semântica de ambos os conhecimentos é distinta. Para resolver este problema, é necessário recorrer ao conceito de *Quality* da UFO-A que representa uma concretização de um *Quality Universal* para um indivíduo. A DecKiPO propõe a definição de dois *Qualities*: *Conceived Enforcement Level* e o *Perceived Enforcement Level*.

O Nível de Cumprimento Concebido (*Conceived Enforcement Level*) representa o grau de cumprimento à regra em âmbito de modelo, geralmente definido por um especialista ou inferido através de ferramentas de mineração de processos e regras de negócio. Esse elemento é o responsável garantir a restrição das instâncias e definir as sanções caso a regra a ele vinculada seja violada.

O Nível de Cumprimento Percebido (*Perceived Enforcement Level*) representa o grau de cumprimento percebido pelos Agentes na instância do processo. Essa representação engloba não só a percepção equivocada do grau de cumprimento da regra na tomada de decisão como também a decisão deliberada de não a cumprir, mesmo que ela seja conhecida pelo Agente.

Na KiPO, um Agente (*Agent*) toma uma Decisão (*Decision*) considerando uma Restrição (*Restriction*) do domínio após optar por uma série de Alternativas (*Alternative*) apresentadas. Como demonstrado na Figura 33, as entidades Decisão e Alternativa consideram Regras de Negócio (*Business Rule*) e, portanto, estão suscetíveis às restrições impostas por esses elementos do domínio. Neste ponto, foi levantado o questionamento sobre se percepção da regra de negócio e seu grau de severidade ocorre no momento da análise das alternativas ou quando a decisão é efetivamente tomada pelo Agente.

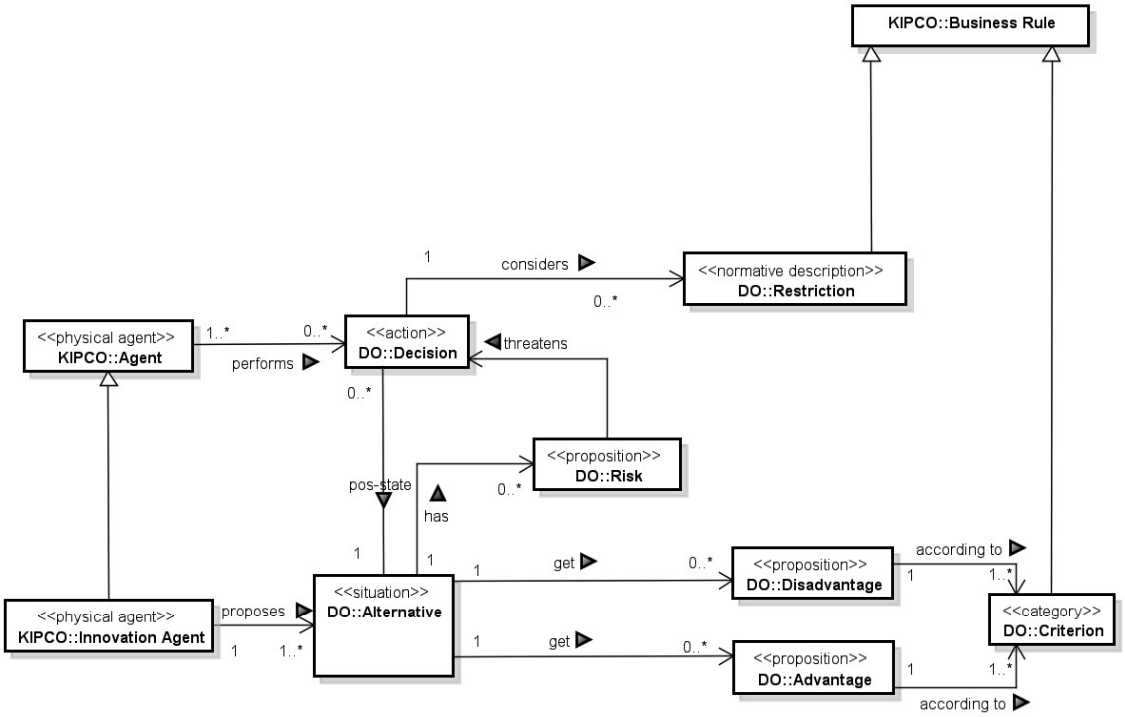

Figura 33 - Extrato do processo de tomada de decisão na KiPO (FRANÇA, NETTO, *et al.*, 2014)

PEREIRA e SANTORO (2010) afirmam que uma Alternativa representa as opções consideradas para tomada de decisão, identificando seus custos, tempo e recursos necessários assim como sua viabilidade. Essas alternativas podem simplesmente ser apresentadas como opções à tomada de decisão, sem que se leve em consideração eventuais regras de negócio que restrinjam seu comportamento ou adoção.

Cada alternativa pode ser avaliada pelo Agente sob a perspectiva de suas Vantagens (*Advantages*) e Desvantagens (*Disadvantages*) com base em Critérios (*Criterions*), momento no qual seria possível considerar que ele toma consciência das regras de negócio existentes para cada uma das opções.

Esses elementos são considerados pelo Agente durante o processo de tomada de decisão (FRANÇA, NETTO, *et al.*, 2014) concluindo-se, portanto, que a percepção das regras de negócio e de seu nível de cumprimento se concretiza neste momento, materializando o relacionamento *according to* que avalia os critérios estabelecidos pelas regras de negócio comportamentais, conforme demonstrado na Figura 34.

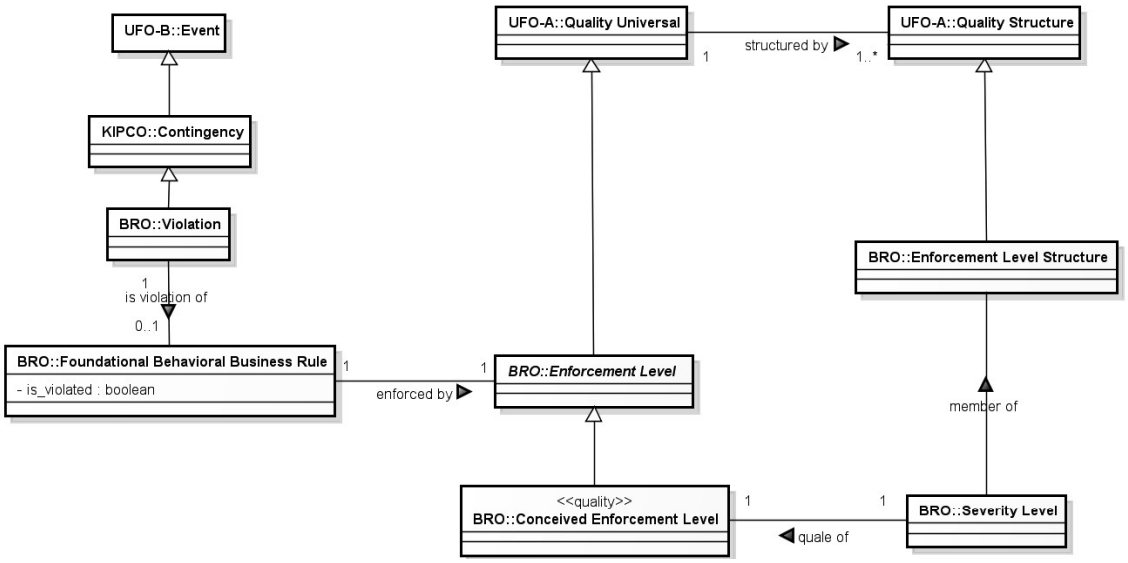

Figura 34 - Relação entre o Nível de Cumprimento Concebido (*Conceived Enforcement Level*) e os elementos da KiPO

Quanto à Decisão (*Decision*), o elemento na KiPO identifica informações relacionadas à tomada de decisão como um todo e não mais a cada alternativa em particular (FRANÇA, NETTO, *et al.*, 2014). Isso reforça a análise de que a percepção da regra de negócio acontece já na análise das alternativas e auxilia o processo decisório do Agente (*Agent*). Porém, na ação da Decisão, o Agente ainda deve considerar as regras de negócio que se apliquem ao estado atual do processo.

Na ação de tomada de decisão, portanto, o Agente (*Agent*) percebe a regra de negócio através na anuência de seu nível de cumprimento, materializada através da relação entre a tomada de decisão propriamente dita (*Decision*) pelo Agente (*Agent*) considerando a Regra de Negócio Comportamental (*Foundational Behavioral Business Rule*) a ser cumprida e o Grau de Severidade (*Severity Level*) percebido pelo Agente (*Agent*). Essa relação é modelada nessa proposta como um Nível de Cumprimento Percebido (*Perceived Enforcement Level*), representada na DecKiPO pela **Classe Perceived Enfocement Level**, conforme ilustrado na Figura 35.

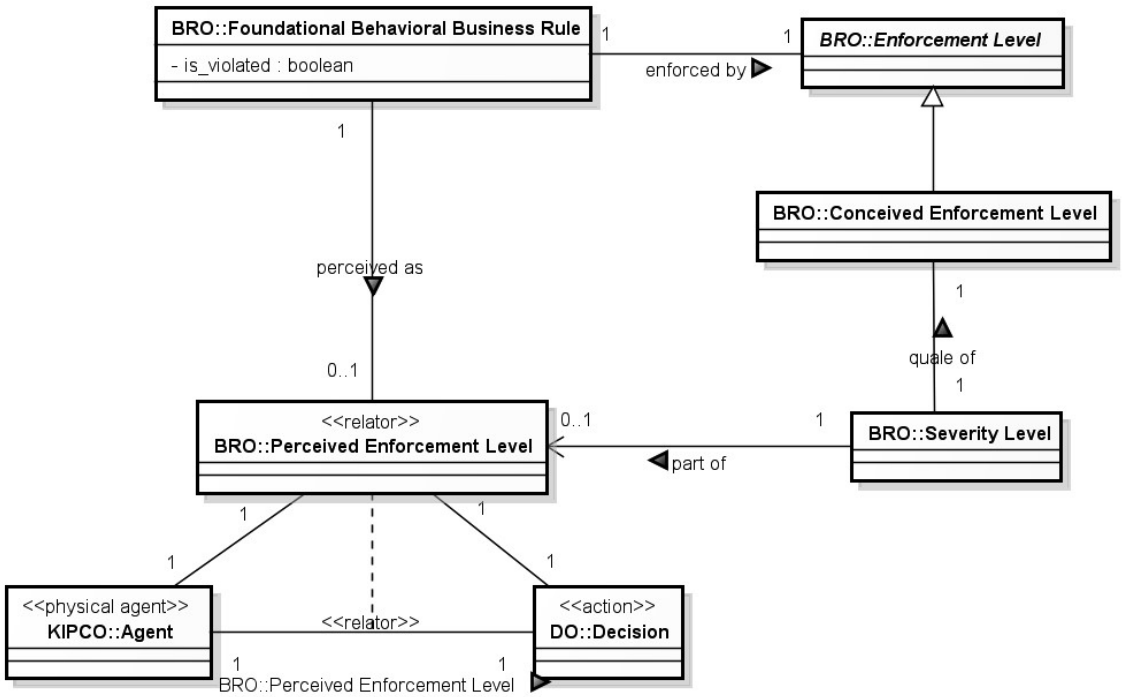

Figura 35 - *Perceived Enforcement Level* mediando a consideração da Regra de Negócio Comportamental (*Foundational Behavioral Business Rule*) para tomada de Decisão (*Decision*) pelo Agente (*Agent*)

Diferentemente dos demais elementos descritos até então, o Nível de Cumprimento Percebido (*Perceived Enforcement Level*) normalmente não será representado em um modelo de processo, visto que a percepção da ação ocorre somente durante a execução das instâncias.

Considerando que ao existir uma regra de negócio comportamental que possua um determinado grau de severidade associado a seu nível de cumprimento, espera-se que em algum momento na execução do processo essa regra de negócio seja percebida pelo Agente em uma tomada de Decisão.

Axioma A35: $\forall_{ip}(\exists_{ibr,cel} FoundationalBehavioralBusinessRule(ibr) \wedge ConceivedEnforcementLevel(cel) \wedge enforcedBy(ibr, cel) \rightarrow \Diamond\exists_{pel,a,d} PerceivedEnforcementLevel(pel) \wedge Decision(d) \wedge Agent(A) \wedge mediates(pel, a, d, cel)$

O nível de cumprimento percebido (*Perceived Enforcement Level*) da regra de negócio em uma tomada de decisão representa o consenso entre todos os agentes que participaram da ação. As percepções individuais dos agentes são representadas pelos Sentimentos (*Feeling*) ou Crenças (*Belief*) que influenciam na decisão.

Se uma regra de negócio descrita no modelo do processo não for percebida ou foi executada pelo Agente contrariando o nível de cumprimento concebido, então sanções e consequências podem ser impostas pela violação da regra, podendo ser deliberadamente definida por um especialista no domínio.

Porém, o evento causador da violação da regra pode ser imprevisível e, por esse motivo, não ser descrito em um modelo de processo, o que não o isenta de ser representado para que a análise futura da instância provenha um melhor entendimento sobre o que vem acontecendo no processo.

A violação de uma regra de negócio ocorre quando, durante a execução da instância de um processo, não é possível observar uma instância da regra de negócio obrigada por um nível de cumprimento. Conforme descrita pela SBVR, todas as regras de negócio comportamentais possuem um atributo que define se ela está em estado violada ou não (OMG, 2017). Uma regra de negócio é considerada violada se em qualquer estado das coisas (*state of affairs*) em que o nível de cumprimento da regra de negócio seja obrigatório, não existir uma instância dessa regra de negócio.

Para representar a mudança de estado, o axioma que define essa violação está descrito a seguir:

Axioma A36: $isViolated(br, ip) \leftrightarrow FoundationalBehavioralBusinessRule(br) \wedge ConceivedEnforcementLevel(cel) \wedge enforcedBy(br, cel) \wedge KiP(p) \wedge instanceof(ip, p) \wedge \nexists_{ib}\ (instanceof(ibr, br) \wedge partof(ibr, ip))$

Na Ontologia de Processos intensivos em Conhecimento (KiPO), o elemento Contingência (*Contingency*) trata justamente de eventos não previstos no domínio e este conceito é o que é necessário para tratar violações às regras de negócio. No caso das regras de negócio, a violação de uma regra é um evento de Contingência bastante característico cujo próprio significado da palavra permite ao agente ou envolvidos nos processos reconhecerem o problema ocorrido. Desta forma, para caracterizar eventos de transgressão às regras de negócio, foi criada a **Classe Violation** sendo uma especialização de *Contingency* permitindo caracterizar de forma clara o não-cumprimento de uma regra de negócio.

A Classe *Violation* relaciona-se diretamente com a regra de negócio comportamental (*Foundational Behavioral Business Rule*) que foi violada através do

relacionamento *is_violation_of*. Uma instância de uma regra de negócio pode ser violada zero ou mais vezes na execução de um processo enquanto que o evento que ocasionou a transgressão é único e, portanto, tem cardinalidade de um com a regra violada.

Além da violação poder ser caracterizada por um evento que contradiz a regra de negócio executado por um agente do processo, ela também pode se dar pela não execução de uma atividade ou comportamento esperado pelo domínio. Conforme descrito na Seção 4.2, as restrições de existência, relação e escolha adicionadas ao metamodelo da Ontologia de Regras de Negócio podem impor condições acerca do número de instâncias esperadas ou sequenciamento temporal das atividades. Essas relações podem não ser cumpridas pela instância do processo, o que resulta em violações do domínio, impostas pelo **Axioma 29** e, como definido até aqui, deveria causar consequências no processo de acordo com a severidade imposta à regra violada.

5 Avaliação e aplicabilidade da proposta

Este trabalho se propõe a expandir a capacidade de representação da Ontologia de Processos Intensivos em Conhecimento (KiPO), enfatizando a perspectiva de regras de negócio representada pela BRO para permitir que as relações temporais admissíveis entre tipos de atividades e as consequências de violações de regras de negócio sejam corretamente modeladas.

Para avaliar a proposta de solução descrita ao longo dos capítulos deste trabalho, duas abordagens de avaliação são utilizadas: a avaliação dos conceitos adicionados e a aplicabilidade desses conceitos em cenários de estudo.

A avaliação se baseia na avaliação da qualidade da ontologia baseada em critérios estabelecidos pela literatura, em especial as propostas por RAAD e CRUZ (2015) e as técnicas descritas por GUARINO (1998). A aplicabilidade da ontologia se dará através do desenvolvimento de experimentos baseados em cenários de estudo, que contém informações suficientes para validar as hipóteses que guiaram o desenvolvimento do trabalho. Ao longo deste capítulo são discutidas as decisões tomadas acerca do método de avaliação escolhido bem como realizada a apresentação dos cenários de estudo utilizados para atestar a avaliação dos resultados.

5.1 Cenários de estudo

5.1.1 Atendimento helpdesk

O primeiro cenário é uma empresa real, sediada na cidade do Rio de Janeiro, especializada em infraestrutura e em gestão e administração de ambientes de TI com foco em disponibilidade, segurança e desempenho. Para assegurar a qualidade dos serviços prestados, a empresa implanta processos de qualidade total e estabelece, juntamente com seus clientes, acordo de níveis de serviço (SLA) para atendimento rápido dos problemas identificados.

Todo o atendimento da empresa é realizado através de processos de *Service Desk*, sendo a maior parte do atendimento e contato entre cliente e empresa realizado

através de um software especializado em gestão de chamados de serviço. Cada demanda é registrada em um Ticket, que sempre é aberto com uma prioridade padrão definida como média quando uma solicitação é realizada pelo cliente por telefone ou e-mail direcionado a um dos técnicos responsáveis por realizar a triagem dos chamados. Estes técnicos avaliam e direcionam os Tickets aos especialistas nas diversas áreas cobertas pela empresa, a saber: infraestrutura, segurança, servidores, serviços de internet e suporte técnico geral.

Todas as interações realizadas pelos técnicos e especialistas são registradas no software na forma de comentários e quaisquer alterações na responsabilidade pelo atendimento são automaticamente comunicadas ao cliente via e-mail. Os chamados podem eventualmente sofrer uma alteração de prioridade, para maior ou menor, dependendo do tempo de atendimento decorrido desde a abertura do chamado e o nível de acordo de serviço contratado pelo cliente.

Os especialistas nas áreas têm autonomia para realizar diversas ações internas e externas, inclusive contatando diretamente o cliente, para tentar solucionar os problemas identificados. Entretanto, algumas ações devem passar pela aprovação do gerente da área na qual o especialista está alocado.

Ao final de cada atendimento, o especialista responsável deve encerrar o Ticket, ação que dispara automaticamente um e-mail para o cliente solicitante, que pode confirmar o encerramento do Ticket, fechando-o, ou solicitar reabertura caso o problema não tenha sido sanado.

A seguir, são descritas as regras de negócio identificadas no domínio e utilizadas ao longo dos exemplos deste trabalho:

Regra-HD1: Só pode haver, no máximo, 2 trocas de técnico ao longo do atendimento a um chamado

Regra-HD2: Os atendimentos a problemas informados por clientes devem ser solucionados em até 72 horas

Regra-HD3: Todo atendimento a problemas de clientes sempre deve ser identificado por um único Ticket

5.1.2 Liberação de crédito

O cenário baseia-se em dados fornecidos por uma instituição financeira alemã para o *Workshop Business Process Intelligence* 2017 (BPIC, 2017) que representa um processo de solicitação de empréstimo pessoal.

O processo é iniciado através da criação de uma solicitação por um cliente, que é enviada para o banco tanto via meios eletrônicos quanto pessoalmente, com um consultor do banco. Essa solicitação é avaliada por um Consultor, que oferece uma ou mais propostas de empréstimos para o cliente, que pode ou não aceitá-la.

Esse fluxo é repetido diversas vezes até que o cliente aceite uma proposta, podendo o Consultor entrar em contato com o Cliente de diversas maneiras, seja por contato telefônico, seja por e-mail. Quando um cliente aceita a proposta de empréstimo, o Consultor realiza uma análise completa da documentação do Cliente, buscando o atendimento a todos os requisitos do banco e analisando outras informações como pontuação de crédito do Cliente.

Estando os dados completos, o Consultor finaliza o processo, aprovando o empréstimo. Em caso negativo, o Consultor pode tanto solicitar a complementação dos dados pelo Cliente quanto negar a proposta, dependendo da situação.

A Figura 36 ilustra o fluxo do processo descrito a partir da análise realizada por BLEVI, DELPORTE E ROBBRECHT (2017).

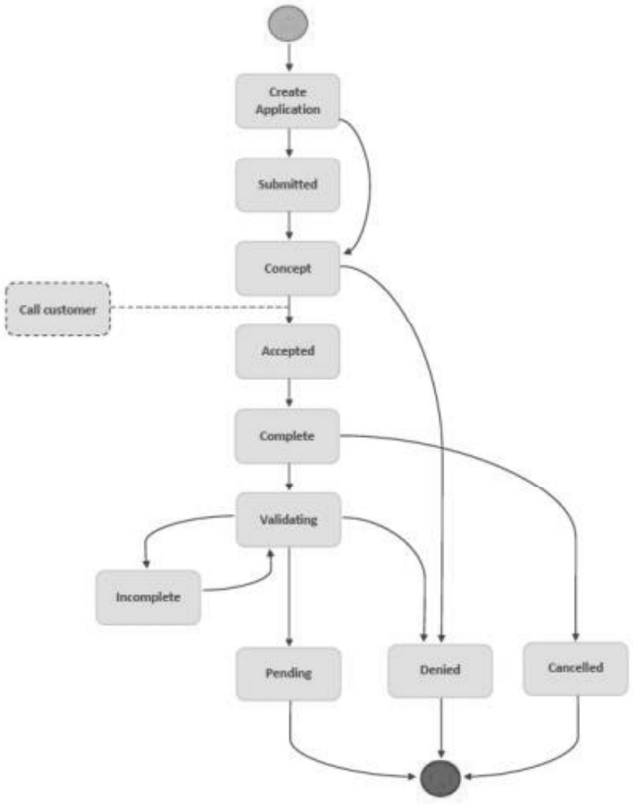

Figura 36 – Modelo do processo de liberação de crédito

A seguir, são descritas as regras de negócio identificadas no domínio e utilizadas ao longo dos exemplos deste trabalho:

Regra-LC1: Ao menos 3 tentativas de ligação devem ser feitas para clientes antes do cancelamento da proposta de empréstimo

Regra-LC2: É obrigatório que o cliente escolha somente uma proposta de empréstimo dentre as oferecidas pelo banco.

5.1.3 Aluguel de veículos

A SBVR (OMG, 2016) descreve um cenário de estudo referente a uma empresa fictícia chamada EU-Rent, cujo negócio envolve o aluguel de veículos de diversas categorias, desde "econômica" até "*premium*".

A empresa aluga veículos para Clientes individuais ou membros de alguma empresa, que firmam um contrato entre as partes sendo o cliente responsável pelo pagamento do aluguel e quaisquer outros custos associados (exceto os cobertos por seguro).

Os Clientes retiram e devolvem o carro alugado em uma das lojas da EU-Rent, gerando taxas extras caso os locais de retirada e devolução sejam diferentes. Um aluguem sempre terá uma data de retirada e devolução pré-estabelecidos, sendo permitido ao Cliente solicitar prorrogação do prazo de devolução sempre que necessário, sendo a extensão concedida mediante aprovação da empresa.

A seguir, são descritas as regras de negócio identificadas no domínio e utilizadas ao longo dos exemplos deste trabalho:

Regra-AV1: É obrigatório que toda solicitação de prorrogação do período de um aluguel em aberto seja analisada por um gerente durante a vigência de uma locação;

Regra-AV2: É obrigatório que um pedido de prorrogação do período de locação em aberto seja analisado antes que uma nova solicitação possa ser feita;

Regra-AV3: É obrigatório que uma confirmação por e-mail com o resultado da análise seja enviada ao cliente imediatamente após a conclusão da verificação pelo gerente.

Regra-AV4: É obrigatório que seja enviado um e-mail ou realizado um contato telefônico com o cliente para tentar aprovar a proposta de empréstimo antes de cancelá-la

Regra-AV5: O motorista que aluga o veículo deve ter, no mínimo, 25 anos

Regra-AV6: É necessário que cada aluguel de veículo tenha exatamente um carro associado

Regra-AV7: É obrigatório que uma locação incorra em uma taxa se a filial de devolução da locação for diferente daquela em que o automóvel foi alugado

5.2 Avaliação

A medição de ontologias é necessária para avaliar tanto a construção quanto a aplicação dos conceitos descritos por ela, sendo uma pré-condição para garantir sua qualidade e controle no processo de aprimoramento (VRANDEčIć e SURE, 2007). Como ontologias

são consideradas modelos de referência, é preciso considerar sua avaliação sobre as perspectivas de qualidade e exatidão, segundo RAAD e CRUZ (2015) e são avaliados segundo os seguintes critérios (VRANDEčIć, 2010):

- **Acurácia**, que afirma se as definições, descrições das classes e propriedades do domínio presentes na ontologia estão corretas;
- **Completude**, que mede se um determinado domínio é apropriadamente coberto pela ontologia;
- **Concisão**, que atesta se a ontologia possui elementos irrelevantes para o domínio coberto;
- **Adaptabilidade**, que mede o quanto a ontologia consegue prever variedades distintas de instâncias;
- **Clareza**, que define o quão efetiva é a comunicação dos elementos da ontologia com seu significado;
- **Eficiência computacional**, que mede a flexibilidade de utilização da ontologia por ferramentas de inferência (*reasoners*);
- **Consistência**, que atesta que a ontologia não inclui ou permite contradições.

Essas características, descritas por VRANDEčIć (2010), são utilizadas para a avaliação de construção de uma ontologia completa, o que não é o foco deste trabalho. Assumimos como premissa que a ontologia que esse trabalho se propõe a estender, a KiPO, possui os critérios de qualidade e exatidão necessários.

As características descritas podem ser avaliadas seguindo diversas métricas, que diferem entre si pelo número de critérios que conseguem avaliar e a técnica de avaliação adotada. RAAD e CRUZ (2015) descrevem quatro das possíveis abordagens existentes para avaliação de uma ontologia: *Gold Standard-based*, *Corpus-based*, *Task-based* e *Criteria based*.

A abordagem *Gold Standard-based*, também conhecida como alinhamento de ontologias, presume a comparação da nova ontologia com uma ontologia de referência previamente existente e já avaliada, que é classificada como *gold standard* e representa uma visão idealizada do resultado esperado. Essa abordagem é comumente utilizada para ontologias de domínios semelhantes, mas que possuem diferenças estruturais ou conceituais devido a particularidades do domínio. Por exemplo, uma ontologia de verbos

em português poderia utilizar a WordNet como a *gold standard* para avaliação do resultado (RAAD e CRUZ, 2015).

Essa abordagem é eficiente para avaliar a **Acurácia** de uma ontologia, pois garante que as definições, descrições das classes e propriedades estejam corretas, e a **Completude** e **Concisão**, pois se a ontologia utiliza como base de avaliação uma outra que é a idealização do resultado, pode-se garantir que ela contém todos os elementos que representam o domínio sem a utilização de elementos irrelevantes.

A abordagem de *Corpus* é utilizada para avaliar o quão bem uma ontologia consegue descrever um domínio. O conceito por trás dessa abordagem é a utilização de um conjunto de dados do domínio (corpus) para comparação com a ontologia descrita e avaliar o quanto ela consegue cobrir do domínio (RAAD e CRUZ, 2015). Essa comparação pode ser realizada através de extração automática de textos e comparando os termos encontrados com os conceitos descritos na ontologia. PATEL et al. (2003) utiliza a extração de texto para avaliar sua ontologia, relacionando os conceitos e relações encontradas com os elementos descritos pela ontologia.

Essa abordagem permite avaliar, assim como na abordagem *gold standard*, os critérios de **Acurácia**, **Completude** e **Concisão** pois a comparação dos itens encontrados nos dados com os elementos presentes na ontologia é facilitada.

A abordagem de *Corpus* será extrapolada para ser utilizada por este trabalho para medir a qualidade e exatidão da extensão desenvolvida para a Ontologia de Processos intensivos em Conhecimento em virtude da existência de fontes de dados e cenários de estudo que permitem a extração, ainda que manual, de instâncias suficientes para avaliar a ontologia sob os critérios definidos anteriormente. Além disso, realizando a mesma avaliação sobre a versão atual da Ontologia de Regras de Negócio (BRO) é possível derivar que as extensões propostas nesse trabalham atende à métrica de **Adaptabilidade**, ao demonstrar que alguns aspectos do domínio não conseguem ser representados com a ontologia atual. A natureza das fontes de dados e cenários de estudo serão descritas na seção seguinte.

A abordagem *Tasks* tenta medir o quanto uma ontologia consegue aprimorar os resultados de uma determinada tarefa. Essa abordagem considera que a ontologia é desenvolvida para atender uma tarefa específica e é avaliada sob a perspectiva do

desempenho dessa atividade antes e depois da utilização da ontologia. WELTY (2003) utiliza essa abordagem para avaliar sua ontologia para melhoria de performance de motores de busca na web, avaliando se ao executar diversas consultas distintas os resultados obtidos são mais ou menos relevantes com a ontologia proposta.

Por fim, a avaliação baseada Critérios (*Criteria*) mede o quanto uma ontologia está aderente a um determinado critério desejado. É o método de avaliação mais completo, mas que demanda uma maior complexidade de definição dos métodos de medição. Por exemplo, pode-se desejar avaliar apenas a estrutura da uma ontologia, onde se busca apenas observar critérios de qualidade como Acurácia e Completude. Essa abordagem também permite avaliar critérios que nenhuma das outras abordagens são capazes de fazer, como a busca por contradições na ontologia ou a verificação de sua eficiência computacional. Essa abordagem não será utilizada neste trabalho por não haver, na proposta, a preocupação com o desempenho computacional dos elementos descritos na extensão da ontologia.

A abordagem de avaliação da DecKiPO foi a abordagem por Corpus, que presume a utilização de um conjunto de dados do domínio para verificar se os elementos descritos na ontologia atendem às métricas de avaliação e oferecem uma descrição mais concisa das características propostas por VRANDEčIć (2010). Por esse motivo, a avaliação ontológica e a aplicabilidade da proposta se misturam para, ao mesmo tempo que demonstram a utilização e aplicabilidade dos novos conceitos introduzidos, explicitam as métricas de avaliação que atestam a qualidade e exatidão da extensão da ontologia.

Uma avaliação baseada em dados foi utilizada para atestar a qualidade e exatidão da ontologia. O critério de **Acurácia** será avaliado a partir da verificação dos axiomas dos elementos propostos em comparação com a LTL Declare, taxonomia de origem da DecKiPO, a fundamentação na UFO e as descrições das classes adicionadas à ontologia; O critério de **Completude** será avaliado a partir da representação de diversos modelos de processos intensivos em conhecimento dos três domínios de estudo deste trabalho em comparação com o atual modelo da KiPO; O critério de **Concisão** será avaliado identificando os relacionamentos encontrados nas instâncias dos processos de negócio modelados, verificando se todos os conceitos da DecKiPO podem ser identificados nos cenários propostos; O critério de **Adaptabilidade** será avaliada através da modelagem

das regras de negócio identificadas nos três cenários de estudo apresentados; Por fim, o critério de **Clareza** será avaliado em termos da definição formal das entidades e documentação dos termos (VRANDEčIć, 2010).

A métrica de **Eficiência computacional** não será objeto de avaliação deste trabalho pois não é escopo do estudo a performance computacional da KiPO e suas perspectivas.

5.3 Aplicabilidade da proposta

Ao longo dessa seção, são apresentados diversos modelos de processos intensivos em conhecimento extraídos de três domínios distintos e cenários de estudo diversificados. Além dos modelos, são apresentadas instâncias diversas que exemplificam casos reais analisados. Cada um dos elementos propostos na DecKiPO é avaliado individualmente sob as métricas descritas na seção anterior e, ao final, uma avaliação geral é fornecida para a DecKiPO.

Os novos conceitos propostos nesse trabalho também serão avaliadas com a utilização de uma ferramenta para conversão automática de modelos LTL Declare nos conceitos propostos neste trabalho.

Para efeito de referência, as regras de negócio utilizadas nas seções seguintes para avaliar a proposta deste trabalho estarão numeradas sequencialmente, utilizando um prefixo que identifica de qual dos três cenários de teste apresentados ela faz parte, a saber:

- Regra-HD – Cenário de estudo de atendimento a *helpdesk*, apresentado na seção 5.1.1;
- Regra-LC – Cenário de estudo de liberação de crédito, apresentado na seção 5.1.2;
- Regra-AV – Cenário de estudo de aluguel de veículos, apresentado na seção 5.1.3.

5.3.1 Avaliação das *Foundational Existence Rules*

As *Foundational Existence Rules* da DecKiPO definem um conjunto de regras que definem necessidade de existência, ou não, de instâncias de uma atividade durante a execução de uma instância de processo, especializando-se nas regras de negócio *Existence*, *Exactly* e *Absence*.

Com base nos cenários descritos na seção 5.1.1, considere as seguintes regras para exemplificar a modelagem das regras de restrição de existência desta proposta:

Regra-HD1: Só pode haver, no máximo, 2 trocas de técnico ao longo do atendimento a um chamado

Regra-LC1: Ao menos 3 tentativas de ligação devem ser feitas para clientes antes do cancelamento da proposta de empréstimo

A regra de negócio **Regra-HD1** restringe um comportamento do domínio, determinando a quantidade máxima de alteração de técnicos responsáveis pelo atendimento do ticket. Atualmente na KiPO, esse tipo de restrição não pode ser representado visto que as regras de negócio descritas pela BRO não conseguem descrever a semântica necessária para restringir o comportamento do domínio. Com a adição das *Foundational Existence Rules* da DecKiPO, é possível representar a limitação imposta pela regra de negócio **Regra-HD1**, através do conceito *Absence* que delimita o número máximo de instâncias de uma determinada atividade ao longo da instância do processo de negócio. A Figura 37 ilustra o modelo do processo considerando a extensão proposta.

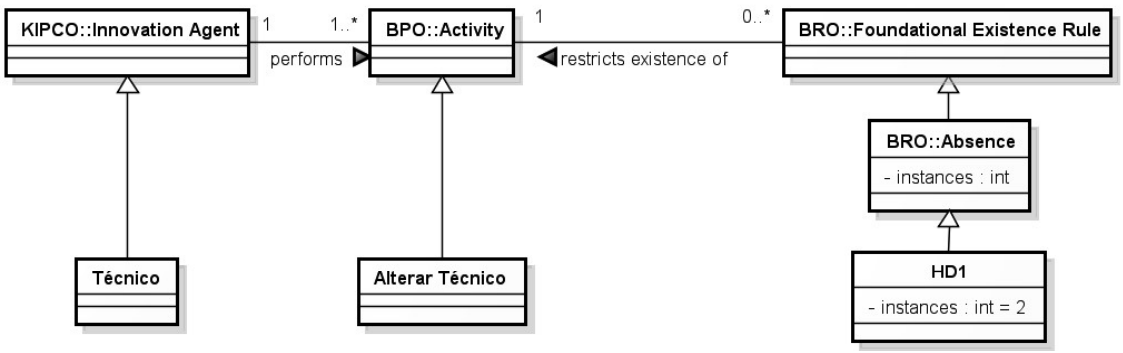

Figura 37 – Modelagem da Regra-HD1 com a DecKiPO

Da mesma forma, a regra de negócio **Regra-HD1** descreve uma restrição onde um número mínimo de instâncias de determinada atividade deve ser executado. Essa restrição é representada pelo elemento *Existence* da DecKiPO que, associado à atividade através da relação *restricts existence of*, obriga a execução da atividade uma quantidade de vezes mínima definida.

Conforme explicado na seção 4.2.1 - Restrições de Existência, o LTL Declare inclui em sua taxonomia dois elementos que não foram considerados na extensão DecKiPO: *init* e *finish*. Esses dois elementos possuem a mesma semântica dos elementos

StartEvent e *EndEvent* da *Business Process Ontology* (BPO) que indicam, respectivamente, o evento que inicia e o evento que encerra o processo de negócio. Portanto, considera-los nesta extensão causaria uma redundância de conceitos na ontologia ferindo, portanto, a métrica de **Concisão**. Outro problema ao incluir os templates *init* e *finish* do LTL Declare seria a contradição que poderia haver em modelos de processos que utilizam a KiPO. Enquanto podem haver diversos *StartEvents* em um modelo de processo com condições de disparo distintos (OMG, 2011), só pode haver uma única atividade com o template *init* em um modelo LTL Declare (PESIC, 2008). Esse tipo de contradição viola o critério de **Consistência** da ontologia, trazendo problemas para modelagem e inferência de modelos de processos de negócio representados pela DecKiPO.

5.3.2 Avaliação das *Foundational Relation Rules*

As *Foundational Relation Rules* da DecKiPO definem um conjunto de regras que definem a relação entre duas ou mais atividades do domínio, englobando aspectos temporais que ordenam e orientam o andamento da execução de atividades do processo de negócio. Conforme especificado na seção 4.2.2, os conceitos *responded existence*, *coexistence*, *precedence*, *response*, *succession*, *alternate precedence*, *alternate response*, *alternate succession*, *chain precedence*, *chain response* e *chain succession* são especializados da *Foundational Relation Rule* para permitir à KiPO representar mais precisamente diversos aspectos temporais das relações entre atividades em um cenário de processo intensivo em conhecimento.

Para avaliar os conceitos adicionados à KiPO, as seguintes regras foram selecionadas:

Regra-AV1: É obrigatório que toda solicitação de prorrogação do período de um aluguel em aberto seja analisada por um gerente durante a vigência de uma locação;

Regra-AV2: É obrigatório que um pedido de prorrogação do período de locação em aberto seja analisado antes que uma nova solicitação possa ser feita;

Regra-AV3: É obrigatório que uma confirmação por e-mail com o resultado da análise seja enviada ao cliente imediatamente após a conclusão da verificação pelo gerente.

A regra de negócio **Regra-AV1** representa uma restrição onde, dada a realização da atividade "Solicitar prorrogação do período de aluguel" por um cliente, a atividade "Analisar solicitação de prorrogação de aluguel" deve ser realizada por um gerente. Essa regra evidencia uma característica importante da extensão proposta por este trabalho pois, apesar de seu texto claramente explicitar a sequência de execução de atividades, ele possui características que passam desapercebidas pela abordagem imperativa tradicional ou pelos conceitos existentes na Ontologia de Regras de Negócio (BRO).

A Figura 38 demonstra a Regra de Negócio **Regra-AV1** modelada utilizando a versão atual da KiPO.

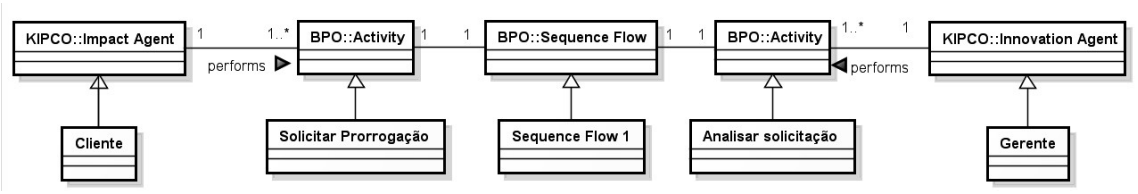

Figura 38 – Modelagem da Regra-AV1 na versão atual da KiPO

A Regra de Negócio **Regra-AV1** pode ser descrita como uma relação *Response* uma vez que a atividade "Analisar solicitação de prorrogação de aluguel" terá que acontecer eventualmente depois que a atividade "Solicitar prorrogação do período de aluguel" for realizada. Utilizando os conceitos fornecidos pela *Business Process Ontology* (BPO), essa restrição seria implementada utilizando o conceito *Sequence Flow*, que relaciona duas atividades e direciona o fluxo de execução. Em uma primeira análise, podemos considerar que o elemento *Sequence Flow* da BPO possui a mesma semântica do elemento *Response* da DecKiPO pois ambos descrevem uma relação de sequenciamento de atividades.

Porém, como a semântica do *Sequence Flow* é oriunda da especificação BPMN (OMG, 2011), a relação descrita na Figura 38 não responde a questões sobre quando a atividade "Analisar solicitação de prorrogação de aluguel" deve ser realizada. Essa afirmação pode ser verificada comparando-a com a regra de negócio **Regra-AV1**, cuja modelagem utilizando a KiPO está representada na Figura 39.

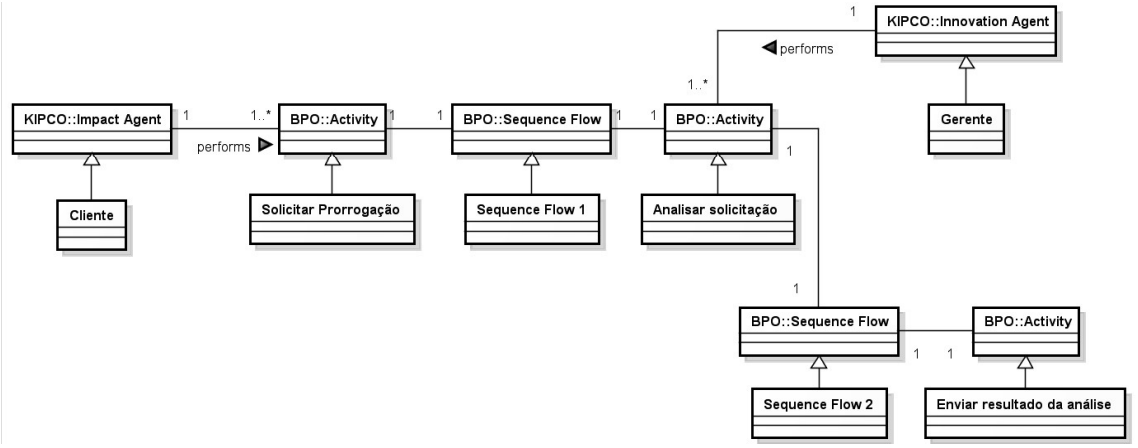

Figura 39 – Modelagem da Regra-AV1e Regra-AV3 na versão atual da KiPO

Não é possível definir, utilizando a BPO, o aspecto temporal envolvendo a relação representada pelas especializações *Sequence Flow 1* e *Sequence Flow 2*. Mesmo considerando a possibilidade de utilização do atributo *isImmediate* do elemento *Sequence Flow* que determina se outras atividades podem ocorrer entre o período de execução das atividades conectadas, em termos semânticos não significa dizer que a atividade A ocorre imediatamente depois da atividade B.

Utilizando os conceitos estendidos através da DecKiPO, as relações temporais descritas anteriormente são explicitadas, sendo possível descrever claramente as restrições impostas pelas regras de negócio. Conforme descrito na seção 4.2.2 - Restrições de Relação, existem dois tipos de regras de negócio capazes de explicar com clareza as duas relações citadas anteriormente: *Response* e *Chain Response*. Essas duas regras de negócio comportamentais adicionam ao metamodelo a semântica de relação entre duas atividades e especificam as restrições temporais entre elas, determinando claramente a precedência entre atividades e a dependência entre elas. A Figura 40 apresenta o modelo do processo de negócio sob a ótica dos elementos estendidos na DecKiPO.

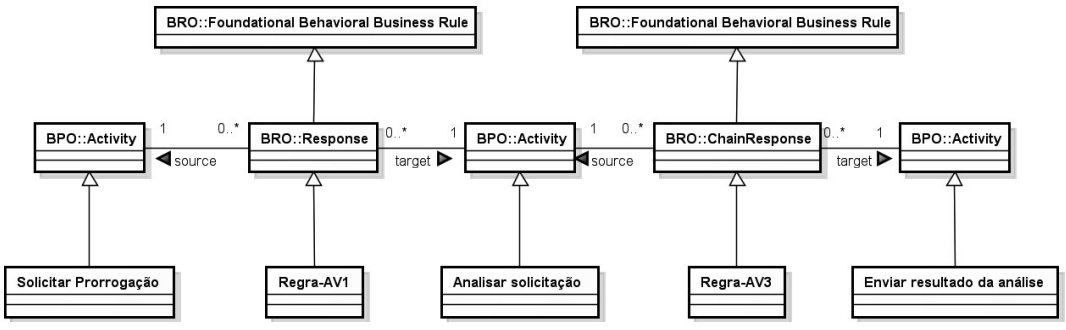

Figura 40 – Modelo com a extensão DecKiPO da Regra-AV1e Regra-AV3

Analisando a regra de negócio **Regra-AV2**, pode-se perceber que ela condiciona a existência de uma nova instância da atividade "Solicitar prorrogação do período de aluguel" à conclusão da atividade "Analisar solicitação de prorrogação de aluguel", de modo que não pode haver dois pedidos de prorrogação do período de aluguel em aberto ao mesmo tempo.

Essa restrição pode ser modelada na KiPO conforme demonstrado na Figura 41 onde a regra de negócio é levada em consideração na avaliação do gerente a partir da questão levantada pelo cliente.

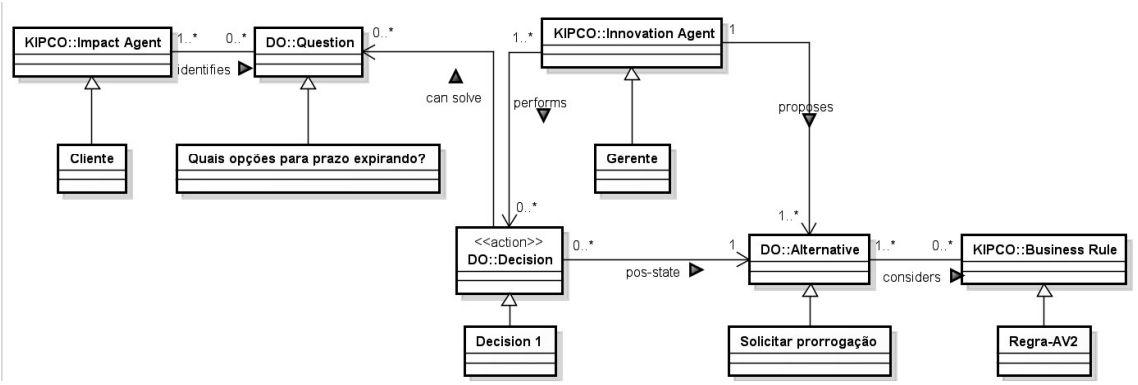

Figura 41 – Modelo da Regra-AV2 na versão atual da KiPO

Esse modelo, apesar de representar a restrição imposta pela regra de negócio, possui algumas características que são melhoradas com a extensão DecKiPO. Primeiramente, pela descrição da regra, é possível concluir que ela se adequa à restrição *Alternate Response*, cuja descrição implica que a atividade A não pode ser executada novamente enquanto a atividade B não for realizada.

Na versão atual da KiPO, os tipos de regra de negócio existentes não fazem referência a quais eventos estão restringindo e, em um cenário onde se pretende deixar clara a relação entre atividades de um processo de negócio, a necessidade de se testar todas as regras de negócio para analisar as restrições pertinentes em cada instante da instância torna o modelo pouco amigável para um especialista, dificultado que ele visualize com rapidez os passos a serem seguidos.

Com a simples adição do conceito *Alternate Response* como uma especialização de regras de negócio, tanto a modelagem quanto a avaliação computacional das restrições são facilitadas, visto que o modelo passa a representar claramente, conforme Figura 42, as associações impostas às atividades do domínio.

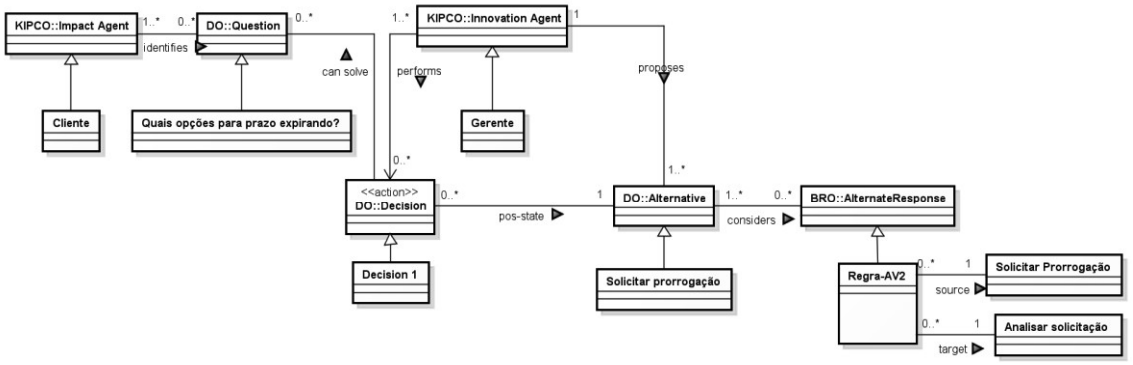

Figura 42 – Modelo da Regra-AV2 com a extensão DecKiPO

Além disso, o conceito *Alternate Response* permite que duas regras de negócio, **Regra-AV1** e **Regra-AV2**, sejam representadas com um único elemento pois além de definir a semântica da atividade alternada, também formaliza a sequência esperada de execução da atividade, melhorando a clareza do modelo, conforme.

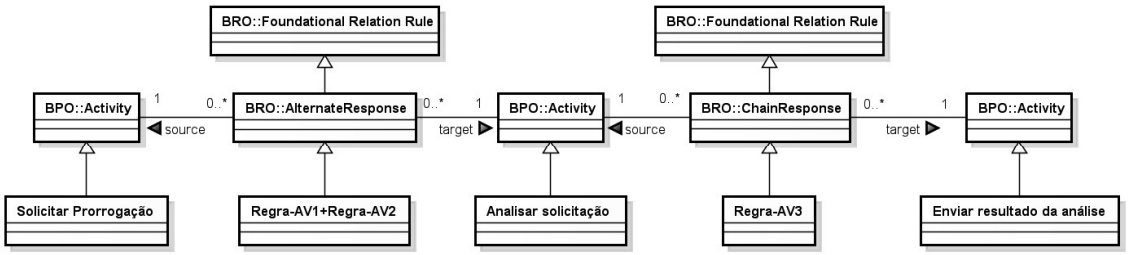

Figura 43 – Modelo com a Regra-AV1, Regra-AV2 e Regra-AV3 utilizando o metamodelo estendido com a DecKiPO

Apesar de não terem sido utilizadas ferramentas para realizar a medição do desempenho de avaliação do metamodelo, é seguro afirmar que a avaliação da ontologia através de *reasoners* utilizando os elementos estendidos da DecKiPO é computacionalmente mais eficiente do que a abordagem anterior, principalmente no que tange às verificações de restrições de execução de atividades.

Um dos principais benefícios da utilização dos elementos da *Foundational Relation Rules* é a possibilidade de identificação imediata tanto por especialistas quanto por *reasoners* das atividades disponíveis para execução em um determinado instante da execução de uma instância de KiP. As alternativas consideradas para tomada de decisão, representadas pelos conceitos homônimos na KiPO, que estejam relacionadas diretamente à possibilidade de execução de uma atividade, são diretamente identificadas através das relações *source* e *target* que todos os elementos da *Foundational Relation Rule* herdam.

Considerando a métricas de avaliação de ontologias propostas por VRANDEčIć (2007), a adição das *Foundational Relation Rules* propostas na DecKiPO contribuem para que a KiPO cubra aspectos diversos de um domínio, evidenciando o critério de qualidade da **Completude**, além de permitir que relações antes não previstas pela Ontologia de Regras de Negócio (BRO) sejam agora representadas, contribuindo para que a **Adaptabilidade** da ontologia ao domínio de processos intensivos em conhecimento seja ainda maior.

5.3.3 Avaliação das *Foundational Choice Rules*

As *Foundational Choice Rules* da DecKiPO limitam as escolhas dos eventos que podem ser realizados durante a execução de um KiP. Esse tipo de regra está diretamente relacionado às alternativas disponíveis em cada estado do processo e delimitam não só a quantidade de escolhas possíveis, mas também como elas podem ser selecionadas.

Conforme especificado na seção 4.2.3, os conceitos *Choice* e *ExclusiveChoice* são especializados da *Foundational Choice Rule* para permitir à KiPO representar os critérios para escolha de execução de eventos em cenários de processos intensivos em conhecimento.

Com base nos cenários descritos na seção 5.1.1, considere as seguintes regras para exemplificar a modelagem das regras de restrição de escolha desta proposta:

Regra-LC2: É obrigatório que o cliente escolha somente uma proposta de empréstimo dentre as oferecidas pelo banco.

Regra-AV4: É obrigatório que seja enviado um e-mail ou realizado um contato telefônico com o cliente para tentar aprovar a proposta de empréstimo antes de cancelá-la

Segundo a KiPO, uma atividade de tomada de decisão pode envolver a escolha de uma dentre diversas alternativas possíveis em um estado do processo, sendo estas alternativas apresentadas considerando regras de negócio do domínio.

Durante uma *Decision* o Agente considera estados possíveis (*Alternatives*) que seriam trazidos para a realidade com a execução de uma atividade que tenha esta situação como Post-State. Depois de definida a *Chosen Alternative*, ou seja, o estado desejado para a realidade, o processo continua com a execução de uma atividade que resulte neste

estado. Neste momento, atividades são apresentadas ao agente para que este atinja o objetivo que o levará a mudar a realidade para o estado escolhido.

Considerando a **Regra-LC2**, pode-se imaginar que a situação desejada seja a mesma para ambos os casos: “Cliente contatado”. Nesse caso não haveria nenhuma distinção para embasar a Decisão (*Decision*) do Agente (*Agent*). Então, pode-se pensar em situações (*Alternatives*) distintas como “Cliente contatado por e-mail”/”Cliente contatado por telefone”), e neste caso a Decisão da Alternativa escolhida (*Chosen Alternative*) poderia levar em conta os demais elementos relacionados a cada alternativa (Vantagens/Desvantagens/Riscos):

- a Alternativa “Cliente contatado por telefone” teria como Vantagens por exemplo “Reduzir tempo de resposta do cliente”, “Criar relação interpessoal com Cliente”
- a Alternativa “Cliente contatado por e-mail” teria como vantagem “Manter registro do contato com cliente”, e como Risco “e-Mail não recebido em função de filtro de Spam”

Uma vez definida a alternativa escolhida, seria necessário apresentar as atividades do domínio que possivelmente levariam ao estado desejado pelo Agente. A Figura 44 ilustra a representação da restrição de escolha utilizando o metamodelo da DecKiPO.

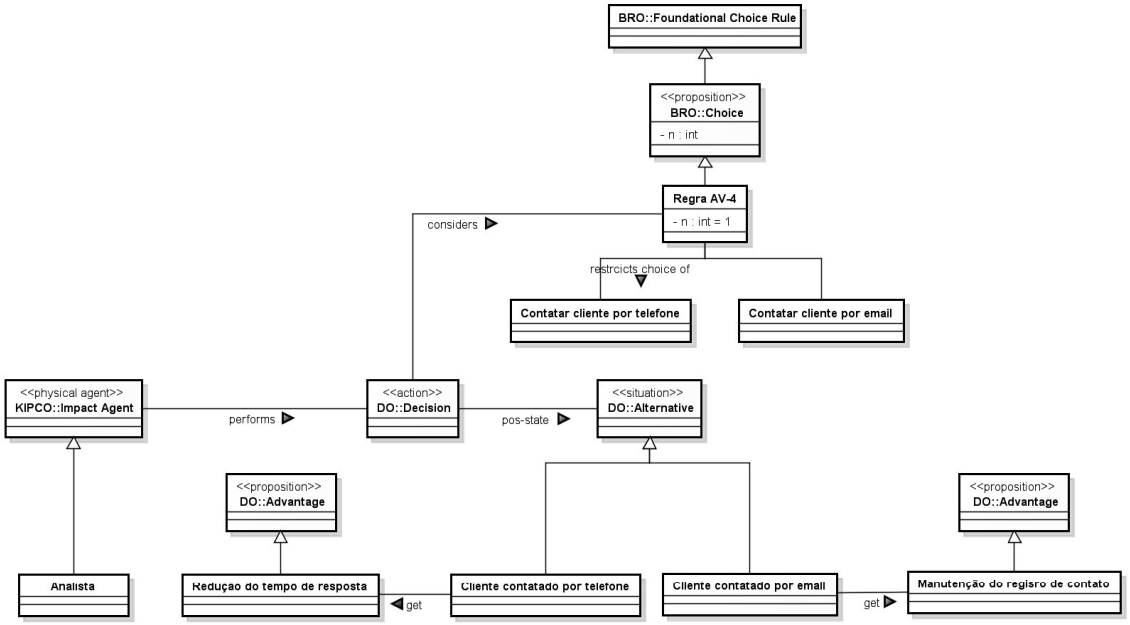

Figura 44 – Modelo com a Regra-AV4 utilizando o metamodelo estendido com a DecKiPO

5.3.4 Avaliação dos *Enforcement Levels*

Uma das principais características processos intensivos em conhecimento é sua imprevisibilidade e baixa repetitividade (HAGEN, RATZ e POVALEK, 2005) e, por essa razão, a definição dos limites ou restrições de execução do processo torna-se uma melhor abordagem para descrever os domínios que apresentam essa característica. Por esse motivo, esse trabalho apresentou até este capítulo uma proposta de extensão onde os conceitos da modelagem declarativa de processos são adotados na Ontologia de Processos Intensivos em Conhecimento (KiPO) para permitir que as atividades inerentes a todo processo de negócio sejam orquestradas em função das restrições do domínio ao invés do sequenciamento imperativo tradicional.

Entretanto, em virtude na natureza imprevisível dos processos intensivos em conhecimento, as regras de negócio que restringem o comportamento do domínio podem não ser consideradas nas atividades de tomada de decisão, gerando um estado de violação de regras de negócio que podem trazer consequências tanto para o agente quanto para o objetivo do processo.

A severidade dessas consequências está diretamente relacionada ao nível de cumprimento esperado para a regra de negócio, que impõe ao agente uma necessidade de atendimento à restrição imposta e definindo a consequência sofrida caso seja violada.

Conforme especificado na seção 4.3 - Níveis de cumprimento com SBVR, foram adicionados diversos conceitos para representar os níveis de cumprimento: *Enforcement Level*, *Conceived Enforced Level*, *Perceived Enforcement Level* e *Severity Level*.

Para avaliar os conceitos adicionados à KiPO, as seguintes regras foram selecionadas:

Regra-AV7: É obrigatório que uma locação incorra em uma taxa se a filial de devolução da locação for diferente daquela em que o automóvel foi alugado

Regra-LC1: Ao menos 3 tentativas de ligação devem ser feitas para clientes antes do cancelamento da proposta de empréstimo

A regra de negócio **Regra-LC1**, como visto na seção 5.3.1, é representada na DecKiPO como uma regra de negócio do tipo *Existence*, uma vez que representa uma restrição de obrigatoriedade de execução de, no mínimo, três instâncias da atividade "Realizar ligação telefônica para cliente".

Em condições normais de execução de uma instância do processo, essa regra de negócio seria de conhecimento dos agentes do processo e considerada por todos sempre que uma situação de cancelamento de proposta de empréstimo fosse identificada, sendo representada na KiPO através do modelo ilustrado na Figura 45.

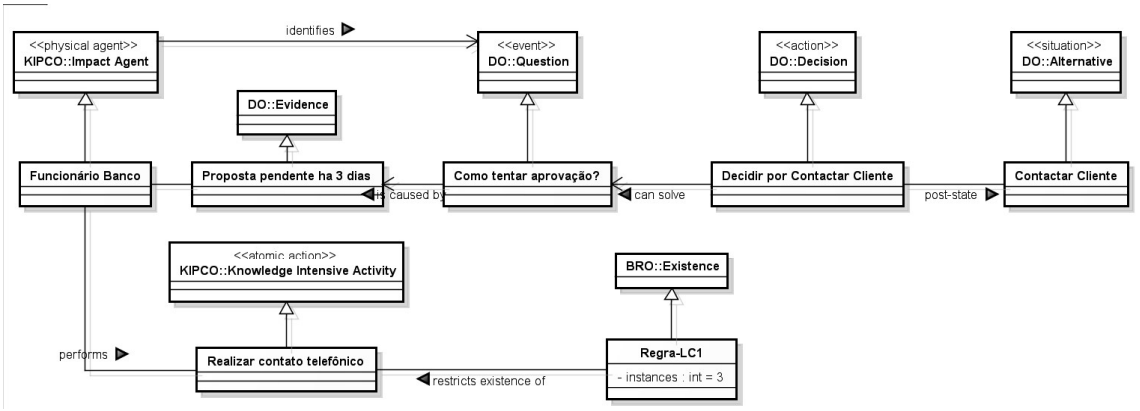

Figura 45 – Modelo em DecKiPO da tomada de decisão e restrições envolvendo a Regra-LC1

A simples referência à regra de negócio, porém, não fornece ao modelo a semântica completa do domínio pois, dependendo das diretrizes fornecidas pelo negócio, a restrição imposta é mais ou menos rígida quanto à necessidade de cumprimento. No modelo da

Figura 45, não é possível determinar se a restrição imposta pela **Regra-LC1** é estrita ou apenas uma recomendação do negócio.

Com a adição dos níveis de cumprimento, é possível representar esses conceitos, aumentando a **Clareza** dos modelos de processo e garantindo que a ontologia **abranja todos os conceitos** do domínio e se **adapte** a diversos modelos de processos de negócio.

A semântica do nível de cumprimento concebido para a **Regra-LC1** é fornecida através do elemento *Conceived Enforcement Level*. O cenário de estudo de liberação de crédito, descrito na seção 0, não fornece maiores detalhes sobre o nível de cumprimento esperado para a regra de negócio então, na Figura 46, ele é definido como sendo *Strictly Enforced*.

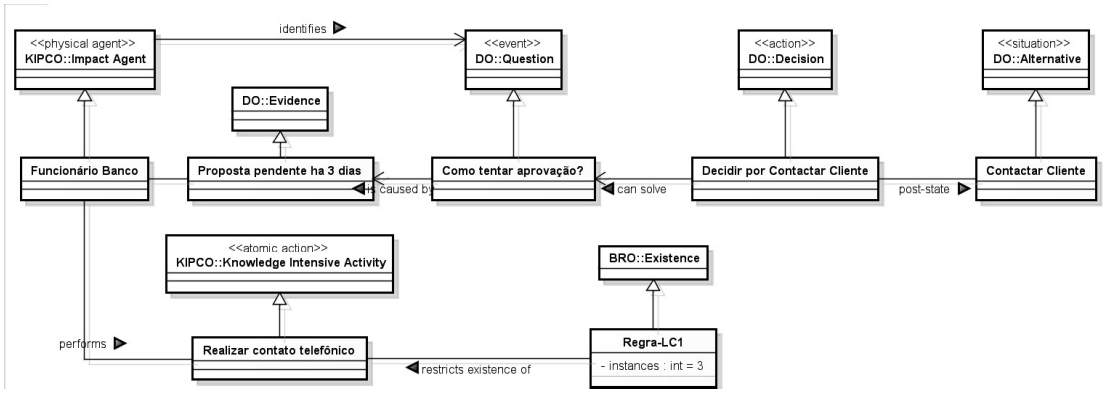

Figura 46 – Modelo em DecKiPO com a adição do nível de cumprimento concebido para a Regra-LC1

Outro cenário possível do domínio restrito pela **Regra-LC1** é aquele onde, após uma ou duas tentativas, o processo é aprovado, não sendo necessário realizar uma terceira tentativa. Novamente, o cenário de estudo avaliado neste trabalho não possui evidências desse desvio mas a DecKiPO permitiria sua representação através da alteração do grau de severidade do nível de cumprimento concebido, atribuindo através do *Severity Level PreAuthorized Override* a semântica de possibilidade de descumprimento pré-autorizado da regra de negócio caso um evento de aprovação da proposta seja realizado antes de seu término, cujo modelo está representado na Figura 47.

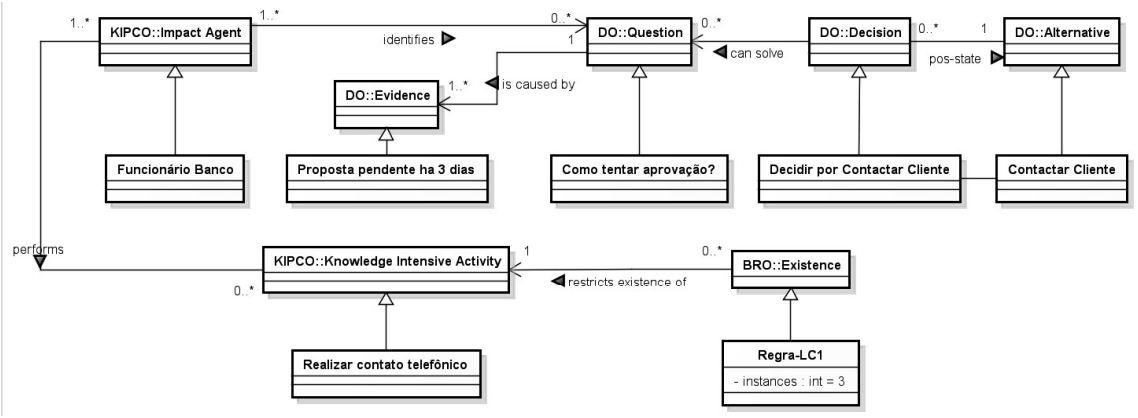

Figura 47 – Modelo em DecKiPO com a adição do nível de cumprimento pré-autorizado concebido para a Regra-LC1

O cenário de uma regra de negócio cujo nível de cumprimento seja *PreAuthorized Override* obriga, através do **Axioma A36**, que especialista do domínio ou agente do processo defina uma atividade anterior à restrita pela **Regra-LC1** para que não haja uma violação à regra de negócio pois, uma vez que a regra de negócio **Regra-LC1** é instanciada, ela necessariamente deve atender à sua especificação.

Os dois exemplos anteriores presumem que o agente conhece e executa a regra de negócio, mesmo que parcialmente. Porém, podem existir cenários onde o agente desconheça a existência da regra de negócio ou opte ativamente por descumpri-la, causando uma violação no domínio. Por exemplo, o agente poderia realizar o cancelamento da proposta mesmo sem realizar as ligações necessárias.

Na KiPO esse tipo de violações à regras de negócio, seja por ação direta do agente, seja por consequência do estado do domínio oriundo de mudanças ocasionadas por outras regras de negócio, não são facilmente identificadas pois na ontologia não é possível relacionar esses eventos não esperados a sua origem. O conceito que mais se aproxima dessa representação é o *Contingency*, que descreve a ocorrência de um evento não previsto no domínio originalmente.

Ainda assim, o elemento *Contingency* representa apenas um evento criado durante a execução de um processo de negócio em decorrência de troca de informações ou socialização. Para que o metamodelo possa representar corretamente um evento de violação à regras de negócio, é preciso primeiramente identificar de que maneira esta regra de negócio foi percebida pelo agente. Na Figura 48, é apresentada a modelagem da **Regra-AV7** sob a perspectiva do modelo do processo e seu nível de cumprimento.

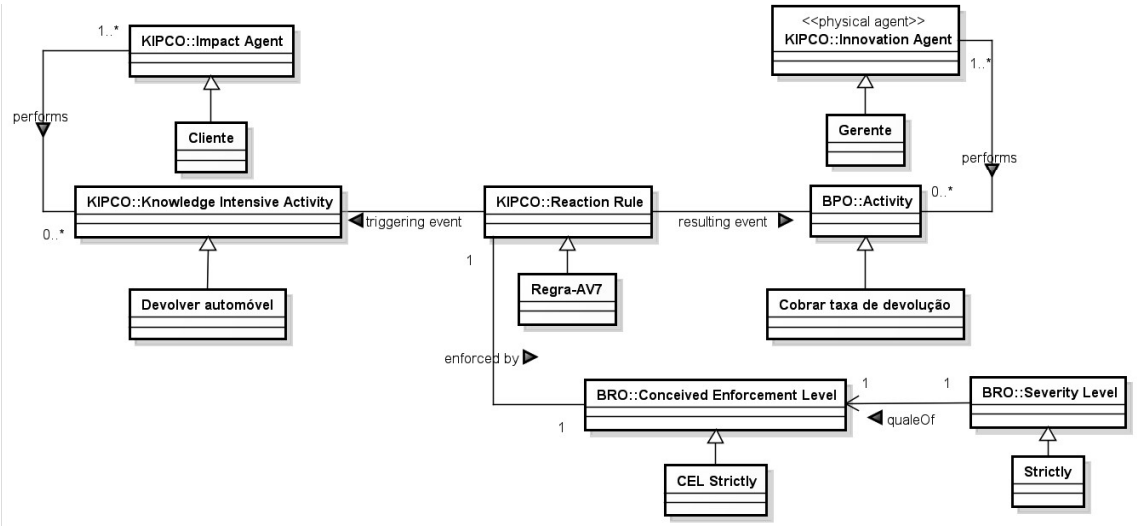

Figura 48 – Modelo em DecKiPO da Regra-AV7 com o nível de cumprimento derivado da descrição da regra de negócio

Conforme descrito na seção 4.3.1, o elemento *Perceived Enforcement Level* da DecKiPO representa a relação entre o agente, a decisão tomada e a regra de negócio relacionada considerada, ou não, na ação. Este elemento, ao contrário dos demais apresentados nessa proposta, só possui uma representação em nível de instância do processo, visto que somente durante a execução é que se pode determinar como um agente percebe uma regra de negócio. A Figura 49 apresenta uma representação resultante da instância executada a partir do modelo da Figura 48.

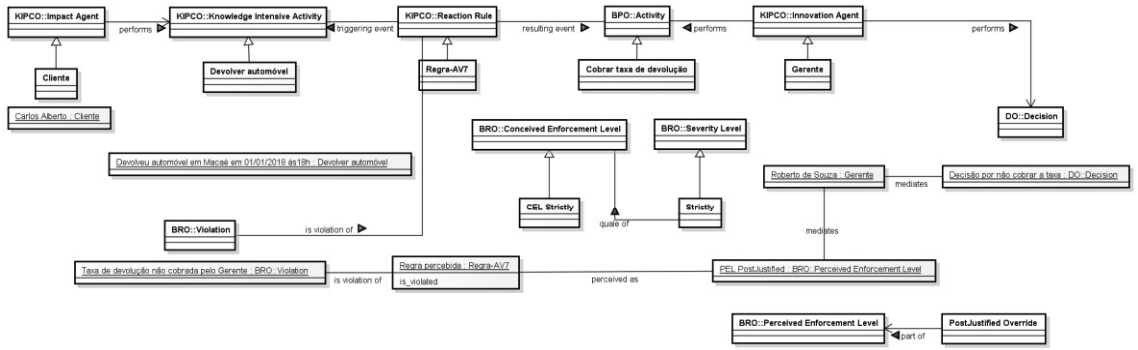

Figura 49 – Modelo em DecKiPO da instância de execução do processo que viola a Regra-AV7

A possibilidade de representar a violação de uma regra de negócio através do nível de cumprimento percebido pelo agente garante ao metamodelo da KiPO uma maior adaptabilidade a cenários de processos intensivos em conhecimento além de permitir maior clareza ao especialista do domínio quanto às restrições e desvios no processo.

A verificação pela violação ou não da regra de negócio pode-se avaliada através do **Axioma A36**, que busca, na instância do processo, uma instância de cada regra de negócio.

5.3.5 Transformação de modelos LTL Declare em Modelos DecKiPO

A utilização da KiPO e as extensões propostas pela DecKiPO para modelagem de processos intensivos em conhecimento foram apresentadas ao longo dessa proposta como uma alternativa de elevada expressividade semântica e capacidade de representação de conceitos em diversas perspectivas que permeiam um KiP.

Entretanto, atualmente ainda não existem soluções para mineração de processos intensivos em conhecimento capazes de gerar, a partir de um log de eventos, modelos de processos KiPO. Entretanto, dada a natureza não estruturada desses tipos de processo e o paradigma de modelagem declarativa, conforme amplamente abordado neste trabalho, é possível extrair regras de negócio e atividades a partir de log de eventos, utilizando ferramentas já adotadas na literatura.

Como esse é trabalho fundamenta a extensão de elementos a partir dos *templates* LTL Declare, foi possível criar uma ferramenta capaz de converter os modelos gerados por mineradores de processo desse metamodelo para o metamodelo da KiPO.

O Declare Miner é uma extensão para ProM que permite a extração de modelos LTL Declare a partir de um log de eventos estruturado. Para este trabalho, foi utilizado o log de eventos do primeiro cenário de estudo, descrito na seção 5.1.1 do trabalho, que contém 246.283 eventos registrados distribuídos entre 6.337 instâncias únicas de processos.

O resultado da mineração das instâncias dos processos após a execução da extensão Declare Miner está ilustrado na Figura 50.

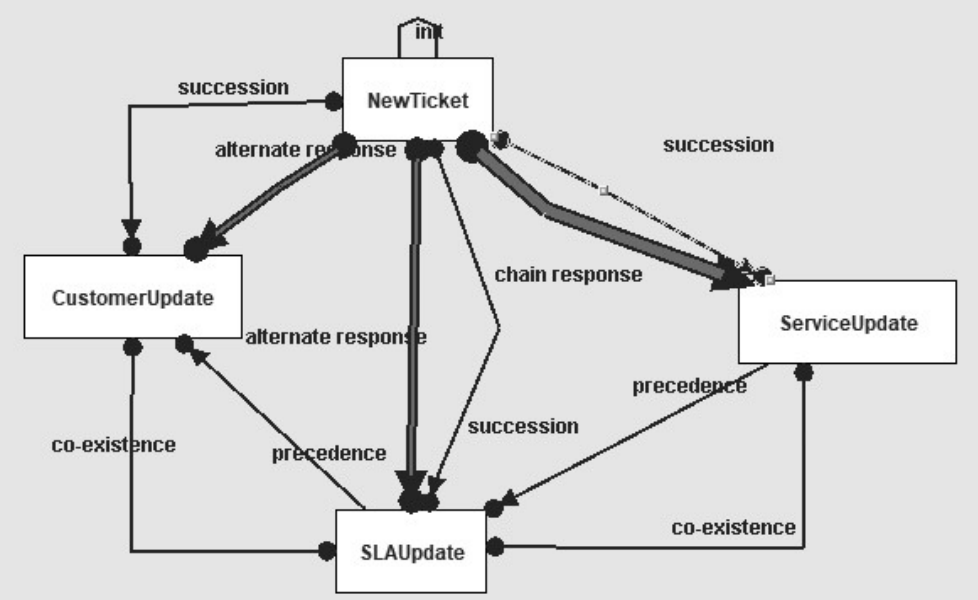

Figura 50 – Modelo LTL Declare gerado após a mineração das instâncias dos processos

A ferramenta gera um arquivo em formato XML que é utilizada como arquivo de entrada para a ferramenta de transformação dos modelos. A ferramenta avalia todos os templates LTL Declare encontrados e as relaciona com as atividades restritas por elas e qual Classe da DecKiPO implementa o *template*, como representa a Tabela 5.

Tabela 5 – Representação da associação dos elementos do LTL Declare com DecKiPO

Template LTL Declare	Classe DecKiPO	Elemento
Activity	KiPCO::KiA	NewTicket
Activity	KiPCO::KiA	ServiceUpdate
Activity	KiPCO::KiA	SLAUpdate
Activity	KiPCO::KiA	CustomerUpdate
init(A)	BPO::StartEvent	a:NewTicket
succession(A,B)	BRO::Succession	a:NewTicket – b:ServiceUpdate
chain response(A,B)	BRO::ChainResponse	a:NewTicket – b:ServiceUpdate
coexistence(A,B)	BRO::CoExistence	a:SLAUpdate – b:ServiceUpdate
succession(A,B)	BRO::Succession	a:NewTicket – b:CustomerUpdate
alternate response(A,B)	BRO::AlternateResponse	a:NewTicket – b: CustomerUpdate
coexistence(A,B)	BRO::CoExistence	a:CustomerUpdate– b:SLAUpdate

Template LTL Declare	Classe DecKiPO	Elemento
precedence(A,B)	BRO::Precedence	a:ServiceUpdate – b:SLAUpdate
succession(A,B)	BRO::Succession	a:NewTicket – b:SLAUpdate
alternate response(A,B)	BRO::AlternateResponse	a:NewTicket – b:SLAUpdate
precedence(A,B)	BRO::Precedence	a:SLAUpdate– b:CustomerUpdate

Esses elementos são incorporados a uma versão OWL da Ontologia de Processos intensivos em Conhecimento (KiPO), contendo toda a estrutura da ontologia. Cada elemento identificado é adicionado à ontologia como subclasses da classe representada na coluna **Classe DecKiPO** com um identificador único composto pelo nome do *template* e um número sequencial.

Com a utilização da ferramenta DecKiPO[1], é possível gerar modelos KiPO a partir de qualquer log de eventos utilizando a extensão Declare Miner do ProM 6.7. A Figura 51 ilustra um extrato da OWL resultante da transformação realizada pela ferramenta.

```
- <SubClassOf>
    <Class IRI="#chainresponse2"/>
    <Class IRI="#BRO::ChainResponse"/>
  </SubClassOf>
- <SubClassOf>
    <Class IRI="#chainresponse2"/>
   + <ObjectExactCardinality cardinality="1">
  </SubClassOf>
- <SubClassOf>
    <Class IRI="#chainresponse2"/>
   - <ObjectExactCardinality cardinality="1">
        <ObjectProperty IRI="#Target"/>
        <Class IRI="#ServiceUpdate"/>
    </ObjectExactCardinality>
  </SubClassOf>
```

Figura 51 – Extrato da OWL gerada pela ferramenta da DecKiPO

Essa primeira versão da ferramenta limita-se a realizar a transformação dos *templates* LTL Declare para DecKiPO, não sendo capaz de identificar os níveis de cumprimento das regras de negócio. Porém, devido à restrição de integridade imposta pela DecKiPO através do **Axioma A34** que impõe a necessidade de toda regra de negócio comportamental (*Foundational Behavioral Business Rule*) possuir um nível de cumprimento concebido (*Conceived Enforcement Level*), todos os templates identificados

[1] Disponível em http://tiny.cc/deckipotool

são automaticamente associados, através da relação *enforced by*, a um nível de cumprimento concebido com grau de severidade (*Severity Level*) *strictly enforced*.

Outra limitação da ferramenta é que o nome da regra de negócio é mantido conforme o padrão gerado automaticamente (por exemplo, *precedence7*), não permitindo edição direta. Isso implica que a descrição da regra de negócio não é transcrita para o modelo final e não é intuitiva para um especialista na área. Versões futuras devem permitir a edição do nome da regra de negócio de forma a permitir que possam ser inseridos o texto em linguagem natural, conforme os diversos exemplos descritos nesta proposta.

Um dos exemplos de utilização da OWL gerada é a importação para um banco de dados não-relacional, como por exemplo o Neo4J, que permite armazenar os dados como uma estrutura de grafos, gerando visualizações gráficas que facilitam a identificação de elementos e navegação pelas classes do modelo de processo, conforme ilustrado na Figura 52.

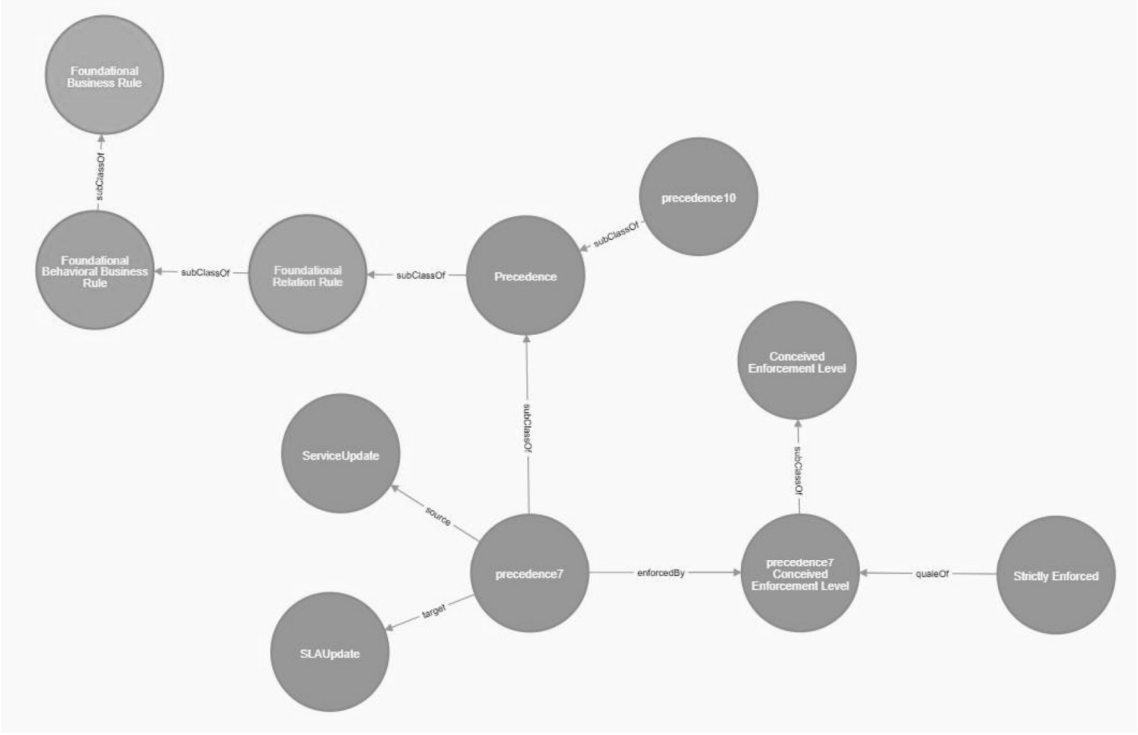

Figura 52 – Exemplo de aplicação da OWL gerada pela ferramenta de transformação da DecKiPO, com um grafo gerado a partir do Neo4J

Futuras versões da ferramenta podem implementar uma análise mais apurada do log de eventos para identificar níveis de cumprimento com graus de severidade mais brandos ou

ainda violações claras a regras de negócio nas diversas instâncias do processo de negócio. A avaliação proposta visou demonstrar que a extensão da Ontologia de Processos intensivos em Conhecimento (KiPO) utilizando os conceitos da LTL Declare e de níveis de cumprimento da SBVR é capaz de prover maior expressividade semântica da ontologia.

Com os conceitos propostos e aplicados aos cenários de estudo apresentados, foi possível demonstrar que a KiPO cobre de forma mais ampla o domínio de processos intensivos em conhecimento a que se destina e que se adapta a cenários de modelagem de processos de negócio que antes não eram contemplados como relações temporais e violações a regras de negócio. A adição dos conceitos de Regras de Fundamentação de Definição (*Definitional Foundational Business Rules*) e Regras de Fundamentação de Comportamento (*Behavioral Foundational Business Rules*) permite que o metamodelo da KiPO expresse com mais clareza as restrições impostas pelas regras de negócio ao domínio modelado bem como permite que especialistas das áreas de negócio representem corretamente os limites do domínio estudado.

6 Conclusão

Ao longo dos Capítulos deste trabalho, foi desenvolvida a extensão à Ontologia de Processos intensivos em Conhecimento (KiPO), chamada de DecKiPO, que adicionou conceitos do paradigma de modelagem declarativa de processos de negócio e os aspectos de níveis de cumprimento e violação de regras de negócio, motivadas pela necessidade de representação mais expressiva das relações entre atividades intensivas em conhecimento.

Com a adoção dos conceitos de modelagem declarativa, em especial com a criação de elementos que representam a semântica dos *templates* de restrição desenvolvidas na LTL Declare, a Ontologia de Processos intensivos em Conhecimento (KiPO) passa a ser capaz de representar de forma ainda mais completa os domínios de processos de negócio para o qual se designa. Com a adição das *Foundational Existence Business Rules*, é possível restringir a existência ou não de instâncias de eventos no contexto de um processo de negócio; as *Foundational Relation Business Rules* são capazes de representar relações temporais entre eventos de quaisquer naturezas em um KiP, em especial às Atividades intensivas em Conhecimento (KiA), foco principal deste trabalho; as *Foundational Choice Business Rules* adicionam à Ontologia a capacidade de restringir escolhas de ação em cenários onde diversas atividades podem ser executadas para solucionar um mesmo problema. Por fim, a DecKiPO adicionou a capacidade de representação das violações às regras de negócio, possibilitando a definição de níveis de cumprimento, ou *Enforcement Levels*, e formalizando como as regras concebidas por especialistas do negócio são percebidas pelos agentes do processo.

O trabalho apresentou uma série de aplicações em cenários de estudo reais, definindo claramente as opções de utilização e as diferenças semânticas entre os diversos construtos de regras de negócio adicionados. Os cenários de estudo foram utilizados para avaliar os conceitos propostos neste trabalho em função dos critérios de qualidade propostos por VRANDEčIć (2010), **Acurácia**, **Completude**, **Concisão**, **Adaptabilidade** e **Clareza**, através da análise baseada em *Corpus*, que permitiu a demonstração da expansão da expressividade semântica da KiPO através da representação dos domínios

com os elementos propostos pela DecKiPO, seus conceitos e formalizações através de diversos axiomas em lógica de primeira ordem e modal, além da fundamentação na UFO

A seguir serão discutidas as contribuições e trabalhos futuros que podem ser desenvolvidos a partir dessa proposta.

6.1 Contribuições desse trabalho

A extensão permite que as relações entre atividades realizadas em Processos intensivos em Conhecimento possam ser representadas em termos de restrições de domínio ao invés de um sequenciamento passo-a-passo. Além disso, as adições propostas neste trabalho referentes à modelagem declarativa permitem que quaisquer eventos de um KiP utilizem os conceitos incorporados à KiPO para indicar relações temporais, restrições de integridade comportamental ou relações de possibilidades de escolha de ações em suas instâncias.

Outra contribuição foi a extensão da expressividade de representação das regras de negócio no que tange à definição dos níveis de cumprimento associados a regras de negócio comportamentais. Com os construtos adicionados a partir da SBVR, é possível estabelecer graus de severidade no descumprimento de regras de negócio, permitindo ao especialista do domínio determinar tal nível em nível de modelo, através dos níveis de cumprimento concebidos (*Conceived Enforcement Levels*) e avaliar posteriormente o comportamento dos agentes em nível de instância com os níveis de cumprimento percebidos (*Perceived Enforcement Levels*).

Por fim, este trabalho apresentou uma ferramenta capaz de operacionalizar a tradução de processos modelados no paradigma declarativo utilizando a linguagem LTL Declare, amplamente utilizada no cenário acadêmico. Com essa ferramenta, os modelos de processos descritos nessa linguagem podem ser facilmente convertidos em metamodelos KiPO e enriquecidos através dos conceitos existentes nas demais perspectiva da ontologia.

6.2 Trabalhos futuros

O escopo deste trabalho não engloba a perspectiva de armazenamento dos metamodelos e modelos gerados a partir da DecKiPO em bases de dados não estruturadas. Um trabalho

futuro poderia considerar o desenvolvimento de uma solução onde as regras de negócio modeladas com a partir da KiPO e DecKiPO fossem armazenadas e avaliadas quanto à integridade juntamente com outros aspectos dos processos intensivos em conhecimento. Esse seria um passo em direção à automatização de processos intensivos em conhecimento, especialmente considerando o trabalho de SANTORO e BAIÃO (2018) em direção a *Knowledge-Intensive Process Aware Systems* (KiPAS).

Com a possibilidade de modelagem de Processos intensivos em Conhecimento utilizando o paradigma declarativo, a KiPO passa a permitir que os processos de negócio sejam descritos com mais detalhes, incluindo a percepção dos agentes às regras de negócio e como o fluxo de execução da instância se comporta. Trabalhos futuros podem avaliar o desempenho do processo sob a ótica das regras de negócio e os resultados atingidos, levando-se em conta o quanto os agentes do processo estão aderentes às regras de negócio e como isso está afetando o processo como um todo.

Referências Bibliográficas

ALBUQUERQUE, A.; GUIZZARDI, G. **An Ontological Foundation for Conceptual Modeling Datatypes based on Semantic Reference Spaces**. Research Challenges in Information Science (RCIS). [S.l.]: IEEE Seventh International Conference On. 2013. p. 1-12.

ALLEN, J. F. Maintaining Knowledge about Temporal Intervals. **Communications of the ACM**, v. 26, n. 11, p. 832-843, 1983.

BLEVI, L.; DELPORTE, L.; ROBBRECHT, J. Process mining on the loan application process. **In Business Process Intelligence Challenge (BPIC)**, 2017.

BORST, W. N. **Construction of Engineering Ontologies**. University of Twenty, Enschede, HL: Phd thesis, Centre for Telematica and Information Technology, 1997.

BPTRENDS. **BPTrends: The State of Business Process Management 2016**. BPTrends. [S.l.]. 2016.

BRG. Defining Business Rules - What Are They Really?, 2000. Disponivel em: <http://www.businessrulesgroup.org/first_paper/BRG-whatisBR_3ed.pdf>.

DI CICCIO, C.; MARRELLA, A.; RUSSO, A. A Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches. **Journal on Data Semantics**, v. 4(1), p. pp. 29-57, 2015.

FRANÇA, J. B. S. et al. KIPO: the knowledge-intensive process ontology. **Software Syst Model**, Berlin, Abr 2014.

GÄRDENFORS, P. Conceptual Spaces: the Geometry of Thought. **MIT Press**, USA, 2000.

GRUBER, T. R. **A translation approach to portable ontology specifications**. [S.l.]: [s.n.], 1993.

GUARINO, N. Formal Ontology in Information Systems. **Formal Ontology in Information Systems, FOIS'98**, Trento, Italy, 1998.

GUIZZARDI, G. **Ontological Foundations for Structural Conceptual Models**. University of Twente, The Netherlands: Telematica Instituut Fundamental Research Series No. 15, 2005.

GUIZZARDI, G. On ontology, ontologies, conceptualizations, modeling languages, and (meta) models. **Frontiers in artificial intelligence and application**, 2007. 155.

GUIZZARDI, G. et al. Towards Ontological Foundations for the Conceptual Modeling of Events. In: _____ **Ng W., Storey V.C., Trujillo J.C. (eds) Conceptual Modeling**. [S.l.]: Springer, Berlin, Heidelberg, v. vol 8217, 2013. Cap. Lecture Notes in Computer Science.

GUIZZARDI, G.; FALBO, R. A.; GUIZZARDI, R. S. S. A importância de Ontologias de Fundamentação para Engenharia de Ontologias de Domínio: o caso do domínio de Processos de Software. **IEEE LATIN AMERICA TRANSACTIONS**, Vol. 6, n. NO. 3, July 2008.

GUIZZARDI, G.; WAGNER, G. **A unified foundational ontology and some applications of it in business modeling**. CAiSE'04 Workshops. [S.l.]: [s.n.]. 2004. p. 129-143.

HAGEN, C.; RATZ, D.; POVALEK, R. Towards self-organizing knowledge intensive proccess. J. University on Knowledge Management. **University of Knowledge Management**, 2005.

LOPES, M.; BAIÃO, F.; SIQUEIRA, S. **Expressing business rules in a foundational-based domain ontology:** towards higher-quality conceptual models. International Conference on Information Integration and Web-Based Applications and Services. Paris: [s.n.]. 2010.

MELLOR, S. J. et al. **MDA Distilled:** Principles of Model-driven Architecture. [S.l.]: Addison-Wesley Professional, 2004.

MYLOPOULOS, J. Conceptual modeling and Telos. **In P. Loucopoulos and R. Zicari, editors, Conceptual modeling, databases, and CASE**, Wiley, 1992. chapter 2, pages 49-68.

OLIVEIRA, F. F. Ontology collaboration and its applications. **M.Sc. dissertation. Programa de Pós-Graduação em Informática**, Universidade Federal do Espírito Santo, Vitória, Brazil, 2009.

OMG. **A Proposal for an MDA Foundation Model**. Needham, MA: Object Management Group, 2005.

OMG. Business Process Model and Notation (BPMN), 2011. Disponivel em: <https://www.omg.org/spec/BPMN/2.0/>. Acesso em: 12 Maio 2018.

OMG. **Business Motivation Model**. Version 1.3. ed. [S.l.]: [s.n.], 2015.

OMG. Annex G - EU-Rent Example - SBVR, 2016. Disponivel em: <https://www.omg.org/spec/SBVR/1.4/Annex-G--EU-Rent-Example/PDF>. Acesso em: 25 jun. 2018.

OMG. **Semantics of Business Vocabulary and Business Rules**. Version 1.4. ed. [S.l.]: [s.n.], 2017.

PATEL, C. et al. OntoKhoj: a semantic web portal for ontology searching, ranking and classification. **Proceedings of the 5th ACM international workshop on Web information and data management**, p. 55-61, 2003.

PEREIRA, A. C. T. D.; SANTORO, F. M. Cognitive Decision-Making Process as Context Information. **IOS Press**, 2010.

PESIC, M. Constraint-based workflow management systems: shifting control to users. **PHD thesis, Eindhoven University of Technology, Eindhoven**, 2008.

PESIC, M.; SCHONENBERG, H.; VAN DER AALST, W. M. P. **DECLARE:** Full Support for Loosely-Structured Processes. 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007). Annapolis, MD: [s.n.]. 2007. p. 287-287.

PESIC, M.; SCHONENBERG, H.; VAN DER AALST, W. M. P. **DECLARE:** Full Support for Loosely-Structured Processes. roceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference. Washington, DC, USA: IEEE Computer Society: [s.n.]. 2007.

RAAD, J.; CRUZ, A. **A survey on ontology evaluation**. Proc. of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015). [S.l.]: [s.n.]. 2015. p. 179-186.

SANTORO, F. M.; BAIÃO, F. A. **Knowledge-intensive Process:** A Research Framework. Teniente E., Weidlich M. (eds) Business Process Management Workshops. BPM 2017. Lecture Notes in Business Information Processing. [S.l.]: Springer, Cham. 2018.

SCHWARTZ, B. **The Paradox of Choice**. [S.l.]: Harper Perennial, 2004.

STUDER, R.; BENJAMINS, R.; FENSEL, D. Knowledge Engineering: Principles and Methods. **Data & Knowledge Engineering**, 25 (1,2) 161 (198), 1998.

USCHOLD, M.; JASPER, R. A Framework for Understanding and Classifying. **IJCAI-99 Workshop on Ontologies and Problem-Solving**, Stockholm, Sweden, 1999.

VAN DER AALST, W.; PESIC, M.; AND SCHONENBERG, H. Declarative workflows: Balancing between flexibility and support. **Computer Science-Research and Development**, p. 99-113, 2009.

VAN DER AALST, W.; PESIC, M.; SCHONENBERG, H. Declarative workflows: Balancing between flexibility and support. **Computer Science – Research and Development**, v. 23, p. 99-113, 2009.

VRANDEčIć, D. **Ontology Evaluation**. Karlsruhe, Germany: PhD thesis, Karlsruhe, 2010.

VRANDEčIć, D.; SURE, Y. How to Design Better Ontology Metrics. In: FRANCONI, E.; KIFER, M.; MAY, W. **The Semantic Web:** Research and Applications. ESWC 2007. [S.l.]: Springer, Berlin, v. 4519, 2007.

WAGNER, G. **The Abstract Syntax of RuleML - Towards a General Web Rule Language Framework**. WI '04 Proceedings of the 2004 IEEE/WIC/AC International Conference on Web Intelligence. IEEE Computer Society Washington, DC, USA: [s.n.]. 2004.

WAGNER, G.; GIURCA, A.; LUKICHEV, S. A usable interchange format for rich syntax rules integrating OCL, RuleML and SWRL. **Preceedings of the Workshop of Reasoning of the Web**, Edinburgh, UK, 2006.

WELTY, C. A. . M. R. . C.-C. J. Evaluating ontological analysis. **Semantic Integration Workshop (SI-2003)**, October 2003.

Anexo I – Descrição dos conceitos da DecKiPO

Definição dos conceitos adotados pela Ontologia de Processos intensivos em Conhecimento (KiPO)

Conceito da KiPO	Perspectiva da KiPO	Descrição
Knowledge-intensive Process (KiP)	KiPO Core (KiPCO)	Processos que envolvem tomadas de decisão complexas, não seguem uma sequência previsível e repetível, envolvem interação e colaboração entre as partes interessadas e se apoiam fortemente no conhecimento e experiência de seus agentes (HAGEN, RATZ e POVALEK, 2005).
Knowledge-intensive Activity (KiA)	KiPO Core (KiPCO)	Atividade que visa alcançar um objetivo e é executada por um agente que possui uma intenção a ser atingida. Um agente de inovação incorpora inovação na execução de atividade intensiva em conhecimento (FRANÇA, NETTO, *et al.*, 2014).
Agent	KiPO Core (KiPCO)	Agentes são participantes do processo que têm suas ações motivadas por seus desejos, executam interações comunicativas, são comprometidos a realizar suas intenções, e apresentam crenças, sentimentos e experiência prévia. Os agentes em processos intensivos em conhecimento podem ser do tipo agente de inovação e agente de impacto, e podem assumir os papéis de remetente e destinatário de mensagens (FRANÇA, NETTO, *et al.*, 2014).
Contingency	KiPO Core (KiPCO)	Evento externo e imprevisível que influencia a execução do processo. A Contingência é responsável por determinar a

Conceito da KiPO	Perspectiva da KiPO	Descrição
		execução de atividades não previstas (FRANÇA, NETTO, *et al.*, 2014).
Foundational Business Rule	Business Rule Ontology (BRO)	É uma declaração que define ou restringe algum aspecto de uma organização, sendo atômica, de forma que não pode ser dividida. Tem como objetivo afirmar a estrutura de um negócio ou controlar ou influenciar o comportamento deste (FRANÇA, NETTO, *et al.*, 2014).
Foundational Event	Business Rule Ontology (BRO)	Representação de um indivíduo composto por partes temporais (GUIZZARDI, WAGNER, *et al.*, 2013).
Foundational Integrity Rule	Business Rule Ontology (BRO)	Este tipo de regra é estrutural. Ela não altera o domínio, não cria novo evento ou ação que o altere. Esta regra somente restringe algo já existente entre os conceitos já previstos. (LOPES, BAIÃO e SIQUEIRA, 2010)
Foundational Derivation Rule	Business Rule Ontology (BRO)	Regra que possui condição e conclusão e que explica como um elemento do modelo pode ser derivado. Ela representa a derivação de novos conceitos no domínio a partir de conhecimento já existente no domínio sendo modelado, e apresenta uma condição anterior à derivação. Quando o estado do domínio satisfizer essa condição, uma conclusão acontecerá, acrescentando um novo elemento ao domínio (LOPES, BAIÃO e SIQUEIRA, 2010).
Foundational Reaction Rule	Business Rule Ontology (BRO)	Também conhecida como regra Evento-Condição-Ação (regra ECA, Event-Condition-Action), são afirmações que, no caso de ocorrência de um evento disparador ou no caso de um conjunto de condições ser satisfeito, especificam a execução de uma ou mais ações. Opcionalmente, depois da execução de uma ação, pós-condições podem ser verdadeiras. (LOPES, BAIÃO e SIQUEIRA, 2010)

Definição dos conceitos adicionados pela DecKiPO

Conceito DecKiPO	Especializada de	Descrição
Foundational Definitional Business Rule	BRO::Foundational Business Rule	Regra de fundamentação que descreve conceitos e regras de negócios "necessárias" em um determinado estado de coisas e representam regras aléticas, ou seja, que não podem ser violadas
Foundational Behavioral Business Rule		Regra de fundamentação que descreve conceitos e regras de negócios que são "obrigadas" em um determinado estado de coisas, representando regras deônticas, ou seja, que podem pode ser violadas.
Foundational Existence Rule	BRO::Foundational Behavioral Business Rule	Regra de fundamentação que descreve regras de negócio que denotem a necessidade da existência (ou inexistência) da execução de um evento durante a instância de um processo de negócio.
Foundational Relation Rule		Regra de fundamentação que descreve regras de negócio que definem um tipo de relacionamento entre dois eventos .
Foundational Choice Rule		Regra de fundamentação que descreve as regras de negócio que limitam o número de alternativas que podem ser selecionadas pelo Agente dentre as atividades existentes no processo de negócio.
Existence	BRO::Foundational Existence Rule	Restringe a obrigatoriedade de existência de, no mínimo, uma determinada quantidade de instâncias de eventos em uma instância de um processo de negócio.
Absence		Restringe a obrigatoriedade de existência de, no máximo, uma determinada quantidade de instâncias de eventos em uma instância de um processo de negócio.
Exactly		Restringe a obrigatoriedade de existência da quantidade exata determinada de instâncias de eventos em uma instância de um processo de negócio.

Conceito DecKiPO	Especializada de	Descrição
Responded Existence	BRO::Foundational Relation Rule	Restringe uma relação de obrigatoriedade da existência de uma instância de um evento B em qualquer momento antes ou algum momento depois do evento A.
CoExistence		Restringe uma relação de obrigatoriedade quanto à dependência mútua de instâncias de dois eventos.
Response		Restringe uma relação de obrigatoriedade onde sempre que um evento A ocorre, um evento B deverá ocorrer eventualmente depois de A.
Precedence		Restringe uma relação de obrigatoriedade onde sempre que um evento B ocorre, um evento A deve ter ocorrido em algum momento antes de B.
Succession		Restringe uma relação de obrigatoriedade onde sempre que um evento A ocorre, um evento B ocorre eventualmente depois e um evento B sempre ocorre eventualmente depois de um evento A, nessa ordem.
Alternate Response		Restringe uma relação de obrigatoriedade onde, se um evento A ocorre, um evento B deve ocorrer eventualmente depois de A para que uma nova instância de A possa ser executada.
Alternate Precedence		Restringe uma relação de obrigatoriedade onde, se um evento B ocorre, um evento A deve ter ocorrido antes de B. Um novo evento B só pode ser realizado se um novo evento A ocorrer antes de B.
Alternate Succession		Restringe uma relação de obrigatoriedade onde, os eventos A e depois B precisam ocorrer eventualmente nessa ordem para que novas instâncias de A e B possam ocorrer.
Chain Response		Restringe uma relação de obrigatoriedade onde o evento B ocorre sempre imediatamente depois do evento A.
Chain Precedence		Restringe uma relação de obrigatoriedade onde o evento A ocorre sempre imediatamente antes do evento B.
Chain Succession		Restringe uma relação de obrigatoriedade onde os eventos A e B sempre ocorrem imediatamente um após o outro, nessa ordem.

Conceito DecKiPO	Especializada de	Descrição
Not Responded Existence	BRO::Foundational Relation Rule	Restringe uma relação de obrigatoriedade da inexistência de uma instância de um evento B em qualquer momento antes ou algum momento depois do evento A.
Not CoExistence		Restringe uma relação de obrigatoriedade onde os eventos A e B não podem ocorrer na mesma instância de um processo de negócio.
Not Response		Restringe uma relação de obrigatoriedade onde sempre que um evento A ocorre, um evento B não deverá ocorrer depois de A.
Not Precedence		Restringe uma relação de obrigatoriedade onde sempre que um evento B ocorre, um evento A não deve ter ocorrido em qualquer momento antes de B.
Not Succession		Restringe uma relação de obrigatoriedade onde sempre que um evento A ocorre, um evento B não ocorre eventualmente depois e um evento B não ocorre eventualmente depois de um evento A, nessa ordem.
Not Chain Response		Restringe uma relação de obrigatoriedade onde o evento B nunca ocorre imediatamente depois do evento A.
Not Chain Predecence		Restringe uma relação de obrigatoriedade onde o evento A nunca ocorre imediatamente antes do evento B.
Not Chain Succession		Restringe uma relação de obrigatoriedade onde os eventos A e B sempre nunca ocorrem imediatamente um após o outro, nessa ordem.
Choice	BRO::Foundational Choice Rule	Restringe a escolhe de, no mínimo, uma quantidade determinada de atividades dentre as possibilidades. As demais atividades podem continuar sendo executadas em momentos posteriores.
Exclusive Choice		Restringe a escolha de, no mínimo, uma quantidade determinada de atividades dentre as possibilidades. As demais atividades não podem mais ser executadas na instância do processo de negócio.

Conceito DecKiPO	Especializada de	Descrição
Enforcement Level	UFO-A::Quality Universal	Valor em uma escala ordenada que especificam a severidade de consequência imposta de forma garantir o cumprimento de uma Foundational Behavioral Business Rule.
Conceived Enforcement Level	BRO::Enforcement Level	Nível de cumprimento designado por um Agente à regra de negócio que define o grau de expectativa de cumprimento da regra de negócio e a severidade da punição imposta caso seja violada.
Perceived Enforcement Level	UFO-A::Relator	Nível de cumprimento que representa a percepção da regra de negócio pelo Agente no momento da tomada de decisão.
Enforcement Level Structure	UFO-A::Quality Structure	Representação conceitual do conjunto de todos os valores possíveis que um Quality Universal pode receber.
Severity Level	BRO::Enforcement Level Structure	Escala qualitativa do grau de severidade imposto a uma regra de negócio.