

# Uma Abordagem Lógica para Validação Automática de Modelos de Processos Intensivos em Conhecimento

## Title: A Logic-Based Approach to Automatically Validate Knowledge-intensive Processes

Tatiana Barboza<sup>1</sup>, Fernanda Araujo Baião<sup>2</sup>, Flavia Maria Santoro<sup>3</sup>

<sup>1</sup>IT Department – Fluminense Federal University (UFF)  
Niterói, RJ – Brazil

<sup>2</sup> Department of Informatics – Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, RJ – Brazil

<sup>3</sup>Institute of Mathematics and Statistics (IME) – University of the State of Rio de Janeiro (UERJ), Rio de Janeiro, RJ – Brazil

tatiananti@gmail.com, fbaiao@inf.puc-rio.br, flavia@ime.uerj.br

**Abstract.** *With the recent advances on Business Process Management (BPM) research and practice, organizations are changing their focus towards critical business processes that are poorly structured, dynamic and highly complex, known as Knowledge-intensive Processes (KiP). Due to their characteristics, typical BPM activities such as modeling, instantiation, validation and simulation of process models poses many challenges. This work proposes a rule-based strategy to instantiate, validate and simulate KiP models, and presents KIPAlloy, a computational tool that supports KIP modelling. The proposed strategy considers the Knowledge-intensive Process Ontology (KiPO) as a metamodel for modelling a KiP and transforms its rules into specifications in the Alloy logic-based language. We evaluate the applicability of the proposal in four different scenarios, in which we illustrate the use of the proposed KIPAlloy tool and discuss its benefits to process modelers.*

**Keywords.** *Model Validation; Knowledge-intensive Process; Conceptual Modeling; BPM.*

**Resumo.** *Com o recente avanço da pesquisa e prática em Gestão de Processos de Negócio (Business Process Management – BPM), organizações vêm mudando o seu foco para processos críticos que tipicamente são pouco estruturados, dinâmicos e de alta complexidade, denominados Processos intensivos em Conhecimento (Knowledge-intensive Processes – KiP). Em função de tais características, atividades típicas de BPM como modelagem, instanciação, validação e simulação de processos apresentam muitos desafios. Este artigo propõe uma estratégia baseada em regras para instanciar, validar e simular modelos de KiPs, e apresenta a KIPAlloy, uma ferramenta computacional de apoio à modelagem de KiPs. A proposta considera a ontologia de processos intensivos em conhecimento (KiPO) como um*

*metamodelo para modelar um KiP, e transforma as suas regras em especificações na linguagem Alloy. A aplicabilidade de proposta foi avaliada em um estudo exploratório com 4 cenários nos quais é ilustrada a aplicabilidade da ferramenta KiPAlloy proposta e os resultados evidenciam os benefícios para modeladores de processos.*

**Palavras-Chave.** Validação de Modelos; Processos intensivos em Conhecimento; Modelagem Conceitual; BPM.

## 1. Introdução

Processos de negócio são o que as empresas fazem sempre que entregam um serviço ou um produto aos seus clientes [Dumas et al. 2018]. Um Processo Intensivo em Conhecimento (em inglês, *Knowledge-intensive Process* – KiP) é um processo de negócio pouco estruturado, que apresenta um alto grau de complexidade e dinamismo. Além disso, tais processos envolvem conceitos subjetivos, tais como inovação, colaboração, interação entre agentes, racional para tomadas de decisão, dentre outros [Di Ciccio et al. 2015][Maldonado 2008].

A Teoria da Criação do Conhecimento Organizacional (*Organizational Knowledge Creation*), definida por Nonaka e Takeuchi (1995) e Nonaka et al (1996), é a base teórica desta pesquisa. De acordo com esta teoria, toda organização que quer progredir na sociedade do conhecimento deve garantir que exista uma sinergia entre a Tecnologia da Informação e Comunicação e pessoas com capacidades colaborativas para promover a criação e evolução de conhecimento. Novo conhecimento organizacional (tácito ou explícito) é criado através de interações entre indivíduos. Este processo social e epistêmico configura-se em quatro modos de conversão do conhecimento: socialização (do conhecimento tácito individual ao conhecimento tácito do grupo), externalização (do conhecimento tácito ao conhecimento explícito), combinação (do conhecimento explícito separado ao conhecimento explícito sistêmico), internalização (do conhecimento explícito ao conhecimento tácito). Neste artigo, argumentamos que organizações que realizam processos intensivos em conhecimento devem explicitar seus processos através da modelagem para promover aprendizagem contínua. Porém, quanto maior a intensidade em conhecimento de um processo, menor o seu grau de estruturação, e desta forma mais difícil se torna descobrir e gerar modelos para estes processos [Nascimento et al. 2015].

Para tratar o problema da modelagem de processos desta natureza, França et al. (2015) propuseram uma ontologia para Processos Intensivos em Conhecimento, a KiPO (*Knowledge-intensive Process Ontology*) que apresenta uma conceitualização rica semanticamente, bem fundamentada na UFO (*Unified Foundational Ontology*) [Guizzardi 2005], e que define grupos de conceitos relevantes para um KiP. As sub-ontologias que compreendem a KiPO tratam das suas perspectivas de colaboração (*Collaboration Ontology* – CO), de processo de Negócio (*Business Process Ontology* – BPO), de decisão (*Decision Ontology* – DO), de regras de negócio (*Business Rule Ontology* – BRO), e o núcleo do processo intensivo em conhecimento (*Knowledge-intensive Process Core Ontology* – KiPCO). A KiPO funciona como um metamodelo para apoiar a modelagem de um KiP. A partir desta ontologia, observa-se que um KiP deve ser representado por suas diversas perspectivas, e não apenas por um modelo tradicional baseado em controle de fluxo de atividades.

Mylopoulos (1992) define modelagem conceitual como “a atividade de formalmente descrever alguns aspectos do mundo físico e real para propor entendimento e comunicação”. Desta forma, modeladores percebem o mundo físico e real e, a partir daí, expressam esses conceitos em um modelo. Um dos problemas mais frequentes na área de modelagem é encontrar ferramentas computacionais que consigam validar ou simular modelos (semi) automaticamente, a partir de regras semânticas e sintáticas já pré-estabelecidas. Por exemplo, uma regra simples para a modelagem de processos estabelece que uma Atividade deve ser executada por um Agente. Os modelos também são enriquecidos com a definição de regras que estabelecem formalmente as relações entre os conceitos. A vantagem da disponibilidade de regras é que as críticas que estas impõem aos modelos minimizam os possíveis erros de modelagem, tornando o trabalho do modelador mais simples. No entanto, no caso de processos intensivos em conhecimento, observamos que o modelador encontra dificuldades em saber como relacionar os elementos usando as regras da KiPO, pois esta ontologia ainda não possui uma estratégia automática para validar e simular modelos. A análise da literatura evidenciou que as propostas existentes contemplam a execução de processos de forma geral (regras que determinam o fluxo de atividades), mas não lidam com a riqueza semântica da KiPO, que inclui elementos e restrições adicionais em diversas perspectivas relevantes para um KiP sendo, portanto, insuficientes para solucionar integralmente o problema tratado nesta pesquisa.

Portanto, o presente trabalho trata a seguinte questão de pesquisa **“Quais são os resultados possíveis de uma validação automática de modelos de processos intensivos em conhecimento”**? Neste sentido, propõe-se transformar as regras baseadas na KiPO em especificações existentes na linguagem baseada em lógica *Alloy* [Jackson 2008]. Além disso, a ferramenta *Alloy Analyzer* permite a exploração do modelo, gerando estruturas e checando propriedades, podendo resultar em contraexemplos. Ao contrário de uma linguagem de programação, a linguagem *Alloy* é declarativa, e por isso permite que modelos muito sucintos e parciais sejam construídos e analisados. A linguagem *Alloy* é similar a linguagens de especificação formal, tais como Z, VDM, Larch, B, OBJ mas, ao contrário de todas elas, é passível de análise totalmente automática no estilo de um verificador de modelo. Para avaliar estas transformações, foi realizado um estudo exploratório, de modo a avaliar as evidências sobre a comprovação ou não da aplicabilidade da linguagem *Alloy* para o contexto da KiPO. No estudo exploratório realizado, 4 modelos de processos intensivos em conhecimento diferentes foram elaborados e simulados a partir das regras da KiPO convertidas para a linguagem *Alloy*, usando a ferramenta *Alloy Analyzer*.

O presente trabalho estende o trabalho anterior de Barboza, Baião e Santoro (2019) com uma análise mais ampliada e aprofundada da literatura sobre o tema, acrescenta simulações e resultados obtidos na ferramenta *Alloy Analyzer*, e propõe a ferramenta *KiPALloy*, para apoiar todo o processo de modelagem e validação automática de modelos de processos intensivos em conhecimento, interagindo diretamente com o *Alloy*.

Este artigo é organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica, a Seção 3 descreve trabalhos relacionados à validação de modelos; a Seção 4 descreve a metodologia e definições propostas, a Seção 5 apresenta o estudo exploratório realizado com diferentes cenários; a Seção 6 descreve a ferramenta *KipoAlloy* e ilustra

seu uso através de um dos cenários do estudo exploratório; a Seção 7 discute os resultados desta pesquisa, e a Seção 8 conclui o artigo e aponta os trabalhos futuros em andamento.

## 2. Fundamentação Teórica

### 2.1. Processos Intensivos em Conhecimento

De acordo com Richter-Von Hagen et al. (2005), processos de negócios podem ser classificados em três tipos: estruturados, semiestruturados e não estruturados. Os processos estruturados são completamente predefinidos, ou seja, existem regras fixas que não podem ser alteradas para executar cada atividade. Os processos semiestruturados contêm partes estruturadas e não estruturadas; nem todas as atividades possuem regras pré-definidas em relação aos próximos passos no fluxo.

Os processos não estruturados são imprevisíveis e não apresentam uma ordem pré-definida para a execução das atividades, sendo comumente chamados de processos intensivos em conhecimento, ou em inglês *Knowledge-intensive Processes* (KiP). Tais processos apresentam um alto grau de liberdade para alcançar seus objetivos, pois o valor é criado dentro do processo cumprindo os requisitos de conhecimento dos participantes. Assim, as decisões tomadas durante a realização das atividades em um KiP são diretamente influenciadas pelo conhecimento de quem as executa e, portanto, não são passíveis de automatização.

Di Ciccio et al. (2015) definiram o PIC como processo cuja condução e execução é fortemente dependente de agentes do conhecimento que realizam várias tarefas de tomada de decisão intensivas em conhecimento interconectadas. Estes autores também apontaram como características típicas de um KiP ser: orientado pelo conhecimento e objetivos; orientado para a colaboração; imprevisível; emergente; conduzido por eventos e restrições. Ainda segundo Di Ciccio et al. (2015), um KiP apresenta 5 elementos essenciais (trabalhadores de conhecimento, objetivos, elementos de dados e conhecimento, ações de conhecimento, regras e restrições), para os quais 25 requisitos foram definidos. Em particular, para o elemento "regras e restrições" os autores definem como requisito "Formalizar regras e restrições", que é endereçado na presente proposta.

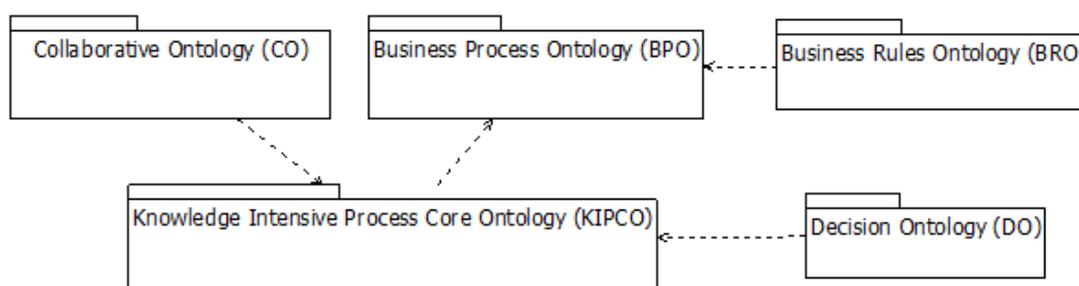
Exemplos de KiP incluem suporte ao cliente, design de novos produtos e serviços, controle de tráfego aéreo, planejamento de campanhas de marketing, governança de TI e planejamento estratégico, análise de incidentes, elaboração de trabalho científico, gestão de emergências, diagnóstico e tratamento médico, além de processos denominados por Di Ciccio et al. (2015) como "Artful", que incluem coordenação de projetos de pesquisa internacionais, publicação de trabalhos científicos, e engenharia empresarial. Por todas estas características, há uma grande dificuldade de modelar e gerenciar estes processos.

### 2.2. Knowledge-intensive Process Ontology (KiPO)

França et al. (2015) propuseram a KiPO (*Knowledge-intensive Process Ontology*), uma ontologia bem fundamentada para Processos Intensivos em Conhecimento. A KiPO representa conceitualização rica semanticamente, cujos conceitos são mais precisamente definidos através das meta-propriedades da ontologia de alto nível UFO (*Unified Foundational Ontology*) [Guizzardi 2005]. A KiPO compreende 5 sub-ontologias: ontologia de Colaboração (CO), ontologia de Processo de Negócio (BPO), ontologia de Decisão (DO), ontologia de Regras de Negócio (BRO), e ontologia Núcleo do Processo

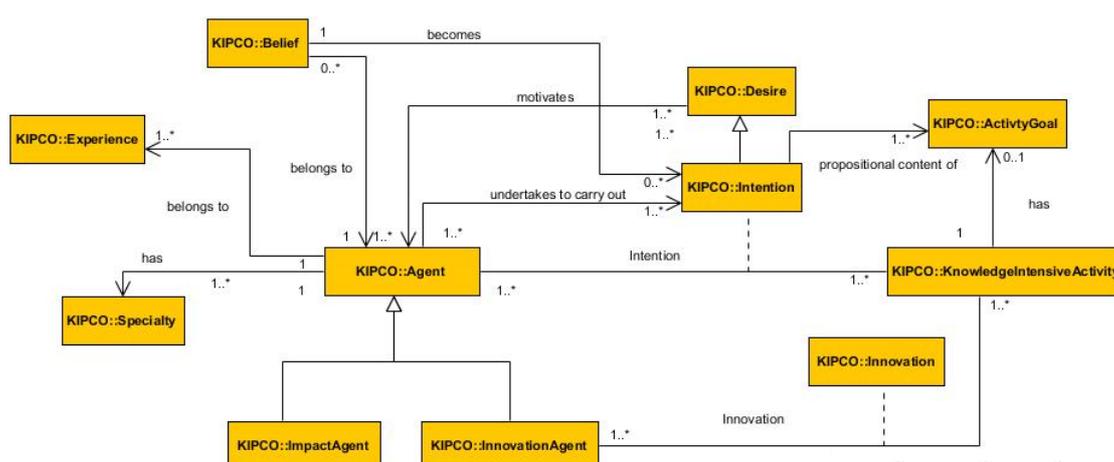
Intensivo em Conhecimento (KiPCO). A Figura 1 mostra os relacionamentos entre as sub-ontologias da KiPO.

O componente central da KiPO é a ontologia do Núcleo do Processo Intensivo do conhecimento (KiPCO) (Figura 1), que reúne elementos específicos de KiP, e relaciona conceitos das outras sub-ontologias. Conforme a Figura 2, um Agente é aquele que intencionalmente se compromete a atingir um Objetivo executando uma Atividade (Intensiva em Conhecimento). O Agente é motivado por seu Desejo e age de acordo com suas Crenças. Essas ideias seguem a teoria do BDI proposta por Bratman (1987) e amplamente adotada em vários trabalhos, sendo o de Rao e Georgeff (1991) um dos mais influentes.



**Figura 1: Sub-Ontologias que compreendem a KiPO (França et al. 2015).**

Um Agente de um KiP pode desempenhar o papel de um Agente de Inovação ou de um Agente de Impacto (França et al., 2015). O Agente de Inovação é especialista (tem Especialidade) em algumas áreas relacionadas ao domínio do KiP e, portanto, normalmente contribui para a execução de uma atividade intensiva em conhecimento com inovação e criatividade. Um Agente de Impacto é o responsável pela execução de uma atividade intensiva em conhecimento e pela identificação de questões e tomadas de decisão durante sua execução.



**Figura 2: Sub-Ontologia Núcleo do Processo Intensivo em Conhecimento (KiPCO). Adaptado de (França et al. 2015).**

### 2.3. Linguagem Alloy e Ferramenta Alloy Analyzer

Jackson [8] afirma que *Alloy* é uma linguagem criada para descrever propriedades estruturais. A linguagem *Alloy* é uma lógica simples, porém expressiva, baseada na noção de relações, e foi inspirada na linguagem de especificação Z e no cálculo relacional de Tarski. *Alloy* disponibiliza uma declaração de sintaxe (inspirada em OMT e UML) compatível com modelos de objetos gráficos e uma sintaxe de fórmulas cujo conjunto é suficiente para expressar regras complexas e expressões de navegação, inclui um rico mecanismo de subtipos que permite fatoração de modelos, e ainda é acessível para análise semântica automática.

*Alloy Analyzer* é uma ferramenta que, partindo das regras de um modelo, descobre estruturas que os satisfazem, e portanto, testa e sugere soluções possíveis (na forma de contraexemplos) para problemas em modelos. Esta ferramenta pode ser usada para explorar os modelos, gerar amostras de estruturas, checar as propriedades do modelo e gerar contraexemplos. As estruturas são apresentadas graficamente, e sua aparência pode ser customizada para um determinado domínio de forma manual. Alloy Analyzer fornece duas funcionalidades principais: simulação e verificação de asserção. De maneira geral, a simulação implica na modelagem de um processo ou sistema, de tal forma que o modelo imite as respostas do sistema real em uma sucessão de eventos que ocorrem ao longo do tempo (Banks et al., 2010). A simulação produz uma instância aleatória do modelo, que está de acordo com a especificação, garantindo que o modelo desenvolvido não seja inconsistente. Asserções são restrições, que o modelo precisa satisfazer. O Alloy Analyzer funciona traduzindo fórmulas *Alloy* para expressões booleanas, que são analisadas por solucionadores SAT embutidos no analisador. Um escopo especificado pelo usuário nos elementos do modelo limita o domínio. Se for encontrada uma instância que viola uma asserção, a asserção não será válida

Na linguagem *Alloy*, fato (*fact*) é uma declaração onde se pode explicitar uma regra (ou uma lista de regras) de um modelo. A ferramenta descarta qualquer violação de qualquer fato. Se o fato for trivialmente falso ela não fornecerá exemplos, mas não será informado que o modelo é inconsistente.

O exemplo a seguir trata do domínio de diretórios e arquivos, e ilustra que tanto um arquivo (File) quanto um diretório (Dir) são extensões (ou especializações) do conceito sistema de arquivos (FsObject):

```
fact { File + Dir = FSObject }
```

Assinatura (signature) é uma forma simples de declarar um conceito na linguagem *Alloy*. O corpo da assinatura é apresentado entre um par de chaves, onde pode ser definida uma série de relações para os quais o conjunto definido pela assinatura é o domínio. Por exemplo, para indicar que um sistema de arquivos (FSObject) é pai (parent) de um ou nenhum (lone) diretório (Dir), seria usada a seguinte assinatura:

```
// A file system object in the file system
```

```
sig FSObject { parent: lone Dir }
```

Na próxima seção, são apresentados trabalhos de validação de modelos similares à proposta deste artigo.

### 3. Trabalhos Relacionados

Os trabalhos relacionados a esta pesquisa podem ser agrupados em duas categorias: (i) o uso e desenvolvimento de soluções baseadas em *Alloy* para outros domínios, e (ii) linguagens e ferramentas de modelagem e simulação de processos não estruturados em geral.

Várias aplicações do *Alloy* são relatadas na literatura em diferentes ambientes. Borbar et al. (2007) apresentam um estudo envolvendo *UML2Alloy*, uma ferramenta para transformar Diagramas de Classe da UML com regras em OCL (Object Constraint Language) para *Alloy*. Esta transformação permite analisar os modelos via *Alloy* para checar consistência. Os autores também apresentam um método para transformação automática que converte instâncias do *Alloy* em Diagramas de Objetos da UML.

Benevides et al. (2010) propõem facilitar o processo de validação de modelos conceituais definidos em OntoUML ao transformarem estes modelos em especificações na linguagem *Alloy* e usando o seu analisador para gerar instâncias de modelos e asserções de contraexemplos. Para permitir a observação de sequências de snapshots de instância de modelo, as dinâmicas de criação, classificação, associação e destruição de objetos são apresentadas.

Guerson et al. (2014) (2015) discutem como a OCL pode ser usada para enriquecer uma ferramenta de avaliação de modelos em OntoUML. Nesta abordagem, expressões OCL são transformadas na linguagem do *Alloy* possibilitando a validação de modelos e a verificação de asserções com o *Alloy Analyzer*.

Outros trabalhos propuseram extensões do *Alloy Analyzer*. Montaghani et al. (2017) desenvolveram o protótipo Bordeaux, que tem como objetivo encontrar exemplos externos para ajudar o usuário a encontrar *bugs* com restrição excessiva em seu modelo. Nelson et al. (2017) apresentam Amalgam, que implementa busca de cenários e fornece uma exploração interativa de exemplos desta busca. Já Milicevic et al. (2015) mostraram como estender um solucionador de primeira ordem (neste caso, Kodkod, um localizador de modelo para lógica relacional usada como o mecanismo do *Alloy Analyzer*) para que ele possa lidar com quantificações de mais alta ordem.

No segundo grupo de trabalhos relacionados, encontram-se ferramentas que simulam modelos de processos não estruturados ou declarativos. O DCR Graphs (Hildebrandt e Mukkamala, 2010) possibilita descrever e simular fluxos de trabalho e processos de negócio de forma interativa, utilizando a linguagem DCR. Seguindo a mesma abordagem, as ferramentas Declare (Westergaard e Maggi, 2011) e DPILNavigator (Schonig et al., 2017) simulam modelos para facilitar a modelagem de processos pouco estruturados. O objetivo do Declare é garantir flexibilidade ao modelar livremente processos estruturados e, ao mesmo tempo, fornecer suporte à decisão durante a execução. Já o DPILNavigator compreende uma interface web que permite aos participantes do processo escolher e executar tarefas e oferece gerenciamento de conteúdo corporativo. As Tabelas 1 e 2 resumem e comparam os trabalhos dos dois grupos identificados.

**Tabela 1 - Soluções baseadas em Alloy em outros domínios.**

Propostas analisadas	Objetivo	Vantagens	Limitações para a proposta corrente
UML2Alloy Borbar et al. (2007)	Transformar Diagramas de Classe da UML com em regras em OCL	Analisar modelos em UML que é uma linguagem padrão Gerar modelos de objetos (instâncias)	A KiPO é fundamentada na UFO, portanto análise de modelos em UML não é suficiente
Benevides et al. (2010)	Validação de modelos em ONTOML	Permite análise detalhada de modelos baseados na UFO	A KiPO possui um conjunto de regras específicas que não estão especificadas nesta implementação
Guerson et al. (2014)(2015)	Transformação de regras OCL em linguagem Alloy	OCL é a linguagem padrão de definição de regras associada à UML	A KiPO não utiliza OCL para definição de regras.
Kodkod Milicevic et al. (2015)	Extensão para solucionador de primeira ordem	Lidar com quantificações de mais alta ordem	Não faz parte do escopo desta proposta
Bordeaux Montaghami et al. (2017)	Busca de exemplos externos para ajudar um usuário a encontrar <i>bugs</i> com restrição excessiva em seu modelo	Estender a validação para exemplos de outros cenários	A idéia implementada nesta proposta poderia ser usada para trabalhos futuros do trabalho atual.
Amalgam Nelson et al. (2017)	Busca de cenários e exploração interativa de exemplos	Ferramenta interativa	A solução proposta nesta pesquisa inclui uma ferramenta interativa para interface com Alloy voltada especificamente para a KiPO

**Tabela 2 - Ferramentas de suporte a processos intensivos em conhecimento.**

Propostas analisadas	Objetivo	Vantagens	Limitações
DCRGraphs (Hildebrandt e Mukkamala, 2010)	Modelagem e simulação de processos de negócio de forma interativa	Interface simples e interativa	Não contemplam todos os elementos de um processo intensivo em conhecimento, portanto não atendem a modelagem em KiPO
Declare (Westergaard e Maggi 2011)	Simulação de modelos de processos pouco estruturados	Inclui um amplo conjunto de templates para representação de restrições	
DPILNavigator (Schonig et al. 2017)	Simulação de modelos de processos pouco estruturados	Interface Web	

Estas aplicações contemplam de maneira geral a execução do processo, ou seja, regras que determinam a sequência de atividades a serem seguidas, mas não são específicas para processos intensivos em conhecimento, ou seja, não lidam com uma série de elementos e restrições que as perspectivas da KiPO propõem. Portanto, não são adequadas para solucionar integralmente o problema tratado nesta pesquisa

#### 4. Transformação das Regras da KiPO para a Linguagem Alloy

A sub-ontologia KiPCO foi selecionada para este artigo, pois ela contém as classes básicas inerentes a um KiP, que são: Knowledge Intensive Activity (Atividade Intensiva em Conhecimento), Belief (Crença), Intention (Intenção), Agent (Agente), InnovationAgent (Agente de Inovação), Impact Agent (Agente de Impacto), Innovation (Inovação), Activity Goal (Objetivo da Atividade), Experience (Experiência), Specialty (Especialidade) e Desire (Desejo), mostradas na Figura 2. A transformação para a linguagem do *Alloy* foi realizada da seguinte maneira:

- i. Entidades foram convertidas em assinaturas na Linguagem *Alloy* (Ex.: sig Specialty{ }).
- ii. Nomes dos relacionamentos são os mesmos para a linguagem *Alloy*, porém os que eram idênticos foram acrescidos com o nome da entidade para onde a direção vai (Ex.: hasInnovation : Innovation).
- iii. Cardinalidades foram transformadas em quantificadores. (Ex.: 0..1 → lone).
- iv. Regras (axiomas) da KiPCO foram convertidas em fatos da linguagem *Alloy*. (Ex.: act{undertakesToCarryOut in Agent some -> some Intention}.

A partir deste modelo KiPCO construído no *Alloy*, todos os modelos de processos intensivos em conhecimento nos diversos domínios construídos com base neste metamodelo podem ser validados e simulados. O modelo completo é apresentado na próxima seção. A Figura 3 mostra a codificação dos elementos da sub-ontologia KiPO (cada uma das classes e seus relacionamentos) para a linguagem *Alloy*. Esta codificação permite que modelos construídos sejam validados.

#### 5. Estudo Exploratório

A partir da dificuldade que modeladores encontram em validar os modelos de domínio construídos usando a KiPO, esta foi codificada em *Alloy*. Para avaliar esta proposta, foi escolhido o estudo exploratório como método de pesquisa (Yin 2015) (Runoson, 201). De acordo com Yin (2015), se as questões de pesquisa se concentram principalmente nas perguntas “o que”, surge a possibilidade de que a pesquisa seja do tipo exploratória. Questões que respondem a argumentos como “o que pode ser aprendido” a partir de um estudo de caso, justificam a condução de um estudo exploratório, cuja meta seria desenvolver hipóteses e proposições para investigação posterior. Como já apresentado na Introdução, a questão formulada nesta pesquisa, e que será avaliada através de um estudo exploratório, é: **quais são os resultados possíveis de uma validação automática de modelos de processos intensivos em conhecimento?**

Busca-se, portanto, descobrir se é possível encontrar resultados a partir do teste de modelos construídos através da KiPO convertidos para *Alloy* em casos reais. O método de estudo exploratório permite avaliar as evidências sobre a aplicabilidade da linguagem *Alloy* para o contexto da KiPO, além de levantar questões para aprofundamento da pesquisa.

```

sig Agent { especializacao_impact : ImpactAgent, especializacao_innovation :
InnovationAgent, undertakesToCarryOut: some Intention }
sig ActivityGoal{}
sig InnovationAgent{ hasSpecialty : some Specialty, innovated: some Innovation}
sig ImpactAgent{}
sig Innovation{}
sig Desire { motivates: lone Agent }
sig Intention extends Desire{ propositionalContentsOf : some ActivityGoal }
sig Specialty {}
one sig Belief {becomes: some Intention, belongsTo: some Agent}
sig KnowledgeIntensiveActivity{ hasIntention: some Intention, hasGoal : lone
ActivityGoal, hasInnovation : some Innovation }
fact{undertakesToCarryOut in Agent some -> some Intention}
fact{hasIntention in KnowledgeIntensiveActivity some-> some Intention}
fact{hasGoal in KnowledgeIntensiveActivity one ->lone ActivityGoal}
fact{hasInnovation in KnowledgeIntensiveActivity some->some Innovation}
fact{innovated in InnovationAgent some-> some Innovation}
fact{hasSpecialty in InnovationAgent some -> some Specialty}
fact{motivates in Desire some -> some Agent}
fact{propositionalContentsOf in Intention some -> some ActivityGoal}
fact{becomes in Belief one -> some Intention}
fact{belongsTo in Belief some -> some Agent}

```

**Figura 3: Modelo KiPCO na linguagem Alloy**

O estudo foi planejado de forma a permitir a observação do fenômeno (validação de modelos construídos a partir da KiPO) em distintos cenários (processos) sob condições semelhantes (implementação das regras na linguagem Alloy). Os cenários foram escolhidos de forma a apresentarem variedade de domínios e disponibilidade de modelos de processos. A unidade de análise foi a quantidade de erros de modelagem que a simulação é capaz de identificar. O objetivo não foi realizar análises comparativas entre os cenários, e em análises quantitativas aplicando métricas, pois o estudo teve caráter exploratório. Desta forma, foi feita uma análise qualitativa a partir dos resultados.

O estudo exploratório foi composto de 4 cenários, em contextos diferentes, com objetivo de avaliar se a linguagem *Alloy* consegue automatizar o processo de validação, trazendo soluções (contraexemplos) em caso de inconsistências encontradas. O estudo exploratório envolveu a conversão das regras definidas na KiPO para a linguagem *Alloy* e aplicação destas regras nos 4 cenários diferentes.

Cada cenário trata de um domínio diferente com características diferentes em seus modelos. Conjuntos de instâncias de cada um dos cenários foram codificadas para linguagem *Alloy* e verificadas de acordo com as regras da KiPO (já convertidas). Com automatização deste processo de validação, foi possível atender a todos os cenários, avaliar de maneira precisa e rápida, além de apresentar soluções de fácil compreensão.

O cenário 1 apresenta um modelo de análise de incidentes, que havia sido previamente modelado de acordo com a KiPO. O objetivo neste cenário foi avaliar se a aplicação é capaz de descobrir as inconsistências e apresentar as novas relações que poderiam acontecer, através de contraexemplos.

O cenário 2 apresenta um modelo de processo de elaboração de dissertação de mestrado, que havia sido previamente modelado de acordo com as regras da KiPO. O objetivo neste cenário foi avaliar se a aplicação é capaz de descobrir erros.

Os cenários 3 e 4 fazem parte de um mesmo domínio, porém com contextos diferentes. O cenário 3 parte diretamente de instâncias de um processo, disponíveis através de um log de eventos do Centro de Investigação e Prevenção de Acidentes Aeronáuticos. Não existe um modelo de processos definido, ou seja, não são previamente conhecidas as relações entre os elementos deste processo. O objetivo neste cenário foi identificar os elementos da KiPO e posteriormente, através da combinação KiPO-Alloy, simular modelos.

O cenário 4 foi um estudo dos modelos simulados no cenário 3, e determinado um modelo ideal para este processo, ou seja, um modelo livre de erros de consistência. A partir deste modelo, foi feita a validação, através da combinação KiPO-Alloy. O objetivo neste cenário foi confirmar que a aplicação valida corretamente um modelo sem erros.

### 5.1. Cenário 1: Apresentação de Contraexemplos de Modelos

O objetivo deste cenário de avaliação foi simular de forma controlada um problema na modelagem de um KiP, de forma a avaliar a geração de contra-exemplos deste modelo pelo *Alloy Analyzer*.

O domínio do processo considerado neste cenário foi a Análise de Incidentes segundo a especificação do padrão ITIL [Magalhães e Pinheiro 2007], em uma empresa real prestadora de serviços de infraestrutura de Tecnologia da Informação, sediada na cidade do Rio de Janeiro. As informações extraídas deste domínio foram mensagens trocadas internamente (entre os funcionários da própria empresa) e externamente (entre funcionários da empresa e os clientes envolvidos) com o objetivo de solucionar problemas nos seus clientes.

Para a avaliação deste cenário, foram elaboradas a asserção e checagem mostradas na Figura 4. Pode-se observar que, na checagem da asserção, foram atribuídas duas instâncias para Knowledge Intensive Activity e uma para Activity Goal. Porém, a cardinalidade deste relacionamento, de acordo com a Figura 2, é  $(1 .. 0,1)$ , ou seja, para cada Atividade Intensiva em Conhecimento pode haver somente um ou nenhum Objetivo de Atividade.

Não foi especificado na asserção à qual Atividade Intensiva em Conhecimento o Objetivo deveria estar relacionado. A partir desta regra, o *Alloy Analyzer* gerou um contraexemplo (Figura 5) que mostra uma das Atividades Intensivas em Conhecimento (*KnowledgeIntensiveActivity1*) que não possui relacionamento com Objetivo da Atividade (*ActivityGoal*), evidenciando a necessidade de inclui-se uma asserção adicional que prevenisse esta instanciação.

```

assert analiseDeIncidente {
  all Cliente:Agent,
    SolicitarCriacaoDeCorreioEletronico:KnowledgeIntensiveActivity ,
    ResolverProblema:Intention,
    Funcionario: InnovationAgent,
    ProblemaResolvido:ActivityGoal,
    Gmail:ImpactAgent,
    AnalistaDeTI:Specialty,
    EnviarEmail:Desire,
    VaiSerResolvidoRapidamente:Belief,
    AdicionarNovasTecnologias:Innovation |
  ResolverProblema in VaiSerResolvidoRapidamente.becomes &&
  Cliente in VaiSerResolvidoRapidamente.belongsTo &&
  Cliente in EnviarEmail.motivates &&
  AnalistaDeTI in Funcionario.hasSpecialty &&
  ResolverProblema in Cliente .undertakesToCarryOut &&
  ResolverProblema in SolicitarCriacaoDeCorreioEletronico.hasIntention &&
  ProblemaResolvido in SolicitarCriacaoDeCorreioEletronico.hasGoal &&
  Funcionario in Cliente.especializacao_innovation &&
  Gmail in Cliente.especializacao_impact &&
  AdicionarNovasTecnologias in Funcionario.innovated &&
  AdicionarNovasTecnologias in SolicitarCriacaoDeCorreioEletronico.hasInnovation }
check analiseDeIncidente for 1 Agent, 2 KnowledgeIntensiveActivity,
1 Innovation, 1 Desire,
1 Intention, 1 ActivityGoal, 1 InnovationAgent,
1 ImpactAgent, 1 Specialty,
1 Belief

```

Figura 4: Asserção e Checagem do processo Análise de Incidentes.

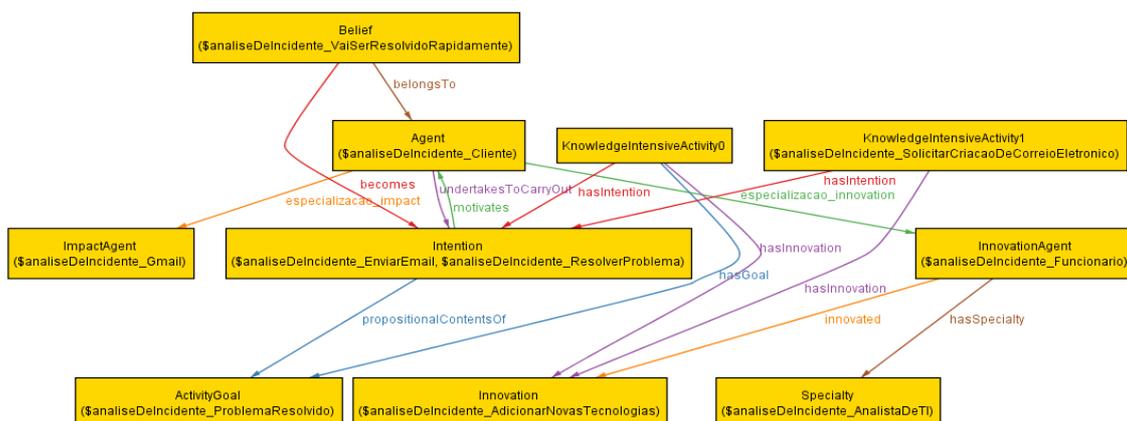


Figura 5: Contraexemplo do Alloy Analyzer, apresentando o erro de atribuição relacional entre Knowledge Intensive Activity e Activity Goal.

A atividade intensiva em conhecimento “SolicitarCriaçãoDeCorreioEletrônico” não tem nenhum Objetivo explícito relacionado, o que pode ser configurado como um “erro” de modelagem. Na Figura 5, observa-se também que a Intenção “Resolver Problema” associada ao Agente “Cliente” está presente no modelo e se relaciona com o Objetivo de Atividade “Problema Resolvido”. Desta forma, é possível assumir que existe conhecimento do domínio do modelo que apenas não foi formalizado. A análise automática parece ter sido útil para este apontamento.

## 5.2. Cenário 2: Erro no modelo de domínio

Neste cenário é apresentado o modelo do processo intensivo em conhecimento “Elaboração de Dissertação de Mestrado” (França et al. 2015), o qual foi modelado de acordo com as regras da KiPO e testado através de uma asserção no Alloy com o objetivo de descobrir erros.

O domínio “Elaboração de Dissertação de Mestrado” (França et al. 2015) é um processo executado no contexto de um Programa de Pós-graduação de uma Universidade pública. Neste domínio, foram modeladas todas as atividades, intenções, crenças, ou seja, grande parte dos elementos que compõem o processo para elaborar uma dissertação.

Com as regras da KiPO descritas na Figura 3 (já implementadas em *Alloy*), foi elaborada uma asserção com a inclusão das instâncias deste KiP para realizar a checagem do modelo de domínio (Figura 6).

```
assert elaboracaoDissertacao {
  all FazerDissertacao:Experience, Aluno: ImpactAgent, Professor:ImpactAgent,
  PesquisarTrabalhosRelacionados:KnowledgeIntensiveActivity,
  DefinirProblemaDePesquisa:KnowledgeIntensiveActivity,
  PesquisaComTemasSemelhantes:Intention,
  AlunosDeMestrado: InnovationAgent,
  DissertacaoDeMestradoConcluida:ActivityGoal,
  VisoesSobreOProblemaDePesquisa:Innovation,
  Doutor:Specialty, EscolhaDaMelhorArea:Desire,
  AlunoAcreditaqueSeuProblemaTeraRelevancia:Belief |
  FazerDissertacao in Aluno.belongsToExperience &&
  Aluno in EscolhaDaMelhorArea.motivates &&
  Aluno in AlunoAcreditaqueSeuProblemaTeraRelevancia.belongsToBelief &&
  PesquisaComTemasSemelhantes
  in AlunoAcreditaqueSeuProblemaTeraRelevancia.becomes &&
  Aluno in PesquisarTrabalhosRelacionados.hasIntention &&
  Professor in DefinirProblemaDePesquisa.hasIntention &&
  Doutor in Professor.hasSpecialty &&
  VisoesSobreOProblemaDePesquisa in AlunosDeMestrado.innovated &&
  DissertacaoDeMestradoConcluida in
  PesquisarTrabalhosRelacionados.hasGoal &&
  DissertacaoDeMestradoConcluida in DefinirProblemaDePesquisa.hasGoal }
```

**Figura 6: Asserção do processo Elaboração de Dissertação de Mestrado.**

De acordo com a Figura 2, cada agente precisa estar relacionado a pelo menos uma Intenção para realizar uma Atividade Intensiva em Conhecimento. Porém, a asserção acima apresenta os agentes “Aluno” e “Professor” relacionados às atividades “PesquisarTrabalhosRelacionados” e “DefinirProblemaDePesquisa”, respectivamente, sem ter uma Intenção relacionada entre eles. Logo, a ferramenta *Alloy* apresenta um aviso (Warning) alertando o erro existente e qual o relacionamento correto que deve ser feito (Figura 7).

```
Starting the solver...
Warning #1
Subset operator is redundant, because the left and right
subexpressions are always disjoint.
Left type = {this/ImpactAgent}
Right type = {this/Intention}

Warning #2
Subset operator is redundant, because the left and right
subexpressions are always disjoint.
Left type = {this/ImpactAgent}
Right type = {this/Intention}

Note: There were 2 compilation warnings. Please scroll up to see them.
Warnings often indicate errors in the model.
Some warnings can affect the soundness of the analysis.
To proceed despite the warnings, go to the Options menu.
```

**Figura 7: Aviso sobre a Asserção do processo Elaboração de Dissertação de Mestrado.**

Neste caso, não há evidências de que exista conhecimento suficiente sobre o domínio para justificar um equívoco do modelador. Eventualmente, seria necessário coletar estas informações com representantes dos participantes do processo (Agentes) e incluir estes elementos na modelagem. Como o elemento Intenção (Intention) é fundamental em um cenário de KiP, mas não é presente em metamodelos de processos tradicionais, é importante que uma análise deste tipo deixe claro o problema com o modelo elaborado.

### 5.3. Cenário 3: Simulação de Modelos

Este cenário foi proposto para identificar os elementos da KiPO e, através da combinação KiPO-Alloy, simular modelos. O domínio escolhido diz respeito ao processo de decolagem de aeronaves, cujos dados foram extraídos de uma base de ocorrências aeronáuticas, gerenciada pelo Centro de Investigação e Prevenção de Acidentes Aeronáuticos (CENIPA). Constam nesta base de dados os eventos aeronáuticos notificados ao CENIPA nos últimos 10 anos e que ocorreram em solo brasileiro. Os dados extraídos foram: Ocorrências, Aeronaves envolvidas, Fatores Contribuintes das Ocorrências, Recomendações de Segurança.

Como neste cenário não havia um modelo de processo já elaborado, inicialmente foi feita uma identificação dos elementos da KiPCO e codificados junto à combinação KiPO-Alloy, para posteriormente serem validadas. Portanto, para a avaliação deste cenário, foi feita uma classificação dos elementos do modelo de domínio e atribuição de cada elemento, como instância, a sua entidade correspondente da KiPCO. A aplicação gerou posteriormente um predicado para executar as instâncias criadas. A partir desta estrutura, o *Alloy Analyzer* foi executado e gerou o diagrama da Figura 8. O usuário pode acessar as instâncias descobertas através do link “*Instance Found*” através da interface do *Alloy Analyzer*, o qual apresentará a imagem a Figura 9. Cada instância que estava vinculada no código aparece também vinculada ao elemento da KiPCO.

A partir do diagrama de instâncias apresentado na Figura 9, foi possível observar todas as possíveis relações e o modelo do PIC construído. Desta forma, foi possível também entender como o ambiente está organizado, como os elementos estão relacionados, e se este modelo corresponde à realidade da organização, sendo particularmente útil em cenários de análise de conformidade, por exemplo.

```

one sig Piloto in Agent{}
one sig CoPiloto in Agent{}
one sig PrevenirAcidente in Intention{}
one sig PousoTranquilo in Desire {}
one sig FazerPousoForcado in KnowledgeIntensiveActivity{}
one sig PousarCorretamente in ActivityGoal{}

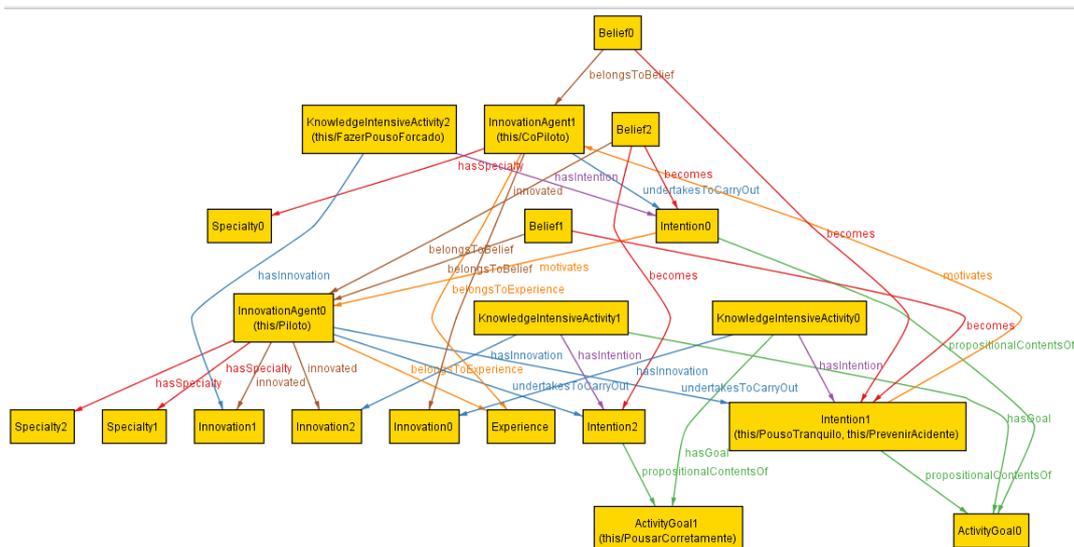
pred exemplo{}

run exemplo

Executing "Run exemplo"
  Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20
  1698 vars. 150 primary vars. 2662 clauses. 104ms.
  Instance Found. . Predicate is consistent. 256ms.

```

**Figura 8: Predicado e Execução de Instâncias das Ocorrências da Aeronáuticas.**



**Figura 9: Instâncias geradas pela ferramenta Alloy Analyzer.**

O contexto deste caso é diferente dos demais, pois algumas instâncias do processo foram coletadas para apoiar a identificação dos elementos do modelo. A Atividade Intensiva em Conhecimento “RealizarPousoForçado”, Intenções e Objetivos estão presentes no modelo deste processo. Porém, realizar uma análise visual da Figura 9 não é uma tarefa fácil para evidenciar possíveis erros de modelagem. O estudo no cenário 4 apoiou a análise destes problemas.

#### 5.4. Cenário 4: Modelo sem Contraexemplo

Neste cenário foi feito um estudo no modelo simulado no cenário 3 e elaborado um modelo ideal para o KiP, ou seja, sem erros. Com este modelo, foi feita uma validação com as regras da KiPO implementadas no *Alloy*, onde foi apresentado um modelo sem erros.

O domínio escolhido para elaboração deste cenário foi também o mesmo do cenário 3, a base de conhecimento do CENIPA. Para avaliar este cenário, foi criada uma asserção com checagem, estendendo os elementos da KiPCO, conforme a Figura 10.

```

assert trafegoAereo{
  all Piloto:ImpactAgent,
  CoPiloto:ImpactAgent,
  PrevenirAcidente:Intention,
  PousoTranquilo:Desire,
  FazerPousoForçado:KnowledgeIntensiveActivity,
  PousarCorretamente: ActivityGoal
|
  PrevenirAcidente in Piloto.undertakesToCarryOut &&
  PrevenirAcidente in CoPiloto.undertakesToCarryOut &&
  PrevenirAcidente in FazerPousoForçado.hasIntention &&
  PousoTranquilo in PrevenirAcidente &&
  PousarCorretamente in FazerPousoForçado.hasGoal}
check trafegoAereo for 1 Agent, 1 KnowledgeIntensiveActivity,
1 Innovation, 1 Desire,
1 Intention, 1 ActivityGoal, 1 InnovationAgent,
1 ImpactAgent, 1 Specialty,
1 Belief, 1 Experience

```

**Figura 10: Asserção e Checagem do processo Decolagem de Aeronaves.**

Porém, como todos os relacionamentos estavam de acordo com as regras ditadas pela KiPCO, o modelo não apresentou contra-exemplos ou erros, conforme informa a Figura 11.

```
Executing "Check validacao for 1 Agent, 1 KnowledgeIntensiveActyi
1 Innovation, 1 Desire, 1 Intention, 1 ActivityGoal, 1 InnovationAgent, 1 ImpactAgent,
1 Specialty, 1 Belief, 1 Experience"
Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20
161 vars. 28 primary vars. 239 clauses. 56ms.
No counterexample found. Assertion may be valid. 73ms.
```

**Figura 11: Resultado da checagem do processo Decolagem de Aeronaves.**

Os cenários 3 e 4 em conjunto evidenciaram grande parte do problema tratado neste artigo. Um modelo de processo preliminar, não consolidado é difícil de ser analisado, ainda mais em um domínio complexo e tratando da representação de elementos não triviais. Os resultados obtidos com a análise automática fornecem apoio para o modelador para garantir a consistência deste modelo.

## 6. A Ferramenta KIPAlloy

Uma das dificuldades encontradas na elaboração dos cenários apresentados na Seção 5 foi a implementação das regras da KiPO na linguagem *Alloy*. Por ser uma linguagem textual, com sintaxe específica e de difícil compreensão, a elaboração de regras para checagem de modelos se torna muito complexa. A dificuldade aumenta quando se constróem modelos de processos intensivos em conhecimento, pois o modelador deve compreender a ontologia KiPO e saber como implementar seu cenário usando a linguagem *Alloy* com a KiPO. Outra dificuldade foi a elaboração dos nomes (rótulos) dos elementos do modelo que se relacionavam com os conceitos da KiPO. Para cada cenário implementado no Alloy, foi necessário criar rótulos e associá-los a conceitos da KiPO. Porém, como o Alloy não possui a funcionalidade de validação de rótulos, isso torna o modelo ainda mais complexo em sua leitura.

Para solucionar os problemas identificados na construção dos cenários relatados Seção 5, foi desenvolvida uma ferramenta, chamada *KIPAlloy*, que auxilia o usuário na construção de modelos de KiP, com uso da KiPO, na linguagem *Alloy*, além de também auxiliá-lo na elaboração dos respectivos rótulos dos elementos destes modelos. Ela é uma extensão do *Alloy* para modelar processos intensivos em conhecimento, usando a KiPO.

A ferramenta *KIPAlloy* foi desenvolvida usando a plataforma Java 8, banco de dados MySQL, linguagem *Alloy* e utiliza a ferramenta de checagem de modelos *Alloy Analyzer*. A cada inclusão de elementos feita pelo usuário, a ferramenta valida o rótulo de acordo com uma estrutura linguística proposta para cada um dos respectivos conceitos da KiPO que ele associou. Depois que todos os elementos são incluídos no banco de dados MySQL, a ferramenta gera um arquivo com regras da KiPO associadas aos elementos do KiP, implementadas em linguagem *Alloy* e abre a ferramenta *Alloy Analyzer* para realizar as validações de modelo.

Para a proposta de definição dos rótulos, foi realizado um estudo na literatura sobre estruturas linguísticas, com o objetivo de criar um padrão para nomear os elementos de KiP. Para cada conceito, foi feita uma pesquisa sobre teorias linguísticas semânticas e sintáticas ou a existência de padrões de especificação textual, com o objetivo de identificar qual a melhor forma de descrevê-lo. Os padrões foram analisados de forma

muito criteriosa, pois as estruturas linguísticas elaboradas poderiam causar dificuldade na compreensão do modelo. Na Tabela 3, são apresentados exemplos de padrões de estruturas linguísticas de alguns conceitos da KiPO.

**Tabela 3 – Padrões de Estruturas Linguísticas de alguns conceitos da KiPO.**

Conceito	Nível	Estrutura Linguística	Exemplo
KIPCO::ActivityGoal	Modelo	<verbo no infinitivo> + <substantivo>	<i>Solve the printer failure</i>
KIPCO::ActivityGoal	Instância	<Rótulo do elemento no nível de Modelo de ActivityGoal>+<rótulo da instância da KIA associada, com verbo substantivado>	<i>Solve the printer failure by Diagnosing problem occurred on April 14,2018 by John</i>
BPO::Activity	Modelo	<verbo no infinitivo>+<objeto direto objeto indireto>	<i>Publish final results of dissertation</i>
BPO::Activity	Instância	<verbo no infinitivo>+<objeto direto objeto indireto>+<"occurred at"><timestamp>+<"in">+<place>+<"by">+<nome do Agent executor da atividade>	<i>Publish final results of dissertation occurred at April 14, 2018, in SBSI 2018, by John.</i>
KIPCO::ReactionRule	Modelo	<If>+<restrição>+<then>+<operador modal>+<evento>	<i>If the devolution does not happen in rental branch, then it is obligatory to pay fine.</i>

No *KIPAlloy*, a validação dos rótulos é feita com o uso de processamento de linguagem natural (NLP), pelo Stanford NLP [Song e Chambers 2014]. Inicialmente, é realizada a uma tokenização a nível de palavras, transformando cada mensagem em um conjunto de sentenças, contendo um conjunto de tokens (palavras, pontuações, etc.). Posteriormente, é feita classificação morfológica (Pos-tagging) de cada token (verbo, pronome, etc). Os módulos do Stanford usados foram POSTaggerME, POSModelLoader, POSModel (Figura 12).

```
File file = new File(".\\lib\\en-pos-maxent.bin");
String caminho = file.getAbsolutePath();
int ponto = caminho.indexOf(".");
caminho = caminho.substring(0, ponto) + "\\lib\\en-pos-maxent.bin";
file = new File(caminho);
POSModel model = new POSModelLoader().load(new File(file.getAbsolutePath()));
POSTaggerME tagger = new POSTaggerME(model);
String tokens[] = WhitespaceTokenizer.INSTANCE.tokenize(mensagem);
String[] tags = tagger.tag(tokens);
```

**Figura 12 – Tokenização e POS-Tagging dos Rótulos.**

Os rótulos podem ser escritos em inglês ou português (ver um exemplo na Figura 15). Logo, para cada um dos idiomas existe uma biblioteca específica para realizar a tokenização e *POS-Tagging* dos rótulos: **en-pos-maxent.bin (inglês)** e **pt-pos-maxent.bin (português)**. Cada padrão de estrutura linguística de cada conceito da KiPO (tanto em nível de modelo quanto de instâncias) foi implementada em Java, usando a

classificação morfológica de cada expressão da sentença do rótulo, de acordo também com o idioma do rótulo (Figura 13).

<pre> public Boolean achouVerbo(String[] tokens, String[] tags, String idioma){     Boolean achou =false;     if(idioma.toUpperCase().equals("INGLÊS")){         while(achou==false &amp;&amp; tags.length &gt; i){             if (tags[i].equals("NN")    tags[i].equals("NNS")    tags[i].equals("VB")                    tags[i].equals("VBG") tags[i].equals("VBD")    tags[i].equals("VBZ")                    tags[i].equals("VBN")) {                     achou = true;                     break;                 }                 i++; }         }         if(idioma.toUpperCase().equals("PORTUGUÊS")){             while(achou==false &amp;&amp; tags.length &gt; i){                 if (tags[i].contains("V-")) {                     achou = true;                     break;                 }                 i++; }         }         return achou;     } } </pre>	<pre> public String padraoBPOActivityModelo(String[] tokens, String[] tags, String msg, String idioma) {     String resultado="";     Boolean achouVerbo=false,     achouSubstantivo=false;     If (achouVerbo(tokens,tags, idioma) == true &amp;&amp;     achouSubstantivo(tokens,tags,idioma) ==true){         resultado += ("\n " + msg + " (Activity1stOT): Rótulo do elemento no nível de Modelo - CORRETO.");     } else {         resultado += ("\n " + msg + " (Activity1stOT): Rótulo do elemento no nível de Modelo - INCORRETO.");         resultado += ("Sugestões:&lt;verbo no infinitivo&gt;+&lt;objeto direto objeto indireto&gt;.É recomendável usar verbos de ação. ");     }     return resultado; } </pre>
---	---

**Figura 13 – Padrão linguístico para BPO::Activity a nível de modelo.**

Toda a codificação é executada a partir do momento que o usuário inclui algum rótulo na ferramenta.

Na ferramenta *KIPAlloy* o usuário pode informar os elementos do cenário que deseja modelar, além de relacioná-lo a um conceito da KiPO. Ele deve informar o título do processo intensivo em conhecimento, o conceito da KiPO que deseja especializar o elemento que deseja inserir e o rótulo deste elemento. Ao selecionar o conceito da KiPO, é apresentada a definição do respectivo conceito na área de texto ao lado, automaticamente. Todos os elementos que o usuário incluir no modelo, devem ser adicionados em uma tabela da própria tela através do botão “Adicionar Item”. É possível excluir o item da tabela ao acionar o botão “Excluir item” (Figura 14). Para exemplificar a usabilidade da ferramenta *KIPAlloy*, foi usada a base de ocorrências aeronáuticas, gerenciada pelo Centro de Investigação e Prevenção de Acidentes Aeronáuticos (CENIPA) apresentado na Seção 5.

A cada inclusão de elemento na tabela, o rótulo é validado de acordo com os padrões de estruturas linguísticas adicionados ao sistema. É verificado também o conceito da KiPO a que o rótulo está vinculado, e se o mesmo está de acordo com o padrão linguístico do respectivo conceito. Se o rótulo não estiver correto, o item será incluído na tabela, porém o plano de fundo da linha fica “vermelho” como alerta para o usuário (Figura 15).

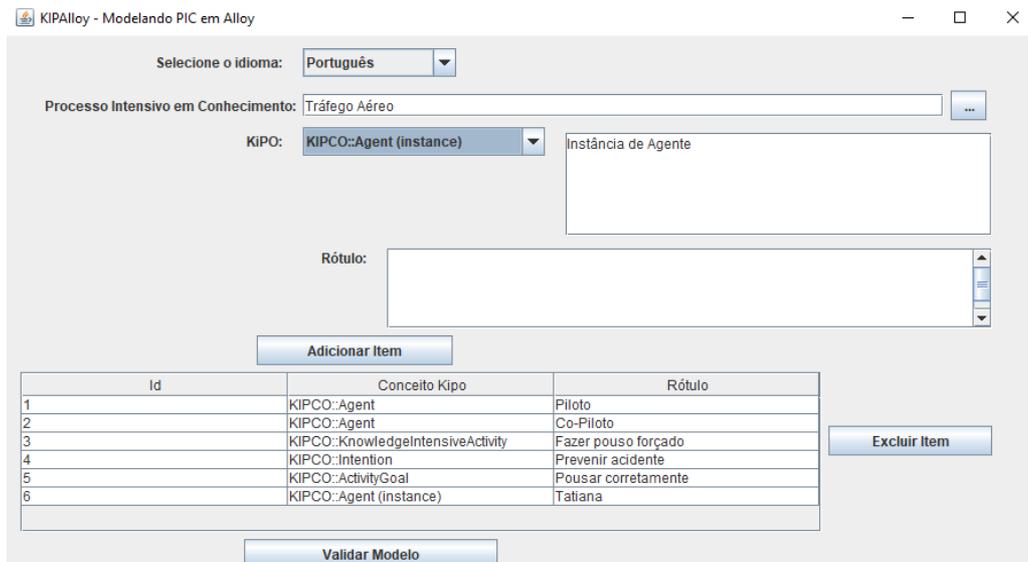


Figura 14 – Tela de Inclusão de Elementos de do KiP modelado.

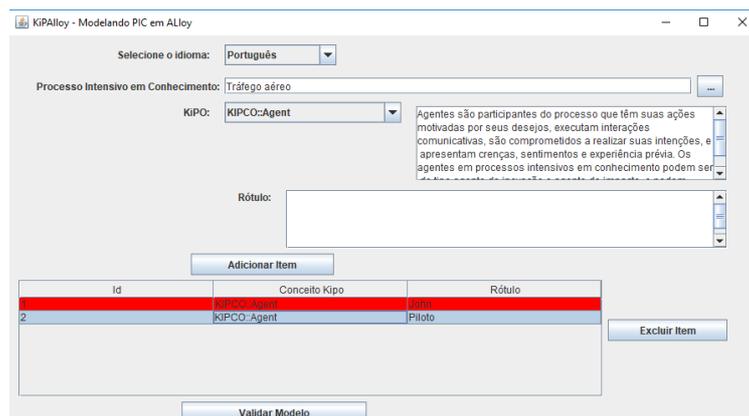


Figura 15 – Linha em vermelho mostrando o rótulo que não está de acordo com o padrão de estrutura linguística.

Ao pressionar o botão “Validar Modelo”, inicialmente os dados são armazenados na tabela *ElementoPic* da base de dados MySQL, e então é gerado um arquivo contendo regras em *Alloy* do modelo do KiP, usando os conceitos da KiPO, os quais foram selecionados pelo usuário. Neste arquivo, estes conceitos – com suas respectivas relações – são convertidas em assinaturas (SIG). Para os elementos do modelo do KiP, foi criada uma asserção, onde cada elemento é associado a seu respectivo conceito da KiPO, definido pelo usuário (Figura 16).

Este arquivo, posteriormente, é transferido para a aplicação *Alloy Analyzer*, com o objetivo de poder simular o modelo de acordo com os elementos definidos pelo usuário. Primeiramente, os elementos são validados através das regras da *Alloy*. Caso exista alguma irregularidade, o *Alloy Analyzer* mostrará o que está descrito na regra e o que o usuário adicionou na ferramenta *KIPAlloy*. Caso as informações incluídas pelo usuário estejam corretas, o *Alloy Analyzer* realizará a simulação do modelo, como apresentado na Figura 17.

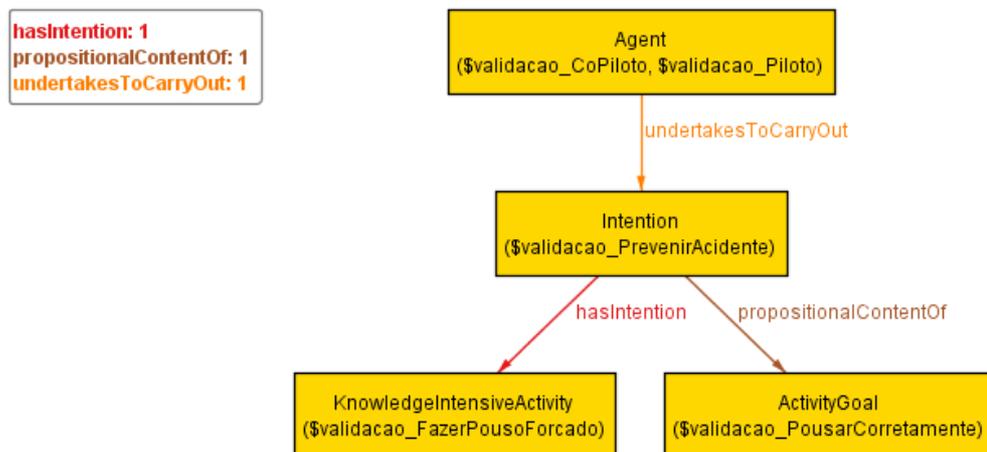
```

some sig Agent {undertakesToCarryOut: some Intention}
some sig ImpactAgent{especializaImpactAgent : some Agent}
some sig Intention {propositionalContentOf: some ActivityGoal,
hasIntention: some KnowledgeIntensiveActivity}
some sig KnowledgeIntensiveActivity{ hasGoal : lone ActivityGoal}
some sig ActivityGoal{}
assert validacao {
all Piloto:Agent, CoPiloto:Agent, PrevenirAcidente:Intention,
FazerPousoForcado:KnowledgeIntensiveActivity,
PousarCorretamente: ActivityGoal |
PrevenirAcidente in Piloto.undertakesToCarryOut
&& PrevenirAcidente in CoPiloto.undertakesToCarryOut &&
FazerPousoForcado in PrevenirAcidente.hasIntention &&
PousarCorretamente in PrevenirAcidente.propositionalContentOf &&
PousarCorretamente in FazerPousoForcado.hasGoal}
check validacao for 1 ImpactAgent, 1 Agent,
1 Intention, 1 ActivityGoal, 1 KnowledgeIntensiveActivity

```

**Figura 16 – Arquivo gerado pelo KIPAlloy a partir de informações incluídas pelo usuário.**

A Figura 17 mostra o processo do cenário 4 do estudo exploratório baseado nas ocorrências aeronáuticas, gerenciada pelo Centro de Investigação e Prevenção de Acidentes Aeronáuticos (CENIPA). Os conceitos e elementos utilizados para esta representação foram: KIPCO::Agent (CoPiloto, Piloto), KIPCO::Intention(PrevenirAcidente), KIPCO::KnowledgeIntensiveActivity (FazerPousoForçado) e KIPCO::ActivityGoal(Pousar Corretamente). Diferente do que foi apresentado na descrição anterior do cenário 4, este modelo ilustra somente os elementos e respectivos conceitos que o usuário deseja incluir, podendo gerar ou não contraexemplos.



**Figura 17 –Modelo gerado pelo AlloyAnalyzer a partir do arquivo gerado.**

## 7. Análise dos Resultados e Discussão

O uso do *Alloy* para modelar Processos Intensivos em Conhecimento, com base nas regras da KiPO, apresentou mecanismos que, automaticamente, foram capazes de validar e simular os modelos de domínio nos cenários do estudo de caso executado. O uso do *Alloy Analyzer* evidenciou que é possível que o modelador consiga validar ou simular os modelos de processo de sua organização de maneira automática e exata. A simulação de

KiPs gera conjuntos de instâncias dos elementos do processo, que caracterizam mundos possíveis do KiP sendo modelado. Tais instâncias podem ser previamente desconhecidas pelo modelador, e representam um corpo de conhecimento que é externalizado pelos mecanismos de transformação de regras e simulação providos pela ferramenta proposta, conforme preconizado por Nonaka e Takeuchi (1995).

O estudo exploratório teve caráter exploratório, portanto, com os resultados obtidos não é possível afirmar que qualquer modelo de processo intensivo em conhecimento pode ser validado a partir do *Alloy*. No entanto, as evidências dos cenários apresentados mostram resultados promissores. Foi possível observar que a ferramenta encontra inconsistências e erros, bem como consegue avaliar positivamente modelos corretos.

Uma limitação encontrada é que não existe um mecanismo visual para adicionar as instâncias e regras no *Alloy Analyzer*. A conversão para a linguagem *Alloy* é feita de maneira manual. O modelador que não tem conhecimento da linguagem pode ter certa dificuldade em representar suas instâncias e construir regras. Seria interessante criar um conversor automático, onde o usuário somente adicionasse as informações, sem a necessidade de codificá-las.

Com a proposta e implementação da ferramenta *KIPAlloy* (extensão do *Alloy* para modelar KiP usando KiPO), o modelador de processos intensivos em conhecimento consegue elaborar seus modelos, validando-os nas regras da KiPO implementadas no *Alloy*. Além disso, elabora rótulos de elementos a nível de instância e modelo, a partir de validações de padrão estruturas linguísticas de cada conceito da KiPO que associa. Comparado com ferramentas de modelagem, como DCRGraphs (Hildebrandt e Mukkamala, 2010), Declare (Westgaard e Maggi, 2011) e DPIL Navigator 2.0 (Schoning, 2001), *KIPAlloy* consegue, além de checar se os dados informados pelo modelador estão corretos de acordo com as regras e restrições da KiPO, valida os rótulos de acordo com cada conceito da KiPO relacionado, tanto a nível de instância quanto de modelo. Portanto, argumentamos que as funcionalidades disponibilizadas pela ferramenta *KiPALloy* (de suporte à modelagem e de integração direta com o mecanismo de simulação de processos do *Alloy Analyzer*) foram capazes de atender à questão estabelecida para esta pesquisa (“Quais são os resultados possíveis de uma validação automática de modelos de processos intensivos em conhecimento?”), uma vez que permitem ao modelador de KiP visualizar e avaliar instâncias de mundos possíveis resultantes da simulação do KiP modelado.

A partir dos resultados do estudo exploratório, novas questões de pesquisa foram levantadas. A primeira questão está relacionada com o modelo de referência para a realização das análises. Em cenários complexos, qual é a relevância da existência prévia de um modelo? No cenário 3, foi necessário criar os elementos do modelo com base nos dados disponíveis. Uma possível sugestão para este problema seria o uso de técnicas de mineração de processos para descoberta do modelo. A segunda questão diz respeito às tomadas de decisões sobre problemas encontrados através das análises. No cenário 1, foi observado um contraexemplo, que precisa ser tratado. A questão que se coloca é como analisar os erros para classificá-los e aplicar uma solução? A ferramenta pode oferecer suporte?

Finalmente, a terceira questão é sobre a variabilidade de KiP, ou seja, as instâncias podem assumir caminhos e formas diferentes umas das outras. Neste caso, pode-se

inclusive considerar a violação de regras de negócio em alguns contextos. As regras de modelagem estão sujeitas a estas mesmas variações? Uma pesquisa mais aprofundada sobre regras de negócios em KiP seria necessária.

## 8. Conclusões

A modelagem de processos de negócio já está estabelecida como prática bastante efetiva para a gestão de uma organização, embora até recentemente tenha dado enfoque prioritário a processos estruturados. No entanto, têm-se observado o surgimento e crescente relevância nas organizações de cenários pouco estruturados, com alto grau de complexidade, dinamismo e inovação, forte colaboração e interação entre os agentes, e tomadas de decisão envolvendo conhecimento tácito, que vêm sendo denominados processos intensivos em conhecimento (KiP).

A modelagem de processos intensivos em conhecimento apresenta uma série de desafios, e várias propostas de metodologias, linguagens e ferramentas têm surgido na literatura recente para apoio computacional à modelagem de KiPs. Em particular, este trabalho propõe a ferramenta *KiPAlloy*, que dá suporte à **validação automática de modelos de processos intensivos em conhecimento**, incluindo o suporte à modelagem e a integração direta com o mecanismo de simulação de processos do *Alloy Analyzer*.

A linguagem *Alloy* analisa as regras de um modelo e descobre estruturas que as satisfazem. Pode ser usado tanto para explorar o modelo gerando simples estruturas, quanto checar propriedades do modelo gerando contraexemplos. As estruturas são apresentadas graficamente e suas apresentações podem ser customizadas para melhor representar o domínio. Os resultados deste trabalho confirmaram a aplicabilidade da ferramenta e linguagem para o contexto de processos intensivos em conhecimento, mais especificamente a ontologia KiPO.

Para checar modelos de processos intensivos em conhecimento, feitos com o uso da KiPO e com validação de estruturas linguística sobre os rótulos dos elementos destes modelos, além de converter regras KiPO para linguagem *Alloy*, foi desenvolvida a ferramenta *KIPAlloy*. Nesta ferramenta, após a inclusão dos elementos do domínio associados aos conceitos da KiPO, é gerado um arquivo com toda esta inclusão convertida em linguagem *Alloy*, sendo transferido à ferramenta *Alloy Analyzer* para checagem do modelo.

Como trabalhos futuros, serão elaboradas as regras das outras perspectivas da KiPO para que outros assuntos, que também compõem um KiP, sejam modelados automaticamente. Adicionalmente, pretende-se realizar uma extensão da aplicação *UML2Alloy* para KiP, onde o modelador construa seus modelos na UML e, através desta aplicação, já conseguiria validar seu modelo.

## Referências

- Banks, J., Carson, J. S., Nelson, B. L. and Nicol, D. (2010) *Discrete-Event System Simulation*, Prentice Hall, Upper Saddle River, NJ, 5<sup>th</sup> edition.
- Barboza, T., Baião, F. and Santoro, F. (2018) *Validação Automática de Modelos de Processos Intensivos em Conhecimento através de Alloy*. XIV Simpósio Brasileiro de Sistemas de Informação (SBSI 2018), Caxias do Sul.

- Benevides, A., Guizzardi, G., Braga, B. and Almeida, J.P. (2010) Validating Modal Aspects of OntoUML Conceptual Models Using Automatically Generated Visual World Structures. *Journal of Universal Computing Systems*, 16 (20), 2904-2933.
- Borbar, B., Anastasakis, K., Georg, G. and Ray, I. (2007) UML2Alloy: A Challenging Model Transformation, In: *International Conference on Model Driven Engineering Languages and Systems (MODELS 2007)*, 436-450.
- Bratman, M. (1987) *Intention, plans, and practical reason*, Harvard University Press.
- Di Ciccio, C., Marrela, A. and Russo, A. (2015) Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. *Journal on Data Semantics*, 4(1), 29-57.
- Dumas, M., Rosa, M., Mendling, J. and Reijers, H. (2018) *Fundamentals of Business Process Management*, Springer, 2<sup>nd</sup> edition.
- França, J., Netto, J., Carvalho, J., Santoro, F., Baião, F. and Pimentel, M. (2015) KiPO: the knowledge-intensive process ontology. *Journal of Software and Systems Modeling (SoSyM)*, 14(3), 1127-1157.
- Guerson, J., Sales, T., Guizzardi, G. and Almeida, J.P. (2015) OntoUML Lightweight Editor: A Model-Based Environment to Build, Evaluate and Implement Reference Ontologies, *IEEE 19<sup>th</sup> International Enterprise Distributed Object Computing Workshop*, doi: 10.1109/EDOCW.2015.17.
- Guerson, J., Almeida, J.P. and Guizzardi, G. (2014) Support for Domain Constraints in the Validation of Ontologically Well-Founded Conceptual Models. In: *Enterprise, Business-Process and Information Systems Modeling*, p. 302-316.
- Guizzardi, G. (2005) Ontological Foundations for Structural Conceptual Models. In: *Universal Press, TheNetherlands*, ISBN 90-75176-81-3.
- Hildebrandt, T. and Mukkamala, R. (2010) Declarative Event-Based Workflow as Distributed Dynamic Condition Response Graphs. In: *Proceedings of PLACES*, doi: 10.4204/EPTCS.69.5.
- Jackson, D. (2006) *Software Abstraction – Logic, Language and Analysis*, In: MIT Press.
- Magalhães, I. and Pinheiro, W. (2007) *Gerenciamento de Serviços de TI na Prática*, Novatec.
- Maldonado, M. (2008) Impact analysis of knowledge intensive process creation and transfer policy: a system dynamic model, M.Sc. dissertation, Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, Universidade Federal de Santa Catarina, Brazil (in Portuguese).
- Milicevic, A., Near, J. P., Kang, E. and Jackson, D. (2015) Alloy\*: a general-purpose higher-order relational constraint solver, In: *Proceedings of the 37th International Conference on Software Engineering (ICSE 2015)*, Volume 1, pages 609-619.
- Mylopoulos, J. (1992) “Conceptual Modeling, Databases, and CASE: An Integrated View of Information Systems Development”, In: *Conceptual Modeling and Telos*, John Wiley & Sons, 49-68.

- Montaghani, V. and Rayside, D. (2017) Bordeaux: A Tool for Thinking Outside the Box, In: International Conference on Fundamental Approaches to Software Engineering (FASE 2017), 22-39.
- Nascimento, A., Santoro, F. and Cereja, J. (2015) Um Método Para Identificar Variabilidade em Processos Intensivos em Conhecimento. XI Simpósio Brasileiro de Sistemas de Informação (SBSI 2015), Goiânia.
- Nelson, T., Danas, N., Dougherty, D.J. and Krishnamurthi, S. (2017) The Power of Why and WhyNot: Enriching Scenario Exploration with Provenance, In: Proceedings of the 11<sup>th</sup> Joint Meeting on Foundations of Software Engineering (ESEC /FSE 2017).
- Netto, J., Santoro, F. and Baião, F. (2014) Evaluating KiPN for Modeling KiP. In: International Conference on Business Process Management (BPM 2013), Business Process Management Workshops, pp 549-561.
- Nonaka, I. and Takeuchi, H. (1995) The Knowledge-Creating Company, Oxford University Press.
- Nonaka, I., Uememoto, K. and Senoo, D. (1996) From information processing to knowledge creation: A paradigm shift in business management, *Technology in Society*, 18 (2), 203-218.
- Runeson, P., Host, M., Rainer, A. and Regnell, B. (2012) Case Study Research in Software Engineering: Guidelines and Examples, Wiley Publishing, 1<sup>st</sup> edition.
- Rao, A. and Georgeff, M. (1991) Modeling Rational Agents within a BDI Architecture. 2<sup>nd</sup> International Conference on Principles of Knowledge Representation and Reasoning (KR'91), Cambridge, Morgan Kaufmann Publishers, 473–484.
- Richter-von Hagen, C., Ratz, D. and Povaleg, R. (2005) Towards self-organizing knowledge intensive processes. *Journal of Universal Knowledge Management*, 2, 148-169.
- Schonig, S., Ackermann, L. and Jablonski, S. (2017) DPIL Navigator 2.0: Multi-Perspective Declarative Process Execution, In: 15<sup>th</sup> International Conference on Business Process Management (BPM 2017).
- Song, M. and Chambers, T. (2014) Text mining with the Stanford CoreNLP", *Measuring scholarly impact*, Springer.
- Walter, T., Parreiras, F. S., Staab, S. (2014) "An ontology-based framework for domain-specific modeling" *Software & Systems Modeling - Springer*
- Westergaard, M. and Maggi, F. (2011) Declare: A Tool Suite for Declarative Workflow Modeling and Enactment, In: 9<sup>th</sup> International Conference on Business Process Management (BPM 2011).
- Yin, R. K. (2015) Estudo de Caso – Planejamento e Métodos, Bookman, 5<sup>th</sup> edition.