

Regras para Apoiar a Especificação de Restrições em Processos-de-Processos de Negócio

Danilo dos Santos de Oliveira, Maria Istela Cagnin Machado (orientadora)

¹Faculdade de Computação (FACOM)

Universidade Federal de Mato Grosso do Sul (UFMS).

Av. Costa e Silva, s/n. - Bairro Universitário - CEP 79070-900 - Campo Grande - MS.

`danilo.s.oliveira@ufms.br, istela.machado@ufms.br`

Abstract. *Collaboration between organizations has become increasingly common, requiring the coordination of interorganizational processes. In this complex scenario, the M-PoP method emerges to properly model and, consequently, facilitate the understanding of Processes-of-Business Processes (PoP), which are complex and dynamic business processes composed of processes from different organizations that collaborate with one another. Although M-PoP defines specific views and models to support such modeling, significant challenges remain, particularly regarding the lack of standardization for specifying constraints, which hinders their precision, clarity, and comprehension. From this perspective, this undergraduate work aims to define a set of rules to assist business analysts in specifying constraints within the context of PoP, by contributing to the evolution of M-PoP and enhancing the representation of constraints in real-world PoP. To this end, a systematic literature mapping was conducted to identify the most suitable notation for representing such constraints, resulting in the selection of the Object Constraint Language (OCL). The rules were then systematically defined and applied to the Public Health PoP, illustrating their usefulness in a real scenario. The specification of the proposed rules were validated using the EclipseOCL and USE tools, demonstrating their consistency and practical applicability.*

Keywords: *interorganizational business process, constraint, OCL, documentation*

Resumo. *A colaboração entre organizações tem se tornado cada vez mais frequente, exigindo a coordenação de processos interorganizacionais. Nesse cenário complexo, surge o método M-PoP para modelar adequadamente e, conseqüentemente, facilitar o entendimento de Processos-de-Processos de Negócios (PoP) que correspondem a processos de negócio complexos e dinâmicos formados por processos de diferentes organizações que colaboram entre si. Embora M-PoP defina visões e modelos específicos para apoiar essa modelagem, ainda existem desafios significativos, especialmente no que diz respeito à ausência de padronização para especificar restrições, dificultando a sua precisão, clareza e compreensão. Sob essa perspectiva, este Trabalho de Conclusão de Curso (TCC) tem como objetivo definir um conjunto de regras para ajudar analistas de negócios a especificar restrições no contexto de PoP, contribuindo para a evolução do M-PoP e aprimoramento da representação de restrições em PoP reais. Para isso, foi conduzido um mapeamento sistemático da literatura a fim de selecionar a notação mais adequada para representar tais restrições, resultando na escolha da Object Constraint Language (OCL). Em seguida, as regras foram definidas sistematicamente e aplicadas ao PoP de Saúde Pública, ilustrando sua utilidade em um cenário real. A especificação das regras propostas foi validada por meio das ferramentas EclipseOCL e USE, demonstrando sua consistência e aplicabilidade prática.*

Palavras-chave: *processo de negócio interorganizacional, restrição, OCL, documentação*

1. Introdução

O cenário organizacional contemporâneo é marcado por mercados globalizados e altamente competitivos, nos quais empresas buscam expandir sua atuação por meio de parcerias estratégicas, e

processos de fusão e aquisição, formando alianças de organizações [Kumar and Sharma 2019]. Esses movimentos colaborativos exigem a integração de processos de negócio provenientes de diferentes organizações, muitas vezes apoiados por sistemas de software distintos e heterogêneos [Cagnin and Nakagawa 2021, Grefen 2024]. Entretanto, essa integração é complexa pois cada organização possui seus próprios fluxos de trabalho, regras operacionais e formas de estruturar atividades e missões. Além disso, tais processos são intrinsecamente dinâmicos e precisam se adaptar para atender novas legislações, estratégias ou comportamentos inesperados [Cagnin and Nakagawa 2022]. Assim, compreender, coordenar e adaptar processos interorganizacionais torna-se um desafio crítico para garantir a eficácia e a continuidade das alianças.

Nesse contexto, os Processos-de-Processos de negócios (PoP) surgem como uma abordagem para lidar com cenários interorganizacionais complexos, nos quais múltiplos processos (denominados processos constituintes), pertencentes a diferentes organizações, precisam ser coordenados para alcançar objetivos compartilhados [Cagnin and Nakagawa 2021]. A importância do PoP está na capacidade de oferecer uma visão mais ampla e integrada de como processos individuais colaboram com outros, dependem um dos outros e se organizam para produzir resultados em ambientes colaborativos [Cagnin and Nakagawa 2021]. No entanto, essa colaboração interorganizacional traz desafios significativos, como a carência de restrições padronizadas que garantam a eficácia das interações entre processos constituintes. A especificação de restrições é importante nesse contexto, pois elas são regras fundamentais para delimitar comportamentos permitidos, determinar dependências entre processos, guiar a execução coordenada das atividades e assegurar a conformidade da colaboração entre as organizações envolvidas [Cagnin and Nakagawa 2022].

Para apoiar a modelagem de PoP, o método M-PoP foi proposto como uma abordagem sistemática com o propósito de superar a dificuldade de modelar PoP, facilitando a identificação, especificação e a verificação de elementos e restrições associadas [Cagnin and Nakagawa 2022]. Oferece uma estrutura iterativa que permite a adaptação contínua às mudanças nas necessidades organizacionais e define três visões do PoP: Visão Geral do PoP, Visão de Missões do PoP e Visão Detalhada de Missão do PoP. Cada uma dessas visões possui um modelo específico que representa as restrições inerentes a ela, permitindo uma compreensão mais clara das interações e dependências entre os processos [Cagnin and Nakagawa 2022]. Além de utilizar a notação *Business Process Model and Notation* (BPMN) [Object Management Group 2013] para representar alguns modelos PoP, M-PoP propõe outros modelos na notação *feature model* [Cagnin and Nakagawa 2022] para representar graficamente restrições em PoP. Porém, há carência de uma representação para evitar imprecisão, falta de clareza e ambiguidade em restrições de PoP.

Por PoP se tratar de um conceito relativamente recente, ainda não há estudos que estabeleçam diretrizes claras e uma linguagem formal e padrão para especificar restrições. Essa falta de padronização torna o gerenciamento de restrições no PoP mais desafiador, suscetível a inconsistências devido à imprecisão ou ambiguidade que podem ocorrer quando as restrições são apenas representadas em modelos. Como consequência, dificulta o entendimento do dinamismo do PoP, a interoperabilidade entre diferentes processos de negócios de organizações distintas e, consequentemente, a integração entre os sistemas de software que automatizam tais processos. Portanto, é essencial investigar mecanismos para padronizar e apoiar a especificação de restrições no PoP.

Na modelagem de processos de negócios, restrições representam regras que regulam o comportamento, a execução e a interação entre processos, garantindo sua conformidade com requisitos organizacionais, legais ou operacionais. Sob essa perspectiva, restrições são importantes no contexto de PoP, pois atuam na coordenação de processos constituintes de organizações distintas, alinhando seus comportamentos e interações para que seus objetivos em comum sejam alcançados de maneira consistente e previsível. A literatura existente aborda principalmente restrições em modelos de processo individual na notação BPMN, concentrando-se em aspectos como conformidade [Bergman et al. 2023], temporalidade [Ocampo-Pineda et al. 2022] [Arévalo Maldonado et al. 2016], atribuição de execu-

res [Fattouch et al. 2022] e privacidade de dados [Labda et al. 2014]. No entanto, esses estudos raramente consideram cenários distribuídos, dinâmicos e interorganizacionais, como os encontrados em PoP. Isso reforça a necessidade de buscar mecanismos adequados para gerenciar restrições em ambientes nos quais múltiplos processos de organizações distintas interagem, muitas vezes com variabilidades em atividades que o negócio requer em determinadas circunstâncias, missões de negócio concorrentes ou conflitantes, exceções que representam comportamentos inesperados do negócio e mudanças constantes de configuração dos processos de negócio envolvidos devido ao dinamismo das alianças de organizações.

Diante desse cenário, o objetivo deste trabalho é ajudar analistas de negócio durante a tarefa de especificar restrições com o intuito de complementar os modelos preconizados pelo M-PoP para aumentar a clareza e precisão da representação de PoP, facilitando o seu entendimento. Para alcançar esse objetivo, foi conduzido um mapeamento sistemático da literatura seguindo o método proposto por [Kitchenham and Charters 2007], que permitiu selecionar estudos primários, extrair dados relevantes e responder às questões de pesquisa formuladas sobre especificação de restrições em modelos BPMN, com enfoque nas linguagens e tipos de restrições mais utilizados. Com base nos resultados obtidos, foram definidos critérios para analisar as linguagens existentes e selecionar a mais adequada para o trabalho. A linguagem *Object Constraint Language* [Object Management Group 2014] (OCL) foi a escolhida por ser linguagem que atendeu a maioria dos critérios definidos. Posteriormente, foi estabelecido um conjunto de doze regras para especificar restrições em modelos do PoP, além de incorporar dois tipos de restrições (ou seja, restrições temporais e de privacidade) não previstos pelo M-PoP. A conformidade da especificação definida em cada regra com a sintaxe da linguagem OCL foi validada por meio das ferramentas EclipseOCL ¹ e *UML-based Specification Environment* (USE) ². O trabalho também oportunizou a evolução do metamodelo do PoP [Cagnin and Nakagawa 2022], que contém todos os principais conceitos envolvidos e relacionamentos entre eles.

A estrutura deste artigo está organizada da seguinte forma. A Seção 2 apresenta o embasamento teórico necessário para compreender o conceito de PoP, o método M-PoP e os diferentes tipos e linguagens ou notações de restrições em BPMN considerados na literatura. A Seção 3 descreve o mapeamento sistemático da literatura, incluindo sua execução e análise dos estudos primários. A Seção 4 apresenta o método de pesquisa utilizado para selecionar a linguagem mais adequada e para definir as regras propostas para especificar restrições em PoP, considerando a linguagem selecionada. Para facilitar o entendimento de cada regra, essa seção também apresenta a sua aplicação em um PoP real. Por fim, a Seção 5 conclui e discute as contribuições e limitações do trabalho e aponta sugestões de trabalhos futuros.

2. Embasamento Teórico

Nesta seção são apresentados os fundamentos para facilitar o entendimento e direcionar o desenvolvimento deste trabalho.

2.1. Processos-de-Processos de Negócios

Processo-de-processos de negócios (PoP) é um conceito inovador que descreve um conjunto complexo e dinâmico de processos de negócios resultantes de alianças entre organizações, como aquisições, fusões e parcerias. Nesse contexto, múltiplas organizações colaboram e interagem por meio de seus próprios processos de negócios e sistemas de software para alcançar objetivos estratégicos em comum [Cagnin and Nakagawa 2021].

A adoção de PoP pode trazer diversos benefícios para a gestão de grandes processos de negócios dinâmicos dentro de alianças estratégicas de organizações, como melhor compreensão da complexidade, flexibilidade e adaptabilidade, otimização da colaboração entre organizações, suporte

¹<https://projects.eclipse.org/projects/modeling.ocel>

²<https://github.com/useocl/use>

a missões paralelas e concorrentes, gestão de exceções e contingências, além de avanços na gestão de processos de negócios [Cagnin and Nakagawa 2021].

No contexto de PoP, diversos conceitos são fundamentais para compreender sua dinâmica. O PoP é composto por processos de diferentes organizações, que colaboram para alcançar objetivos comuns. A independência operacional e gerencial garante que cada processo constituinte possa operar e ser gerenciado individualmente, mesmo quando integrado ao PoP. Os processos constituintes são os processos individuais de cada organização que participam do PoP e colaboram para alcançar os objetivos comuns. A distribuição refere-se ao fato de que os processos constituintes estão distribuídos em várias organizações e se comunicam entre si por meio de mensagens. A melhoria contínua e o comportamento emergente indicam que os PoP estão sujeitos a mudanças constantes e que comportamentos complexos podem surgir da colaboração entre os processos constituintes. A reconfiguração dinâmica permite que os PoP alterem sua configuração durante a execução para garantir estabilidade. As missões do PoP são os objetivos de negócios que o PoP visa alcançar, enquanto as configurações do PoP representam os diferentes caminhos possíveis para atingir essas missões. Além disso, diferentes tipos de restrições devem ser considerados, tais como restrições de missões, de processos constituintes e restrições de variabilidade.

2.2. M-PoP

O M-PoP é um método desenvolvido para modelar processos de negócios complexos e dinâmicos, conhecido como PoP. M-PoP oferece uma abordagem sistemática, utilizando técnicas como o BPMN [Object Management Group 2013] e define três etapas principais: identificação dos elementos do PoP, modelagem do PoP e verificação dos modelos do PoP [Cagnin and Nakagawa 2022].

De acordo com Cagnin e Nakagawa [Cagnin and Nakagawa 2022], o M-PoP propõe a modelagem do PoP em três visões principais, apresentadas na Figura 1:

- Visão Geral do PoP: Nesta visão, são identificados e representados os processos constituintes das organizações pertencentes à aliança, além das colaborações e restrições existentes entre eles.
- Visão das Missões do PoP: Aqui são representadas todas as missões do PoP em um alto nível de abstração, juntamente com suas restrições e como podem ser realizadas.
- Visão Detalhada de Missão do PoP: Esta visão detalha cada missão do PoP, dando enfoque em todas as configurações e restrições de variabilidade.

Este trabalho de TCC possui foco especial na especificação formal das restrições do PoP, complementando os modelos de restrições preconizados pelo M-PoP.

2.2.1. Modelagem do PoP e suas restrições

No M-PoP, a modelagem do PoP envolve a identificação dos elementos do PoP, como missões e processos constituintes, e a representação desses elementos em suas visões. É definido um conjunto de passos para modelar o PoP de maneira sistemática e iterativa. O processo envolve a identificação dos elementos do PoP, a modelagem do PoP nas diferentes visões e a verificação dos modelos para garantir sua consistência e conformidade com os requisitos estabelecidos.

Conforme abordado no M-PoP, existem três tipos de modelos de restrições (i) *Constituent Processes Constraints Model* (Modelo de Restrições dos Processos Constituintes): refere-se à representação das restrições específicas relacionadas aos processos individuais que compõem o PoP e que precisam colaborar com outros. O objetivo dessas restrições é garantir que os processos constituintes do PoP operem de forma eficiente e alinhada com os objetivos da aliança de organizações; (ii) *PoP Missions Constraints Model* (Modelo de Restrições das Missões do PoP): representa as restrições relacionadas às missões, em conformidade com os objetivos estratégicos das alianças. Essas restrições englobam a ordem de execução entre as missões do PoP e especificam também aquelas

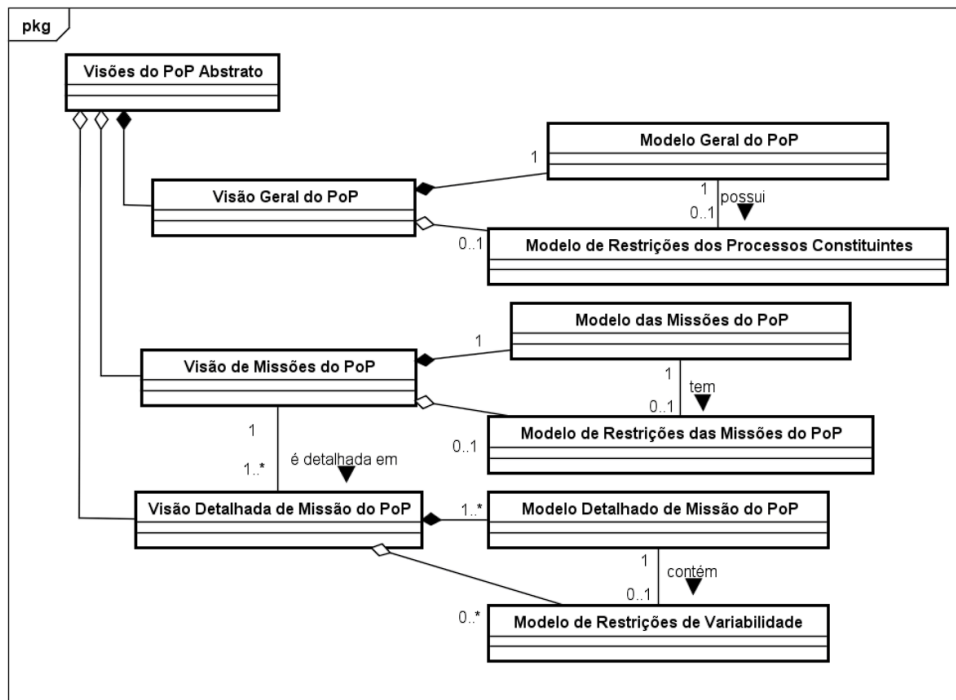


Figura 1. Visões do PoP abstrato e seus respectivos modelos (adaptado de [Cagnin and Nakagawa 2022])

que são conflitantes; (iii) *Variability Constraints Model* (Modelo de Restrições de Variabilidade): refere-se à variações em atividades dos processos de negócio das organizações envolvidas na aliança. As restrições de variabilidade são especificadas para controlar e gerenciar essas variações, garantindo a consistência e a eficácia das operações de negócio no PoP [Cagnin and Nakagawa 2022].

Salienta-se que as restrições do PoP podem abranger limitações de tempo e de recursos, e outras condições ou critérios necessários para garantir que o comportamento do PoP esteja alinhado com os propósitos da parceria entre as organizações.

As restrições devem ser modeladas de forma estruturada e integrada nas diferentes visões do PoP, em modelos de restrições representados em notação específica como o *feature model*. Isso inclui: (i) Identificação clara das restrições relevantes para o PoP alcançar os objetivos de negócio da aliança de organizações; (ii) Representação das restrições nas visões do PoP, em modelos específicos; e (iii) Verificação e validação das restrições para garantir sua completude e conformidade com os objetivos de negócio estabelecidos.

A elaboração dos modelos de restrições pode ocorrer paralelamente com outras etapas do M-PoP que modelam os demais modelos das visões, refletindo a complexidade envolvida no gerenciamento do processos de negócio em um ambiente de colaboração entre organizações.

2.2.2. Necessidade de padronizar a especificação de restrições no PoP

A padronização da especificação de restrições no contexto do PoP é um passo essencial para garantir consistência e clareza na modelagem desses processos de negócio complexos. Embora o método M-PoP estabeleça três modelos distintos de restrições, ainda não existe uma linguagem ou notação adotada de maneira padronizada que complemente formalmente a especificação dessas restrições. Padronizar é fundamental para garantir a interoperabilidade e a eficácia na colaboração entre diferentes organizações. As restrições no PoP podem incluir limitações sobre a ocorrência de processos constituintes, a sequência de execução de missões e a variabilidade das atividades. Elas são essenciais para assegurar que os processos operem de maneira coesa e eficiente [Cagnin and Nakagawa 2022].

Cagnin e Nakagawa [Cagnin and Nakagawa 2022] sugerem a adoção de *feature model* para

representar restrições em PoP de forma clara e estruturada. Esses modelos ajudam a visualizar as relações entre elementos do PoP e as condições sob as quais elas podem ser aplicadas.

A padronização da especificação de restrições permite que diferentes organizações compreendam PoP de forma consistente. Isso facilita a comunicação e a colaboração entre as partes envolvidas, reduzindo a ambiguidade e melhorando a eficiência operacional. Sendo assim, é um requisito essencial para a evolução do método M-PoP e para a aplicação prática desse método em cenários reais.

2.3. Restrições em Modelos BPMN

A seguir são apresentados os diferentes tipos de restrições encontrados em diversos estudos que tratam de restrições em modelos de processos de negócio utilizando a notação BPMN. Essas restrições desempenham um papel crucial na verificação e garantia da conformidade e execução correta dos processos de negócio.

- Restrições de conformidade [Bergman et al. 2023]: são usadas para garantir que a execução dos processos esteja alinhada com as regras ou padrões estabelecidos no modelo BPMN. Essas restrições são geradas a partir dos diagramas BPMN e verificadas em relação à execução real do processo. As restrições de conformidade são fundamentais para assegurar que o processo de negócio siga corretamente o fluxo de trabalho especificado, sinalizando qualquer desvio como não conformidade ao fluxo do processo que foi previamente estabelecido. Exemplos comuns incluem:
 - Sucessão (A precede B): verifica se, sempre que a atividade “A” ocorre “B” também ocorre, logo após “A”.
 - Escolha Exclusiva (B XOR C): verifica se, em uma bifurcação, apenas uma das atividades (B ou C) é executada.
- Restrições temporais [Ocampo-Pineda et al. 2022]: modelam a flexibilidade e incertezas dos processos, permitindo o ajuste dinâmico dos cronogramas de execução dos processos, conforme necessário. As restrições temporais podem ser classificadas de acordo com o grau de controle e flexibilidade. Alguns tipos identificados incluem:
 - Durações controláveis: definem limites de tempo que podem ser ajustados pelo executor do processo dentro de intervalos pré-definidos. Exemplo: uma tarefa pode ter duração ajustável entre 5 e 10 minutos;
 - Durações contingentes: dependem de fatores externos, como eventos fora do controle do executor. Exemplo: uma tarefa pode durar de 2 a 8 horas; e
 - Restrições relativas: especificam o intervalo de tempo entre duas atividades, como o início de uma tarefa após a conclusão de outra, ajudando a modelar interdependências temporais dentro dos processos.
- Restrições de tempo em BPMN 2.0 [Arévalo Maldonado et al. 2016]: contempla uma ampla gama de restrições de tempo, diferentemente das restrições temporais, descritas no item anterior, não lida com incertezas ou probabilidades. As restrições de tempo levantadas são:
 - Restrições flexíveis: permitem que o início ou término de uma atividade seja ajustado dentro de um intervalo, como “o mais cedo possível” ou “o mais tarde possível”; e
 - Restrições inflexíveis: fixam horários exatos para início e término de atividades, como “Must Start On” (MSON) ou “Must Finish On” (MFON).
- Restrições de atribuição de tarefas e executores [Fattouch et al. 2022]: identificadas em um estudo realizado no contexto da Internet das Coisas Robóticas (IoRT), essas restrições são voltadas para a escolha adequada de dispositivos executores em processos de negócio.
 - Restrições para situações específicas: garantem que o dispositivo correto (robô, atuador IoT ou bot) seja escolhido para cada tarefa;
 - Tarefas que exigem movimento vs. sem movimento: apenas um robô como máquina pode realizar uma tarefa que exige movimento. Um dispositivo IoT (atuador) pode executar tarefas sem movimento.

- Restrições de *scripts*: tarefas que exigem execução de código devem ser realizadas por bots; e
- Exclusividade de execução: uma tarefa realizada por um atuador não pode ser realizada por um robô, e vice-versa. Isso garante que não haja sobreposição ou conflito na atribuição de tarefas entre esses dois tipos de executores.
- Restrições de privacidade [Labda et al. 2014]: esse tipo de restrição assegura a proteção de dados pessoais e é essencial para garantir a conformidade com requisitos de privacidade durante o *design* e a execução dos processos. Como restrições de privacidade encontradas tem-se:
 - Restrições de controle de acesso: definem quem pode acessar determinados dados;
 - Restrições de separação de tarefas (SoT): garantem que diferentes usuários executem partes distintas do processo;
 - Restrições de vinculação de tarefas (BoT): permitem que um usuário realize várias tarefas, desde que respeite suas permissões;
 - Restrições de consentimento do usuário (*User Consent*): garantem que dados pessoais só possam ser acessados ou manipulados se houver consentimento explícito do proprietário dos dados; e
 - Restrições de necessidade de saber (*Necessity to Know* - NtK): asseguram que os usuários só podem acessar as informações estritamente necessárias para realizar suas funções. Isso limita o acesso e evita a exposição desnecessária de dados.

Os estudos mostram a diversidade de tipos de restrições aplicáveis em modelos de processos de negócio e como elas desempenham papéis essenciais na garantia de conformidade, precisão, flexibilidade e segurança na execução de processos de negócios.

2.3.1. Notações e linguagens para especificação de restrições

A linguagem DECLARE é uma linguagem declarativa amplamente utilizada para especificar restrições de processos de negócios [Bergman et al. 2023]. Diferente de linguagens imperativas, que definem um fluxo rígido de atividades, DECLARE permite que as atividades sejam descritas de forma mais flexível, usando restrições que precisam ser atendidas, mas sem determinar exatamente como o processo deve ocorrer. DECLARE é uma das linguagens usadas pela ferramenta BPMN2Constraints para gerar restrições a partir de diagramas BPMN [Bergman et al. 2023]. As restrições geradas podem incluir sucessões, co-existências e escolhas exclusivas, permitindo uma maior variabilidade no comportamento do processo, contanto que as restrições sejam respeitadas.

Finite-trace Linear Temporal Logic (LTLf) é uma variação da lógica temporal linear, aplicada em traços finitos, como os processos de negócios [Bergman et al. 2023]. Essa linguagem é usada para descrever restrições temporais sobre a ordem de eventos em um processo. No contexto do BPMN2Constraints, LTLf é uma das linguagens na qual as restrições são geradas automaticamente a partir dos diagramas BPMN. As restrições podem ser expressas de forma a garantir que certos eventos ocorram antes ou depois de outros dentro de um traço finito de execuções.

SIGNAL é uma linguagem de consulta proprietária utilizada na indústria, especialmente pela SAP Signavio, para consultar processos [Bergman et al. 2023]. Ela é baseada em expressões regulares para consultar o fluxo de controle dos processos a fim de analisar a sequência de atividades que ocorrem em um processo de negócio. Assim como DECLARE e LTLf, a ferramenta BPMN2Constraints gera restrições também nesta linguagem, que é especialmente utilizada em aplicações de grande escala na indústria.

A notação CSTNU (*Constraint Systems with Time, Numerical values, and Uncertainty*) é uma rede temporal voltada para a modelagem de restrições temporais em processos de negócios que lidam com incertezas [Ocampo-Pineda et al. 2022]. Ela permite especificar dois tipos de durações: (i) durações controláveis: são aquelas que estão sob o controle do agente que executa o processo. Por

exemplo, a duração de uma tarefa pode ser ajustada ou controlada diretamente pelo executor; e (ii) durações contingentes: são durações que estão fora do controle do agente executor. Essas durações são determinadas por agentes externos e o agente executor só pode observá-las após a conclusão da tarefa. Na CSTNU, cada processo de negócio modelado com restrições temporais deve ser validado para garantir a controlabilidade dinâmica. Essa propriedade verifica se existe uma maneira de agendar as atividades controláveis de forma que todas as restrições temporais sejam atendidas, independentemente das variações nas durações contingentes. Durante a verificação, o modelo BPMN é convertido em uma rede temporal condicional equivalente (CSTNU) e, então, um algoritmo de verificação é utilizado para assegurar que o modelo é dinamicamente controlável [Ocampo-Pineda et al. 2022].

A OCL é uma linguagem formal amplamente utilizada para descrever expressões que restringem elementos de um modelo, como em modelagem de processos de negócio [Fattouch et al. 2022]. A OCL permite a especificação rigorosa de regras e restrições que determinam o comportamento dos processos, incluindo aspectos como restrições temporais e dependências entre atividades. Essa linguagem de especificação pura tem capacidade de navegar e aplicar restrições diretamente nas classes e associações de um metamodelo. Por meio da OCL, é possível modelar aspectos como duração de atividades, cardinalidade de iterações e condições de ausência, sem modificar o fluxo de controle principal do processo [Arévalo Maldonado et al. 2016, Fattouch et al. 2022]. A OCL garante que regras sejam respeitadas durante a execução de processos [Arévalo Maldonado et al. 2016].

A linguagem EssentialOCL é o subconjunto mínimo da linguagem de restrições OCL necessário para operar com a EMOF (*Essential Meta-Object Facility*) [Ma et al. 2015]. A EssentialOCL é usada para definir invariantes, ou seja, regras e condições que devem ser mantidas em um metamodelo ou modelo dentro do contexto de desenvolvimento baseado em modelos. Ela permite que desenvolvedores especifiquem restrições sobre os elementos dos modelos, como classes e associações, assegurando que os modelos sigam as regras definidas [Ma et al. 2015].

Por fim, *Semantic Web Rule Language* (SWRL) é uma linguagem de regras que permite representar e compreender sobre restrições e requisitos de privacidade em modelos de processos de negócios [Labda et al. 2014]. Essa linguagem é utilizada para especificar as condições sob as quais certos recursos podem ser acessados e quais ações podem ser realizadas por diferentes usuários dentro de um processo de negócio. É usada para transformar construtos de modelagem de processos de alto nível em regras executáveis que garantem a aplicação de restrições de privacidade durante a execução dos processos [Labda et al. 2014].

3. Revisão da literatura

Para levantar os trabalhos relevantes sobre representação de restrições em modelos BPMN foi conduzido um mapeamento sistemático da literatura, seguindo as etapas do processo definido por [Kitchenham and Charters 2007]. As seções a seguir apresentam a descrição de cada uma dessas etapas que se referem ao planejamento, execução, e análise e divulgação dos resultados do mapeamento realizado.

3.1. Planejamento da pesquisa

Esta seção descreve o planejamento detalhado da pesquisa realizada para identificar e analisar artigos relevantes sobre notações ou linguagens utilizadas para especificar restrições em modelos de processos de negócio em BPMN.

O objetivo desta pesquisa é identificar e compreender as notações ou linguagens mais comumente empregadas para tratar restrições em modelos BPMN.

Para isso, foram formuladas as seguintes questões de pesquisa:

- QP1 - Quais são as notações ou linguagens utilizadas para especificar restrições em modelos de processos de negócio em BPMN? *Esta questão busca levantar os mecanismos utilizados para especificar restrições de negócio.*

- QP2 - Quais são os tipos de restrições especificados nos estudos selecionados? *Esta questão busca correlacionar os tipos de restrições com as linguagens ou notações utilizadas para compreender a forma mais adequada para especificar cada tipo de restrição.*
- QP3 - Qual é o domínio do modelo de processos de negócio? *Esta questão busca identificar se existe correlação entre domínios de processos de negócios semelhantes e as linguagens ou notações de restrições adotadas.*
- QP4 - O modelo de processos de negócio se refere a processos de negócios de várias organizações que trabalham em conjunto e como as restrições são tratadas? *Esta questão busca identificar como é feita a especificação de restrições em modelos de processos de negócio de alianças de organizações.*
- QP5 - O estudo fornece diretrizes para ajudar os analistas de negócio na identificação e especificação de restrições? *Esta questão busca observar o apoio da literatura existente para identificar e especificar restrições de negócio.*

Para identificar os artigos de interesse, foi adotada uma estratégia de busca eletrônica na biblioteca Scopus³, reconhecida pela sua abrangência e relevância na área de Engenharia de Software e de Administração. A elaboração da string de busca foi cuidadosamente planejada, utilizando palavras-chave e sinônimos comumente associados ao tema, como “constraint”, “BPMN”, “notation” e seus respectivos sinônimos relevantes para o contexto deste trabalho, conforme exibido na Tabela 1.

Tabela 1. Termos e sinônimos importantes

Termos Importantes	Sinônimos
Constraint	Restriction
BPMN	Business Process Model and Notation
Notation	Language, Specification

A calibragem da string de busca foi realizada para otimizar a precisão dos resultados, ajustando a inclusão ou exclusão de sinônimos conforme necessário. A string de busca calibrada ficou definida como:

(constraint OR restriction) AND (BPMN OR “Business Process Model and Notation”) AND (notation OR language OR specification)

Foram estabelecidos os seguintes critérios de para orientar a seleção dos artigos relevantes.

- **Critério de inclusão:**
 - **CI1:** O trabalho define ou utiliza mecanismos (notação, linguagem ou especificação) para especificar restrições em modelos de processos de negócio em BPMN.
- **Critérios de exclusão:**
 - CE1: O trabalho não define ou utiliza mecanismos para especificar restrições em modelos de processos de negócio em BPMN.
 - CE2: O trabalho especifica restrições em modelos de processos de negócio em BPMN de maneira *ad hoc*, sem utilizar um mecanismo apropriado.
 - CE3: O trabalho especifica restrições em modelos de processos de negócio em notação diferente de BPMN.
 - CE4: O trabalho não está nos idiomas inglês ou português.
 - CE5: O trabalho não é revisado por pares (por exemplo, livros, sumários de conferências, palestras, artigos não revisados por pares e resumos).
 - CE6: O artigo não está disponível.

³<https://scopus.com/>

3.2. Execução

A etapa de Execução se iniciou no dia 05 de abril de 2024, com a realização de buscas na base digital Scopus, utilizando a string calibrada na etapa anterior. Após a execução da busca inicial, os 168 estudos retornados foram filtrados em três etapas de seleção.

Etapa 1: Selecionar os artigos que possuem mais de 10 citações por terem alta relevância e impacto e os publicados a partir de 2010 para identificar os estudos produzidos nos últimos anos.

Etapa 2: Aplicar os critérios de seleção no resultado da Etapa 1, levando em consideração o título, *abstract* e palavras-chave.

Etapa 3: Aplicar os critérios de seleção no resultado da Etapa 2, levando em consideração a leitura completa do artigo.

A primeira etapa de seleção foi realizada entre 15 e 19 de abril de 2024, obtendo-se 80 estudos, com redução significativa no número de artigos retornados pela Scopus. A segunda etapa ocorreu de 3 a 24 de maio de 2024, por meio da leitura do título e do resumo dos artigos resultando em 22 estudos. Nessa etapa de seleção, os critérios de exclusão que mais eliminaram estudos não relevantes foram CE1, CE3 e CE5. Posteriormente, na terceira etapa de seleção, foi conduzida uma leitura completa dos artigos resultando em 7 estudos primários selecionados.

Para caracterizar os estudos primários e para responder as questões de pesquisa definidas na Seção 3.1, foram extraídos os seguintes dados de cada estudo por meio de um formulário de extração em planilha Excel: (i) ID do estudo, (ii) ano de publicação, (iii) tipo de publicação (Conferência, Periódico, Workshop), (iv) objetivo geral do estudo, (v) principais resultados, (vi) linguagem ou notação utilizada, (vii) tipos de restrições especificados, (viii) domínio do modelo de processo, (ix) se o modelo de processos representa várias organizações, quais são as diretrizes para identificar e (x) especificar restrições, (xi) apoio computacional (caso exista) e (xii) observação adicional (caso necessário).

A Tabela 2 mostra os resultados quantitativos das etapas de seleção 2 e 3, pois a seleção na Etapa 1 foi utilizada apenas para levantar os estudos com maior impactos e mais recentes, conforme critérios mencionados anteriormente.

Tabela 2. Resultados Quantitativos

Etapa	Total Etapa Anterior	CE1	CE2	CE3	CE4	CE5	CE6	Total
Etapa 2	80	47	0	9	0	2	0	22
Etapa 3	22	1	10	3	0	0	1	7

Os estudos selecionados foram analisados para identificar as práticas predominantes de especificação de restrições em BPMN, contribuindo para o avanço do conhecimento nessa área específica da modelagem de processos de negócio. O total de apenas sete estudos primários evidencia o quão específico e ainda pouco explorado é o tema abordado neste artigo.

3.3. Caracterização dos Estudos Primários

Os estudos primários selecionados foram publicados em diferentes tipos de veículos ao longo dos anos, conforme Tabela 3. A Figura 2 mostra uma visão geral dos estudos quanto ao ano e tipo de publicação, domínio e autores envolvidos. Mais especificamente, três estudos foram publicados em conferências, três em periódicos e um em workshop. Quanto aos autores, não há estudos publicados exclusivamente por autores da indústria, mostrando a importância da colaboração entre academia e indústria, como é caso do estudo S1.

O período de publicação dos estudos varia de 2011 a 2023, com uma concentração inicial de quatro estudos entre 2011 e 2016, seguida por publicações mais recentes, totalizando três estudos entre 2022 e 2023. Os domínios abordados nos estudos incluem Compras, Emergência, Gestão

de Eventos, Agricultura, Crédito Financeiro e Diversos⁴. Observa-se que alguns estudos (S1, S3) exploram domínios específicos, como domínio de crédito financeiro e Agricultura, mas também domínios diversos principalmente durante a avaliação do trabalho com o intuito de validar suas teorias em uma variedade de cenários possíveis. Ademais, os autores dos estudos são predominantemente provenientes da academia. Apenas um dos estudos envolveu a colaboração entre autores da academia e da indústria.

Tabela 3. Lista dos Estudos Primários

ID	Tipo de Publicação	Ano	Título	Referência
S1	Workshop	2023	BPMN2Constraints: Breaking down BPMN Diagrams into Declarative Process Query Constraints	[Bergman et al. 2023]
S2	Periódico	2022	TimeAwareBPMN-js: An editor and temporal verification tool for Time-Aware BPMN processes	[Ocampo-Pineda et al. 2022]
S3	Conferência	2022	Towards a Meta-Modeling Approach for an IoRT-Aware Business Process	[Fattouch et al. 2022]
S4	Periódico	2016	A metamodel to integrate business processes time perspective in BPMN 2.0	[Arévalo Maldonado et al. 2016]
S5	Periódico	2015	A generic model decomposition technique and its application to the Eclipse modeling framework	[Ma et al. 2015]
S6	Conferência	2014	Modeling of privacy-aware business processes in BPMN to protect personal data	[Labda et al. 2014]
S7	Conferência	2011	Quick fix generation for DSMLs	[Hegedüs et al. 2011]

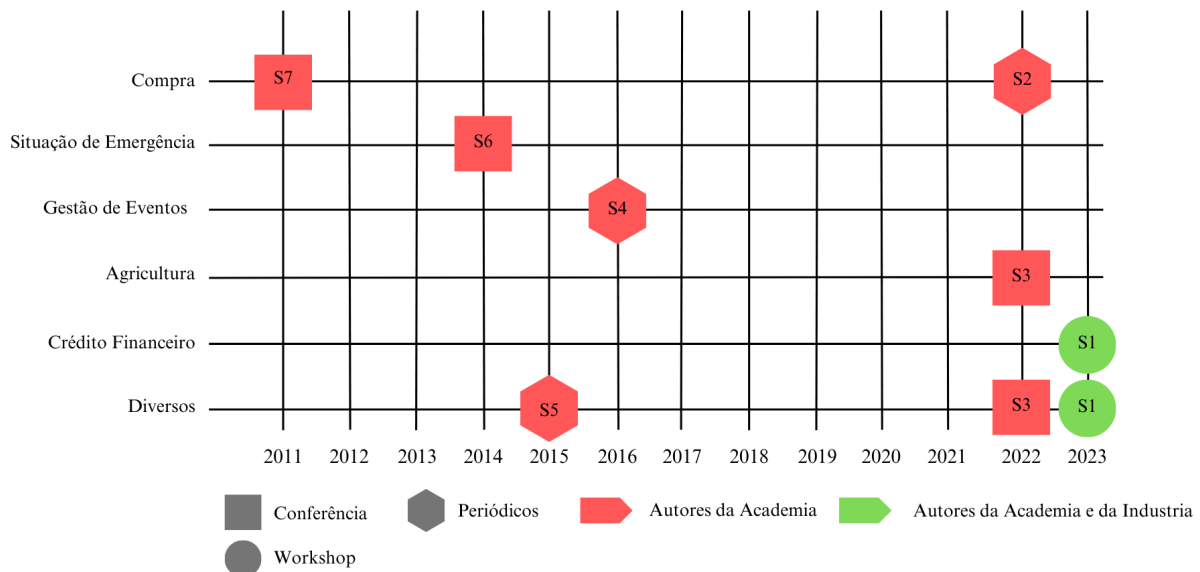
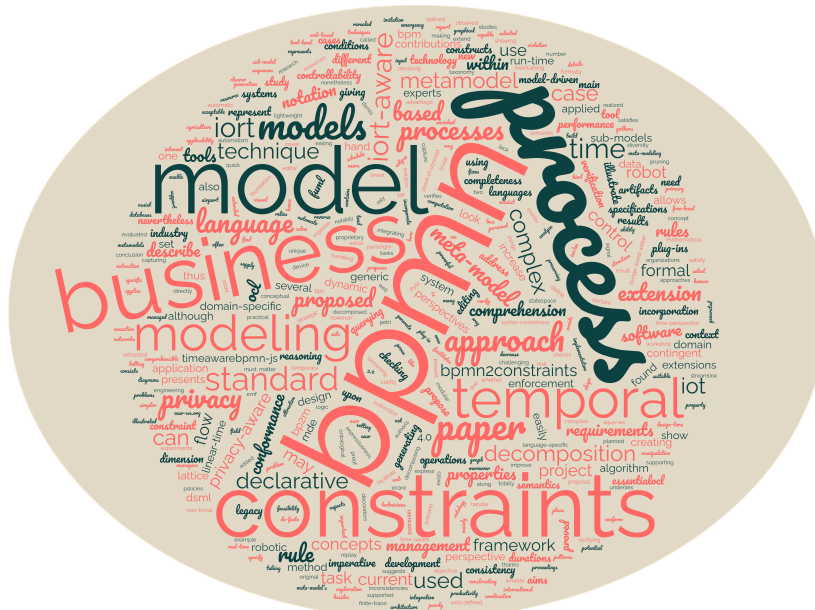
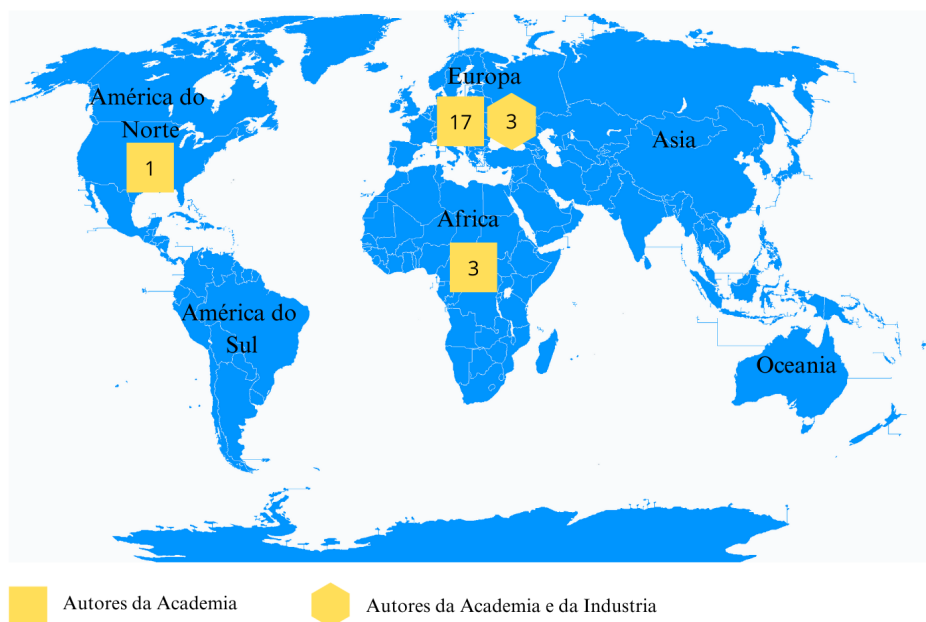


Figura 2. Visão Geral dos Estudos Primários

Conforme apresentado na Figura 3, os autores dos sete estudos primários estão distribuídos em três continentes (América do Norte, Europa e África). Ao todo, foram identificados 24 autores, dos quais a maioria pertence à academia (21 de 24), com exceção de três autores que atuam tanto na academia quanto na indústria. Diante disso, observa-se que o pouco envolvimento da indústria pode ser uma possível razão para a ausência de uma padronização de notação ou linguagem para especificar restrições em modelos BPMN.

⁴Os estudos classificados como domínio Diversos consideraram modelos de processos de negócio de repositórios da academia e da indústria.

A partir do conteúdo do resumo e das palavras-chaves dos estudos primários, foi construída uma nuvem de palavras (Figura 4) com o intuito de observar os termos mais destacados e, consequentemente, comumente citados nesses estudos. Os termos mais mencionados foram “BPMN” e “constraints”, como esperado, pois são o foco da pesquisa. Ademais, apareceram também os termos “model”, “business” e “process”, mostrando que os estudos estão alinhados com nosso objetivo que é a modelagem de restrições em modelos de processos de negócios. Os estudos mencionam também, de maneira recorrente, o termo “temporal”, uma vez que as restrições temporais são as mais estudadas em modelagem de processos de negócios por serem as mais comuns ou relevantes para a maioria dos domínios, culminando na definição de diversas linguagens para especificá-las, logo, “language” é outro termo em destaque.



3.4. Resultados

Nesta seção é apresentada a resposta de cada questão de pesquisa, obtida a partir da extração de dados dos estudos primários selecionados.

QP1 - Quais são as notações ou linguagens utilizadas para especificar restrições em modelos de processos de negócio em BPMN?

A especificação de restrições em modelos de processos de negócio, especialmente em BPMN, pode ser realizada por meio de várias linguagens e notações. Com base nos estudos selecionados, foi identificada uma diversidade de mecanismos. O estudo S1 utiliza as linguagens DECLARE, SIGNAL e LTLf para especificar restrições. Essas notações são especialmente úteis para definir restrições de conformidade, garantindo que o processo de negócios seja executado conforme as regras e diretrizes estabelecidas. Em S2 utiliza-se a notação CSTNU, a qual é uma extensão dos sistemas de restrições temporais que incorpora valores numéricos e incerteza. Essa notação é utilizada para modelar restrições temporais em ambientes incertos, onde os tempos exatos de execução podem variar ou serem desconhecidos.

Notavelmente, a OCL é utilizada em vários estudos (S3, S4, S5 e S7) e aparece como uma notação comum para especificar restrições em processos de negócios. A OCL se destaca por sua flexibilidade e aplicabilidade em diferentes cenários, como a restrição de seleção de executores de tarefas (S3) ou restrições de tempo e duração de processos/atividades (S4).

Outro exemplo de linguagem utilizada é a *Semantic Web Rule Language* (SWRL), mencionada no S6, que se foca em restrições relacionadas a atributos⁵ e privacidade.

QP2 - Quais são os tipos de restrições especificados nos estudos selecionados?

Os tipos de restrições variam de acordo com a linguagem ou notação utilizada nos estudos analisados. No S1, por exemplo, as restrições de conformidade são especificadas com o uso das linguagens DECLARE, SIGNAL e LTLf. Essas restrições garantem que as atividades do processo sigam determinadas regras ou padrões previamente definidos, assegurando a conformidade do processo. Em S2, o foco está em variações de restrições temporais, sendo elas restrições de durações controláveis, de durações contingentes e restrições relativas.

Já em S3 e S4, que utilizam OCL, são encontradas restrições mais específicas. No S3, as restrições relacionadas à seleção de executores de tarefas são cruciais em processos conscientes de *Internet of Robotic Things* (IoRT), garantindo que cada tarefa seja atribuída ao executor correto, seja um robô, dispositivo *Internet of Things* (IoT) ou bot. O S4 menciona várias restrições temporais, como restrições de tempo flexíveis e inflexíveis, bem como restrições de rede e latência.

O S5, que também faz uso do OCL, traz o conceito de restrições de classe, atributo e referência, enquanto o S6 utiliza a SWRL para especificar restrições de privacidade, sendo elas restrições de controle de acesso, separação de tarefas, vinculação de tarefas, consentimento do usuário e necessidade de saber.

Finalmente, S7 também utiliza OCL para especificar restrições estruturais, que asseguram a correta configuração dos elementos do modelo, e restrições negativas, que evitam configurações indesejadas ou incorretas.

A Tabela 4 sumariza as notações e linguagens utilizadas para especificar as restrições tratadas nos estudos primários, evidenciando a diversidade de tipos de restrições que podem ser modeladas e aplicadas em processos de negócios, dependendo da linguagem utilizada.

QP3 - Qual é o domínio do modelo de processos de negócio?

⁵ (Restrições relacionadas a atributos são regras que verificam propriedades ou valores específicos de elementos do processo (como tarefas, executores ou objetos de dados), determinando se uma atividade pode ou não ser realizada com base nesses atributos.

Tabela 4. Notação ou linguagem de especificação utilizada

ID do Estudo	Linguagem/Notação	Tipo de Restrição
S1	DECLARE, SIGNAL, LTLf	Conformidade
S2	CSNTU	Temporais, Durações Controláveis, Durações Contingentes, Relativas
S3	OCL	Restrições Específicas de atribuição de tarefas
S4	OCL	Flexíveis, inflexíveis
S5	OCL, EssentialOCL	Classe, Atributo, Referência
S6	SWRL	Privacidade, Controle de Acesso, Vinculação de Tarefas, Consentimento de Usuário, Necessidade de Saber
S7	OCL	Estruturais, Negativas

Os artigos selecionados revelam uma ampla variação de domínios de aplicação representados pelos modelos de processos de negócio, cada um ilustrando diferentes contextos nos quais as restrições são relevantes. S1 apresenta um modelo aplicado ao domínio de crédito financeiro, mas explora também dois repositórios de modelos de processos de negócio, sendo eles a SAP Signavio Academic Models (SAP-SAM) e SAP Signavio Process Explorer reference BPMN models (VBPMN), compostos por diversos domínios. Isso amplia o alcance da teoria ao demonstrar que ela pode ser aplicada em múltiplos cenários.

Em S2, o domínio abordado é o de compra, com a modelagem de um processo específico dessa área. De forma semelhante, S7 também foca em processos de compra.

O S3 traz o exemplo de um processo no setor de agricultura, embora também valide suas teorias em outros domínios que não são descritos no artigo. Já o S4 concentra-se no domínio de gestão de eventos, onde a modelagem de processos é crucial para a organização e execução eficaz das atividades envolvidas em eventos.

Em S5, observa-se uma abordagem mais ampla, sem restrição a um único domínio, explorando a aplicação de restrições em diversos setores. Por fim, S6 analisa processos em situações críticas, como situações de emergência, um domínio que demanda respostas rápidas e eficientes.

QP4 - O modelo de processos de negócios se refere a processos de negócios de várias organizações que trabalham em conjunto e como as restrições são tratadas?

Apenas em S6 o modelo de processos de negócios discutido pode ser aplicado a processos de várias organizações que trabalham em conjunto. O artigo sugere que as restrições de privacidade podem ser tratadas de maneira colaborativa, mesmo quando várias entidades estão envolvidas no processamento de dados pessoais.

Quando o processo envolve múltiplas organizações, as restrições de privacidade são tratadas com mecanismos de controle de acesso, separação de tarefas, vinculação de tarefas, consentimento do usuário e necessidade de saber, conforme descritos a seguir, com o intuito de assegurar a proteção dos dados pessoais em ambientes colaborativos.

- Controle de acesso entre organizações: As permissões de acesso são baseadas nos papéis e responsabilidades de cada organização dentro do processo. Por exemplo, em um cenário de resposta a emergências em um aeroporto (descrito em S6), diferentes entidades, como equipe médica, bombeiros e passageiros, têm diferentes níveis de acesso a informações sensíveis. Cada organização envolvida no processo deve respeitar as regras de privacidade estabelecidas para evitar o acesso não autorizado aos dados, de acordo com seus papéis e as permissões

associadas.

- Separação e vinculação de tarefas entre organizações: Cada organização pode ser responsável por uma parte do processo de negócios, mas não terá acesso a dados ou tarefas que não são de sua competência. A separação e vinculação de tarefas garante que os dados pessoais não sejam expostos desnecessariamente entre diferentes organizações.
- Consentimento do usuário: Em processos colaborativos, a questão do consentimento do usuário continua sendo fundamental. As organizações que trabalham juntas precisam garantir que os dados pessoais só sejam acessados ou processados por outras entidades com o consentimento explícito do usuário.
- Necessidade de saber (*Necessity to Know*): Quando várias organizações cooperam, as restrições de necessidade de saber limitam o acesso aos dados pessoais apenas àqueles que necessitam deles para executar suas funções. Isso é crucial para minimizar a exposição de dados, permitindo que as organizações compartilhem informações de maneira segura e controlada, com base na função de cada parte no processo.

QP5 - O estudo fornece diretrizes para ajudar os analistas de negócio na identificação e especificação de restrições?

Os estudos S3, S4 e S6 fornecem diretrizes claras e detalhadas para ajudar os analistas de negócio na identificação e especificação de restrições em diferentes contextos, desde a gestão de dispositivos IoT até a proteção da privacidade de dados e a sincronização temporal de atividades. Essas diretrizes são fundamentais para garantir que as restrições sejam aplicadas de forma consistente e que o modelo de processo de negócios atenda aos requisitos específicos de cada cenário. A Tabela 5 apresenta as diretrizes encontradas para identificar e especificar restrições em modelos de processos de negócio.

Tabela 5. Especificação de Restrições

ID do Estudo	Identificação de Restrições	Especificação de Restrições
S3	Movimento Requerido: Tarefas que envolvem movimento físico devem ser realizadas por robôs, garantindo que somente dispositivos com mecanismos de movimentação executem essas atividades. Execução de Scripts: Tarefas que requerem a execução de scripts devem ser atribuídas a bots, assegurando que as automações de software sejam tratadas adequadamente. Exclusividade entre Atuadores e Robôs: Tarefas que podem ser executadas por atuadores IoT não devem ser atribuídas a robôs, e vice-versa, assegurando que os dispositivos corretos sejam escolhidos com base nas capacidades necessárias.	As diretrizes são implementadas e validadas usando a OCL, permitindo a correta atribuição de tarefas aos executores mais adequados. Isso assegura que as restrições sejam aplicadas de acordo com as características de cada tarefa, como mobilidade ou automação de software.
S4	Tipo de Regra Existente no Sistema de Origem: A partir dos dados de sistemas legados, são extraídas regras temporais relacionadas aos logs dos processos. Mapeamento de Artefatos: Relaciona os elementos dos sistemas legados com as atividades do modelo BPMN, garantindo a correspondência entre dependências temporais e atividades. Contexto das Restrições: As regras temporais são aplicadas de acordo com as atividades do processo, respeitando durações fixas ou flexíveis, conforme a lógica de negócios.	As restrições temporais são especificadas com atributos como minDur (duração mínima), maxDur (duração máxima), e regras de dependência como Start to Start (SS) ou Finish to Finish (FF). A OCL é usada para garantir que as dependências sejam respeitadas e que as atividades estejam sincronizadas corretamente.
S6	O estudo orienta a identificação de restrições de privacidade com base em perguntas como: “O que é informação privada?” e “De quem essa informação está sendo protegida?”. Esses conceitos ajudam a definir as ações permitidas sobre os dados, garantindo a proteção adequada.	As restrições de privacidade são especificadas em cinco requisitos principais utilizando a linguagem SWRL: Controle de Acesso, Separação de Tarefas, Vinculação de Tarefas, Consentimento do Usuário e Necessidade de Saber. Essas diretrizes garantem que apenas usuários autorizados possam acessar e manipular dados pessoais, respeitando os limites estabelecidos para a privacidade em todas as fases do processo.

4. Método de Pesquisa

Para elaborar as regras com o intuito de especificar restrições no contexto do PoP, foram seguidos oito passos, conforme exibidos na Figura 5 e mencionados a seguir. Em particular, os passos 2 a 6 contêm uma ou mais regras de especificação de restrição (RER), conforme descrito em detalhes nas subseções a seguir.

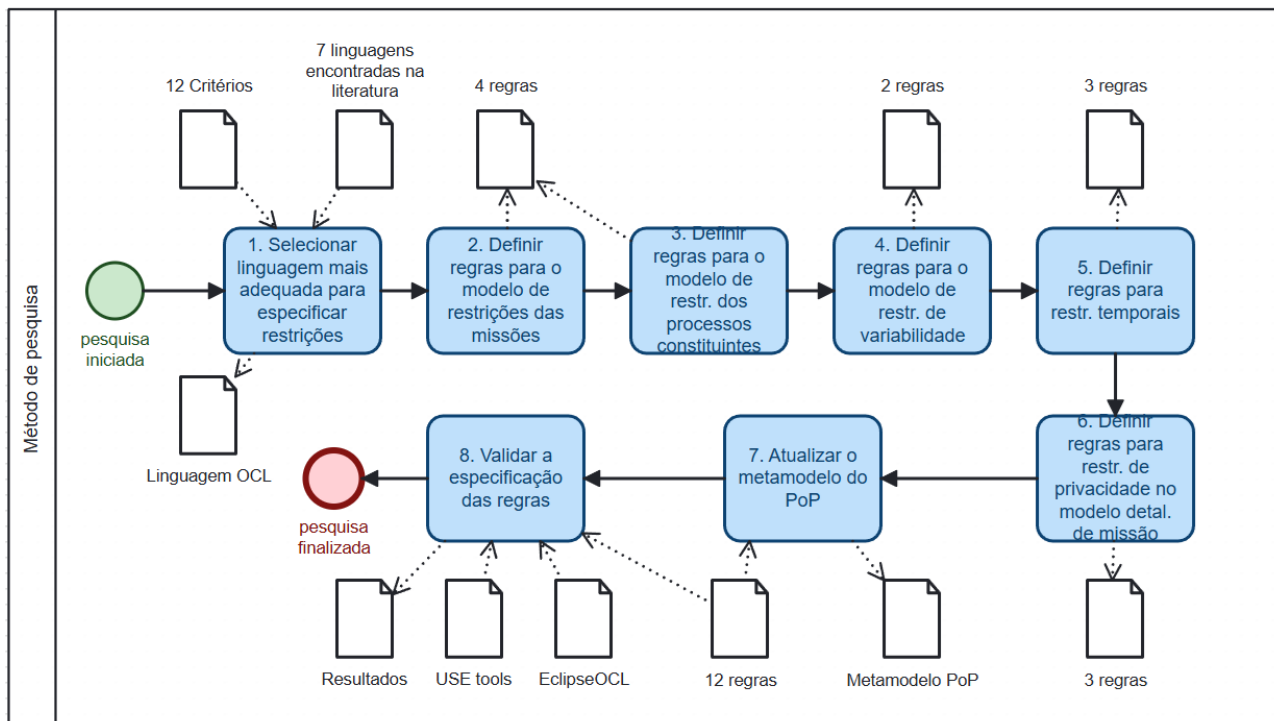


Figura 5. Passos para a definição das regras para especificar restrições em PoP

- **Passo 1:** Selecionar a notação/linguagem mais adequada para especificar restrições em PoP;
- **Passo 2:** Definir regras para especificar restrições do modelo de restrições das missões do PoP;
- **Passo 3:** Definir regras para especificar restrições do modelo de restrições dos Processos Constituintes do PoP;
- **Passo 4:** Definir regras para especificar restrições do modelo de restrições de variabilidade do PoP;
- **Passo 5:** Definir regras para especificar restrições temporais no PoP;
- **Passo 6:** Definir regras para especificar restrições de privacidade no modelo detalhado de missão do PoP;
- **Passo 7:** Atualizar o metamodelo do PoP; e
- **Passo 8:** Validar a especificação de cada regra definida.

Para facilitar o entendimento e a aplicação de cada regra de especificação definida (Subseções 4.2 a 4.6), é apresentado um exemplo de aplicação de cada uma no contexto do PoP Saúde Pública brasileira. Esse PoP refere-se a um conjunto de processos de negócio complexos e dinâmicos que envolvem diversas organizações e unidades de saúde, como a Unidade Central de Regulação (UCR), Unidades Hospitalares de Referência (UHR), Unidades de Atendimento Móvel (UAM) e o Serviço de Compartilhamento de Status de Leitos Hospitalares (SLU) [Cagnin and Nakagawa 2022]. O PoP Saúde Pública tem como uma das missões principais a redução do tempo de espera dos pacientes para internação, promovendo a colaboração entre essas entidades envolvidas para otimizar o fluxo de atendimento e a alocação de recursos. O PoP Saúde Pública busca integrar os processos constituintes de cada organização, permitindo uma comunicação eficaz e a troca de informações em tempo real. Isso

é essencial para garantir que os pacientes recebam atendimento adequado e oportuno, especialmente em situações de emergência [Cagnin and Nakagawa 2022].

Os operadores OCL utilizados nas doze RER, definidas nos Passos 2 a 6, estão resumidos na Tabela 6.

Tabela 6. Operadores OCL utilizados neste trabalho

Palavra-chave	Descrição
<code>context</code>	Define a classe na qual a restrição é aplicada (por exemplo, modelo de restrições das missões do PoP, modelo de restrições dos processos constituintes do PoP, modelo detalhado de missão do PoP).
<code>inv</code>	Define uma invariante, ou seja, uma restrição que deve ser sempre verdadeira.
<code>self</code>	Referência à instância atual do contexto em que a regra está sendo avaliada.
<code>implies</code>	Operador lógico: se A é verdadeiro, então B também deve ser.
<code>not</code>	Negação lógica.
<code>let [nome da variável] in [expressão lógica]</code>	Declara variáveis locais temporárias para tornar expressões mais legíveis.
<code>and</code>	Operador lógico: exige que ambas expressões sejam verdadeiras.
<code>or</code>	Operador lógico: exige que pelo menos uma expressão seja verdadeira.
<code>=</code>	Operador de igualdade entre valores e atributos.
<code><></code>	Operador de diferença entre valores.
<code>->any()</code>	Seleciona um elemento arbitrário de uma coleção com base em predicado lógico.
<code>->select()</code>	Filtra elementos de uma coleção com base em predicado lógico.
<code>->forAll()</code>	Garante que todos os elementos selecionados satisfazem a condição.
<code>->includes()</code>	Verifica se um elemento está contido em uma coleção.
<code>oclIsUndefined()</code>	Verifica se um valor é indefinido (<i>null</i>).

4.1. Passo 1: Selecionar a notação/linguagem mais adequada para especificar restrições em PoP

Neste primeiro passo, primeiramente foram definidos os critérios necessários para selecionar a notação ou linguagem mais adequada para especificar restrições em PoP. Além das restrições definidas por [Cagnin and Nakagawa 2021], observou-se a importância de incluir mais dois tipos de restrições em PoP – temporais e de privacidade, com base no referencial teórico deste trabalho.

A inclusão das restrições temporais foi motivada pela abordagem de [Ocampo-Pineda et al. 2022], que demonstra a importância de modelar e verificar limites de tempo em processos de negócios, garantindo sua viabilidade e controle durante a execução. Já as restrições de privacidade foram consideradas devido ao trabalho de [Labda et al. 2014], que destaca a necessidade de incorporar mecanismos de proteção de dados (ex.: consentimento e controle de acesso) na modelagem de processos interorganizacionais, uma abordagem de interesse para o PoP, pois permite que metaprocessos (como governança e conformidade) assegurem a proteção de dados pessoais desde o *design* até a execução.

Para a definição dos critérios de seleção, foram considerados os seguintes aspectos: capacidade da notação em especificar os tipos de restrição para o contexto do PoP, facilidade de aprendizado da notação, disponibilidade de ferramenta de apoio, capacidade de integração com ferramentas de modelagem, independente do domínio e não proprietária, e apoio na validação formal de restrições.

A Tabela 7 resume os critérios que foram definidos e que são utilizados na análise das notações encontradas na literatura e, conseqüentemente, para a escolha da mais adequada para o PoP.

Tabela 7. Critérios para análise de notações/linguagens de especificação de restrições

ID	Descrição
CS1	A notação ou linguagem permite especificar restrições de processos constituintes do PoP
CS2	A notação ou linguagem permite especificar restrições de missões do PoP
CS3	A notação ou linguagem permite especificar restrições de variabilidades do PoP
CS4	A notação ou linguagem permite especificar restrições temporais
CS5	A notação ou linguagem permite especificar restrições de privacidade
CS6	A notação ou linguagem tem suporte para especificar mudanças de restrições em tempo de execução devido ao dinamismo do PoP
CS7	A notação ou linguagem é fácil de ser aprendida
CS8	A notação ou linguagem possui ferramenta de apoio
CS9	A ferramenta de apoio da notação ou linguagem pode ser integrada à ferramentas BPMN
CS10	A notação ou linguagem é independente de domínio
CS11	A notação ou linguagem não é proprietária
CS12	A notação ou linguagem suporta validação formal de restrições

A Tabela 8 apresenta os resultados da análise da cobertura das notações ou linguagens de especificação de restrições com base nos critérios supramencionados. Para a análise, foi utilizada a seguinte escala de Likert: **A** (Atende), **NA** (Não Atende) e **AP** (Atende Parcialmente). A justificativa para o caso da linguagem de especificação que não atende ou atende parcialmente a um determinado critério é apresentada no Apêndice A.

A notação escolhida para especificar restrições em modelos PoP foi a OCL pois atendeu as características do PoP e foi a notação que atendeu (6 de 12 critérios) ou atendeu parcialmente (5 de 12 critérios) o maior número de critérios estabelecidos. A partir disso, foram elaboradas as Regras de Especificação de Restrições (REr).

4.2. Passo 2: Definir regras para especificar restrições do modelo de restrições das missões do PoP

As restrições de missões (ou *mission constraints*) são regras que estabelecem relações de dependência, exclusividade, ordem ou condições contextuais entre missões de um PoP. Elas asseguram que a execução de missões esteja alinhada aos objetivos estratégicos da aliança organizacional, garantindo coerência, controle e previsibilidade no comportamento emergente do PoP. Para isso, recomenda-se a adição do atributo *isActive* na metaclassa *PoPMission* do metamodelo do PoP.

REr1: Definir o contexto da especificação

Inicie a especificação das restrições referenciando o modelo de missões do PoP correspondente. Deve ser feita antes de qualquer restrição, ou seja, em todos os demais passos (3 a 6).

Especificação em OCL:

context [NomeModeloMissoes]

Tabela 8. Resultado da análise das notações/linguagens de especificação de restrições

Critérios	OCL	DECLARE	LTLF	SIGNAL	CSTNU	SWRL	EssentialOCL
CS1	A	A	AP	AP	AP	A	A
CS2	AP	NA	AP	NA	NA	AP	AP
CS3	AP	NA	NA	NA	NA	NA	NA
CS4	AP	A	A	A	A	NA	AP
CS5	AP	NA	NA	NA	NA	A	AP
CS6	NA	NA	NA	NA	NA	NA	NA
CS7	AP	AP	NA	NA	NA	AP	AP
CS8	A	A	A	A	A	A	A
CS9	A	AP	NA	NA	AP	AP	A
CS10	A	A	A	A	A	AP	A
CS11	A	A	A	A	A	A	A
CS12	A	AP	A	A	A	AP	AP

Exemplo no PoP Saúde:

context ModeloMissoesPoPSaude

REr2: Definir uma restrição para cada relação de dependência entre duas missões

Cada restrição deve ser definida como um `inv`, contendo a lógica da relação de dependência da Missao2 em relação à Missao1, e um identificador. Essa lógica deve garantir que a Missao1 foi definida e finalizada para que a Missao2 seja ativada. Cada `inv` deve ser atômico para facilitar validação e melhoria da especificação.

Especificação em OCL:

```
inv Mission_[nomeDaMissao]_Constraint[id]:
  let m1 = self.missions->any(m | m.name = '[nomeDaMissao1]'),
      m2 = self.missions->any(m | m.name = '[nomeDaMissao2]')
  in
  not m1.oclIsUndefined() and not m2.oclIsUndefined() and m1.hasFinished implies
  m2.isActive
```

Exemplo no PoP Saúde: M3 (Reduzir o tempo de espera do paciente para transferência para exames e procedimentos) requer M1 (Reduzir o tempo de espera do paciente para hospitalização for concluída).

```
inv Mission_M3_Constraint01:
  let m3 = self.missions->any(m1 | m1.name = 'M3'),
      m1 = self.missions->any(m2 | m2.name = 'M1')
  in
  not m3.oclIsUndefined() and not m1.oclIsUndefined() and m3.hasFinished implies
  m1.isActive
```

REr3: Definir uma restrição para cada relação exclusiva entre duas missões (ou seja, se uma missão estiver ativa, outra deve estar inativa)

Cada restrição deve ser definida como um `inv`, contendo a lógica da relação de exclusão da Missao2 em relação à Missao1, e um identificador. Essa lógica deve garantir que a Missao1 foi

definida e está ativa, e a Missao2 não pode ser executada ao mesmo tempo.

Observação: Para uma relação de mútua exclusão, deve-se fazer outra restrição, onde apenas Missao1 e Missao2 trocarão de lugar na lógica da sintaxe.

Especificação em OCL:

```
inv Mission_[nomeDaMissao]_Constraint[identificador]:  
  let m1 = self.missions->any(m | m.name = '[nomeDaMissao1]'),  
      m2 = self.missions->any(m | m.name = '[nomeDaMissao2]')  
  in  
  not m1.oclIsUndefined() and not m2.oclIsUndefined() and m1.isActive implies  
  not m2.isActive
```

Exemplo no PoP Saúde: M4 (Receber informações adicionais do paciente) exclui M5 (Definir a urgência, a especialidade médica necessária e o estado do paciente).

```
inv Mission_M4_Constraint01:  
  let m4 = self.missions->any(m4 | m4.name = 'M4'),  
      m5 = self.missions->any(m5 | m5.name = 'M5')  
  in  
  not m4.oclIsUndefined() and not m5.oclIsUndefined() and m4.isActive implies  
  not m5.isActive
```

RER4: Definir uma restrição para cada relação de concorrência entre duas missões (ou seja, as duas missões devem ser alcançadas em paralelo)

Cada restrição deve ser definida como um `inv`, contendo a lógica da relação de concorrência da Missao1 e Missao2, e um identificador. Essa lógica deve garantir que ambas missões foram definidas e estão ativas.

Especificação em OCL:

```
inv Mission_[nomeDaMissao]_Constraint[identificador]:  
  let m1 = self.missions->any(m | m.name = '[nomeDaMissao1]'),  
      m2 = self.missions->any(m | m.name = '[nomeDaMissao2]')  
  in  
  not m1.oclIsUndefined() and not m2.oclIsUndefined() implies  
  (m1.isActive = m2.isActive)
```

Exemplo no PoP Saúde: M6 (Comunicar o paciente sobre o atendimento) e M7 (Acionar a ambulância adequada) em paralelo.

```
inv Mission_M6_Constraint01:  
  let m6 = self.missions->any(m | m.name = 'M6'),  
      m7 = self.missions->any(m | m.name = 'M7')  
  in  
  not m6.oclIsUndefined() and not m7.oclIsUndefined() implies  
  (m6.isActive = m7.isActive)
```

4.3. Passo 3: Definir regras para especificar restrições do modelo de restrições dos Processos Constituintes do PoP

As sintaxes para especificar as restrições do modelo de restrições dos Processos Constituintes do PoP são similares àquelas definidas para o modelo de restrições das missões do PoP. Para facilitar o entendimento, a seguir, é apresentado um exemplo de uso de RER2 no contexto de processos constituintes do PoP Saúde.

Exemplo no PoP Saúde – Dependência entre processos (REr2):

O processo Central de Regulação de Leitos (RCU) só pode executar a busca por leitos especializados se o processo Serviço de Compartilhamento de Estado de Leitos (HBS) estiver ativo/disponível; ou seja, RCU requer HBS.

```
inv Process_RCU_Constraint01:
  self.processes->any(cp1 | cp1.name = 'RCU'
    and not cp1.oclIsUndefined() and cp1.hasFinished) implies
  self.processes->any(cp2 | cp2.name = 'HBS'
    and not cp2.oclIsUndefined() and cp2.isActive)
```

4.4. Passo 4: Definir regras para especificar restrições do modelo de restrições de variabilidade do PoP

As restrições de variabilidade têm o propósito de garantir que a seleção de variantes em pontos de variação seja consistente com as dependências operacionais do PoP, evitando combinações inválidas e assegurando coerência na configuração do processo. Considerando as relações de variabilidade definidas no M-PoP, este trabalho estabelece regras apenas para dependência e exclusão entre variantes, que são os tipos efetivamente identificados no modelo. Embora existam relações entre variantes e processos constituintes, para fins de clareza e foco, esta seção trata exclusivamente das restrições aplicadas às variantes. Para isso, recomenda-se a atualização do metamodelo do PoP, adicionando o atributo `isSelected` na metaclassa `Variant`.

REr5: Definir uma restrição para realizar a seleção mútua de duas variantes

Cada restrição deve ser definida como `inv`, contendo a lógica que especifica que se a variante1 está definida e foi selecionada, a variante2 também deve ser selecionada.

Especificação em OCL:

```
inv Variability_[Var1]_Constraint[identificador]:
  let v1 = self.variants->any(v | v.name = '[nomeDaVariante1]'),
  v2 = self.variants->any(v | v.name = '[nomeDaVariante2]')
  in
  not v1.oclIsUndefined() and v1.isSelected implies
  (not v2.oclIsUndefined() and v2.isSelected)
```

Exemplo no PoP Saúde: A variante V1 (Buscar automaticamente leitos disponíveis em hospitais especializados) deve ser selecionada somente se a variante V2 (Buscar leitos automaticamente) também for selecionada. Para buscar automaticamente leitos em hospitais especializados, é necessário garantir que a busca seja realizada de maneira automática.

```
inv Variability_V1_Constraint_01:
  let v1 = self.variants->any(v1 | v1.name = 'SearchBedAutoSpec'),
  v2 = self.variants->any(v2 | v2.name = 'SearchBedAuto')
  in
  not v1.oclIsUndefined() and v1.isSelected implies
  v2.isSelected
```

REr6: Definir restrição para especificar exclusão entre variantes (quando uma é selecionada, a outra não pode ser selecionada)

Quando duas variantes pertencem ao mesmo ponto de variação e mantêm uma relação de exclusão (XOR), é necessário definir restrições que garantam que a seleção de uma variante impede a seleção da

outra. Esta regra assegura que o modelo de variabilidade seja consistente e que apenas uma alternativa seja ativada durante a execução do processo constituinte.

Cada restrição deve ser definida como um `inv`, contendo a lógica que impede a seleção mútua das variantes. A regra deve ser atômica e verificar explicitamente que, se uma variante estiver selecionada, a outra obrigatoriamente não pode estar selecionada.

Especificação em OCL:

```
inv Variability_[Var1]_Exclusive_[Var2]_Constraint[identificador]:
  let v1 = self.variants->any(v | v.name = '[Variante1]'),
      v2 = self.variants->any(v | v.name = '[Variante2]')
  in
  not v1.oclIsUndefined() and not v2.oclIsUndefined() implies
    not (v1.isSelected and v2.isSelected)
```

Exemplo no PoP Saúde:

O ponto de variação `ProcurarLeitos` possui as variantes: V1 (Buscar automaticamente leitos disponíveis em hospitais especializados) e V3 (buscar leitos Manualmente). Elas possuem uma relação XOR (uma exclui a outra).

```
inv Variability_SearchBedAutoSpec_Constraint01:
  let v3 = self.variants->any(v3 | v3.name = 'SearchBedAutoSpec'),
      v4 = self.variants->any(v4 | v4.name = 'SearchBedManual')
  in
  not v3.oclIsUndefined() and not v4.oclIsUndefined() implies
    not (v3.isSelected and v4.isSelected)
```

4.5. Passo 5: Definir regras para especificar restrições temporais no PoP

As restrições temporais regulam limites de tempo relacionados à execução das missões do PoP, permitindo controlar prazos, intervalos e sequenciamento temporal entre atividades ou estados das missões. Esse tipo de restrição está associado ao comportamento dinâmico do PoP ao longo do tempo e sua aplicação foi realizada no modelo de restrições missões do PoP, mas pode ser usado também no contexto dos outros modelos de restrições do PoP.

Para especificar as restrições temporais do PoP, recomenda-se adicionar os seguintes atributos `startTime`, `endTime`, `hasStarted` e `hasFinished` nas metaclasses `PoPMission`, `ConstituentProcess`, `Variant` e `Activity` do metamodelo do PoP.

REr7: Definir uma restrição temporal quando há prazos entre a finalização de uma missão e o início da outra

Cada restrição deve ser definida como um `inv`, contendo a lógica que estabelece o tempo máximo permitido entre o término de uma missão e o início da seguinte. Cada `inv` deve ser atômico, de forma a facilitar a validação e a evolução da especificação. Essa lógica deve garantir que a primeira missão foi concluída e que a segunda missão se inicie dentro do prazo estabelecido. (X = Fração de horas)

Especificação em OCL:

```
inv Mission_[nomeDaMissao]_Constraint[identificador]:
  let m1 = self.missions->any(m | m.name = '[nomeDaMissao1]'),
      m2 = self.missions->any(m | m.name = '[nomeDaMissao2]')
  in
  not m1.oclIsUndefined() and not m2.oclIsUndefined()
  and m1.hasFinished implies
    m2.startTime <= m1.endTime + X
```

Exemplo no PoP Saúde: M3 (Reduzir o tempo de espera para transferência para exames e procedimentos) deve ser seguida por M1 (Reduzir o tempo de espera para hospitalização) em até 15 minutos.

```
inv Mission_M3_Constraint02:
  let m3 = self.missions->any(m3 | m3.name = 'M3'),
      m1 = self.missions->any(m1 | m1.name = 'M1')
  in
  not m3.ocIsUndefined() and not m1.ocIsUndefined() and m3.hasFinished implies
  m1.startTime <= m3.endTime + 0.0104 -- 0.0104 = 15 minutos
```

REr8: Definir uma restrição temporal quando uma missão tem um prazo pré-definido para iniciar a sua execução

Cada restrição deve ser definida como um *inv*, contendo a lógica que estabelece o tempo máximo permitido entre o início do processo (ou outro marco de referência definido) e o início da missão em questão. Essa lógica deve garantir que a missão não esteja indefinida e que seu início ocorra dentro do prazo estabelecido (X = Fração de horas).

Especificação em OCL:

```
inv Mission_[nomeDaMissao]_Constraint[identificador]:
  let m = self.missions->any(m | m.name = '[nomeDaMissao]')
  in
  not m.ocIsUndefined() implies
  m.startTime <= self.startTime + X
```

Exemplo no PoP Saúde: A missão M2 (Reduzir o tempo de espera para atendimento na Unidade Básica de Saúde) deve iniciar em até 1 hora após o início do processo PoP.

```
inv Mission_M2_Constraint01:
  let m2 = self.missions->any(m2 | m2.name = 'M2')
  in
  not m2.ocIsUndefined() implies
  m2.startTime <= self.startTime + 1.0 -- (1h)
```

REr9: Definir uma restrição temporal quando uma missão tem um prazo pré-definido para finalizar a sua execução

Cada restrição deve ser definida como um *inv*, contendo a lógica que estabelece o tempo máximo permitido entre o início (ou outro marco de referência definido) e o término da missão. Cada *inv* deve ser atômico para facilitar a validação e a evolução da especificação. Essa lógica deve garantir que a missão não esteja indefinida, e que seu término ocorra dentro do prazo estabelecido. (X = Fração de horas)

Especificação em OCL:

```
inv Mission_[nomeDaMissao]_Constraint[identificador]:
  let m = self.missions->any(m | m.name = '[nomeDaMissao]')
  in
  not m.ocIsUndefined() and m.hasStarted implies
  m.endTime <= m.startTime + X
```

Exemplo no PoP Saúde: Missão M8 (Atendimento Hospitalar) deve finalizar em até 3 horas após o seu início.

```

inv Mission_M8_Constraint01:
  let m = self.missions->any(m | m.name = 'M4')
  in
  not m.oclIsUndefined() and m.hasStarted implies
  m.endTime <= m.startTime + 3.0 -- 3h

```

Salienta-se que as regras de especificações temporais apresentadas também pode ser utilizadas no contexto de atividades e variantes do Modelo Detalhado de Missão do PoP.

4.6. Passo 6: Definir regras para especificar restrições de privacidade no modelo detalhado de missão do PoP

As restrições de privacidade foram definidas apenas no Modelo Detalhado de Missão do PoP porque após analisar os tipos de restrições de privacidade descritas em [Labda et al. 2014], esse modelo apresenta a granularidade necessária para representar os pontos em que dados sensíveis são criados, consultados, modificados ou compartilhados. Diferentemente dos demais modelos do M-PoP (restrições de missões, restrições de processos constituintes e restrições de variabilidade), o Modelo Detalhado de Missão explicita:

- atividades que manipulam dados sensíveis;
- atores envolvidos;
- artefatos de dados manipulados;
- fluxos de entrada e saída;
- contexto operacional em que os dados são tratados.

Premissa: Aplica-se à atividades que manipulam diretamente artefatos de dados sensíveis (por exemplo, RegistroPaciente, CompartilharDadosPaciente, AtualizarRegistroPaciente) no Modelo Detalhado de Missão do PoP. Para isso, este trabalho recomenda a atualização do meta-modelo do PoP, adicionando os atributos `isSensitive`, `hasConsent`, `isAnonymized` e `listRolesAccess` na metaclasses `DataObject`.

REr10: Definir restrição de consentimento para uso de dados

Esta restrição garante que atividades que criem, acessem ou atualizem dados sensíveis só possam ser executadas se o proprietário do dado tiver concedido *consentimento explícito* (`hasConsent = true`). Ela atua como pré-condição base para todas as demais regras de privacidade (como *Necessity to Know*, Acesso Autorizado e Anonimização). Cada restrição deve ser definida como um **inv**, contendo a lógica que identifica se a atividade tem dados sensíveis associados e os seleciona. Uma vez que esses objetos são selecionados, garante-se que todos esses dados possuem consentimento explícito antes de serem utilizados.

Especificação em OCL:

```

inv Constraint_Consentimento_[Activity]:
  let act = self.activities->any(a | a.name = '[Activity]')
  in
  not act.oclIsUndefined() implies
  act.data->select(d | d.name = '[dataObject]' and d.isSensitive = true)
  ->forall(d | d.hasConsent = true)

```

Exemplo no PoP Saúde Pública: Durante o registro e a atualização de dados do paciente, o consentimento explícito deve estar associado ao paciente cujos dados estão sendo manipulados.

```
inv Restricao_Consentimento_RegistrarOuAtualizarRegistroPaciente:
  let act = self.activities->any(a |
    a.name = 'RegistrarDadosPaciente' or
    a.name = 'AtualizarRegistroPaciente')
  in
  not act.oclIsUndefined() implies
  act.data->select(d | d.name = 'RegistroPaciente'
    and d.isSensitive = true)
    ->forall(d.hasConsent = true)
```

REr11: Definir restrição *Necessity to Know* (NtK) ou controle de acesso

Uma restrição *necessity to know* é um tipo de restrição de privacidade e consiste em garantir que cada ator acesse apenas as informações estritamente necessárias ao desempenho de sua função, evitando o acesso desnecessário a dados restritos [Labda et al. 2014]. Esse tipo de restrição é importante para o contexto do PoP, principalmente em domínios que tratam dados sensíveis.

Premissa: Esta restrição é aplicável em atividades do Modelo Detalhado de Missão do PoP que consultam ou manipulam (registro ou atualização) objeto de dados com dados sensíveis. Essa restrição deve sempre ser precedida da restrição relacionada ao consentimento para uso dos dados (REr10).

Cada restrição deve ser definida como um *inv*, contendo a lógica que verifica se a atividade do modelo de missão detalhada possui objetos de dados sensíveis associados, se possuir, então analisa se o usuário deu consentimento de uso e se o papel (*role*) que está tentando acessar/manipular aquele dado está presente na lista de papéis com permissão. Cada objeto de dado deve ser verificado separadamente (por meio de uma restrição distinta) se suas informações são sensíveis.

Especificação em OCL:

```
inv Restricao_NtK_[Atividade]:
  let act = self.activities->any(a | a.name = '[Atividade]')
  in
  not act.oclIsUndefined() implies
  act.data->select(d | d.name = '[DataObject]'
    and d.isSensitive = true and d.hasConsent = true)
    ->forall(d | d.listRolesAccess->includes(act.Role))
```

Exemplo no PoP Saúde Pública: Durante a atividade *AtualizarRegistroPaciente*, o médico atualiza o artefato *RegistroPaciente* com novas informações após definir urgência, especialidade e estado do paciente. O médico deve acessar somente as informações necessárias para o diagnóstico e encaminhamento, e apenas se o paciente tiver concedido consentimento.

```
inv Restricao_NtK_AtualizarRegistroPaciente:
  let act = self.activities->any(a | a.name = 'AtualizarRegistroPaciente')
  in
  not act.oclIsUndefined() implies
  act.data->select(d | d.name = 'RegistroPaciente'
    and d.isSensitive = true and d.hasConsent = true)
    ->forall(d | d.listRolesAccess->includes(act.Role))
```

REr12: Definir restrição de anonimização para análise de dados

Cada restrição deve ser definida como *inv*, assegurando a lógica de que a atividade que está sendo realizada terá acesso aos dados utilizados somente se foram previamente anonimizados, impedindo

a identificação do seu proprietário. Aplica-se exclusivamente às atividades que manipulam dados derivados ou consolidados (por exemplo, geração de consultas ou relatórios e monitoramento de indicadores). Nesses contextos, mesmo que os dados de origem sejam sensíveis, as versões utilizadas nessas atividades devem estar anonimizadas antes do processamento, assegurando conformidade com diretrizes éticas, legais (como a Lei Geral de Proteção de Dados - LGPD) e operacionais.

Especificação em OCL:

```
inv Restricao_Anonimizacao_[Atividade]_[Objeto]:
  let act = self.activities->any(a | a.name = '[Atividade]')
  in
  not act.oclIsUndefined() implies
    act.data->select(d | d.name = '[NomeDoObjeto]' and d.isSensitive = true)
    ->forall(d | d.isAnonymized = true)
```

Exemplo no PoP Saúde: Durante a atividade AnalisarDadosEpidemiologicos, o sistema utiliza dados sensíveis derivados dos registros de pacientes. Esses dados devem obrigatoriamente estar anonimizados antes da análise.

```
inv Restricao_Anonimizacao_AnaliseEpidemiologica_RegistroPaciente:
  let act = self.activities->any(a | a.name = 'AnalisarDadosEpidemiologicos')
  in
  not act.oclIsUndefined() implies
    act.data->select(d | d.name = 'RegistroPaciente'
      and d.isSensitive = true)
    ->forall(d | d.isAnonymized = true)
```

Recomendações gerais para as especificações de restrições:

- Os nomes das restrições devem indicar claramente os processos, missões, atividades ou variantes envolvidas.
- O uso de `let` é opcional, mas facilita a leitura e manutenção de restrições mais complexas.
- Recomenda-se manter cada restrição como um **inv** atômico para permitir validação incremental.
- A ordem de declaração das restrições deve seguir a ordem natural de leitura do modelo (dependências, exclusões, concorrências, temporais e privacidade).
- Quando uma mesma missão, processo constituinte ou variante possuir múltiplas restrições associadas, o identificador da restrição (*constraint*) deve ser incrementado de forma sequencial. Por exemplo, a missão M3 aparece tanto em REr2 quanto em REr7, resultando em identificadores distintos para cada restrição (por exemplo, Constraint01, Constraint02, e assim por diante).

4.7. Passo 7: Atualizar o metamodelo do PoP

A Tabela 9 apresenta os atributos criados em cinco classes do metamodelo do PoP para apoiar a especificação das restrições, conforme indicados nos Passos 2, 4, 5 e 6. Consequentemente, o metamodelo foi atualizado e está ilustrado na Figura 6. As metaclasses atualizadas neste trabalho estão destacadas em cinza.

4.8. Passo 8: Validar a especificação de cada regra definida

Para a validação de cada especificação de restrição definida nos passos 2 a 6, utilizou-se as ferramentas EclipseOCL e USE escolhidas por oferecerem maior facilidade de acesso, suporte à instalação e simplicidade de uso. Considerando que a ferramenta USE é antiga e não recebe atualizações, foi empregada a versão 4.2.0. No caso do EclipseOCL, foi necessário instalar uma versão recente do Eclipse *Modeling Tools*, tendo sido utilizada, neste trabalho, a versão 2024-06-R.

Tabela 9. Atualização do metamodelo do PoP

Classes	Atributos adicionales
PoPMission, ConstituentProcess	isActive, hasStarted, hasFinished, startTime, endTime
Activity	hasStarted, hasFinished, startTime, endTime
Variant	isSelected, hasStarted, hasFinished, startTime, endTime
DataObject	istRolesAccess, isSensitive, hasConsent, isAnonymized

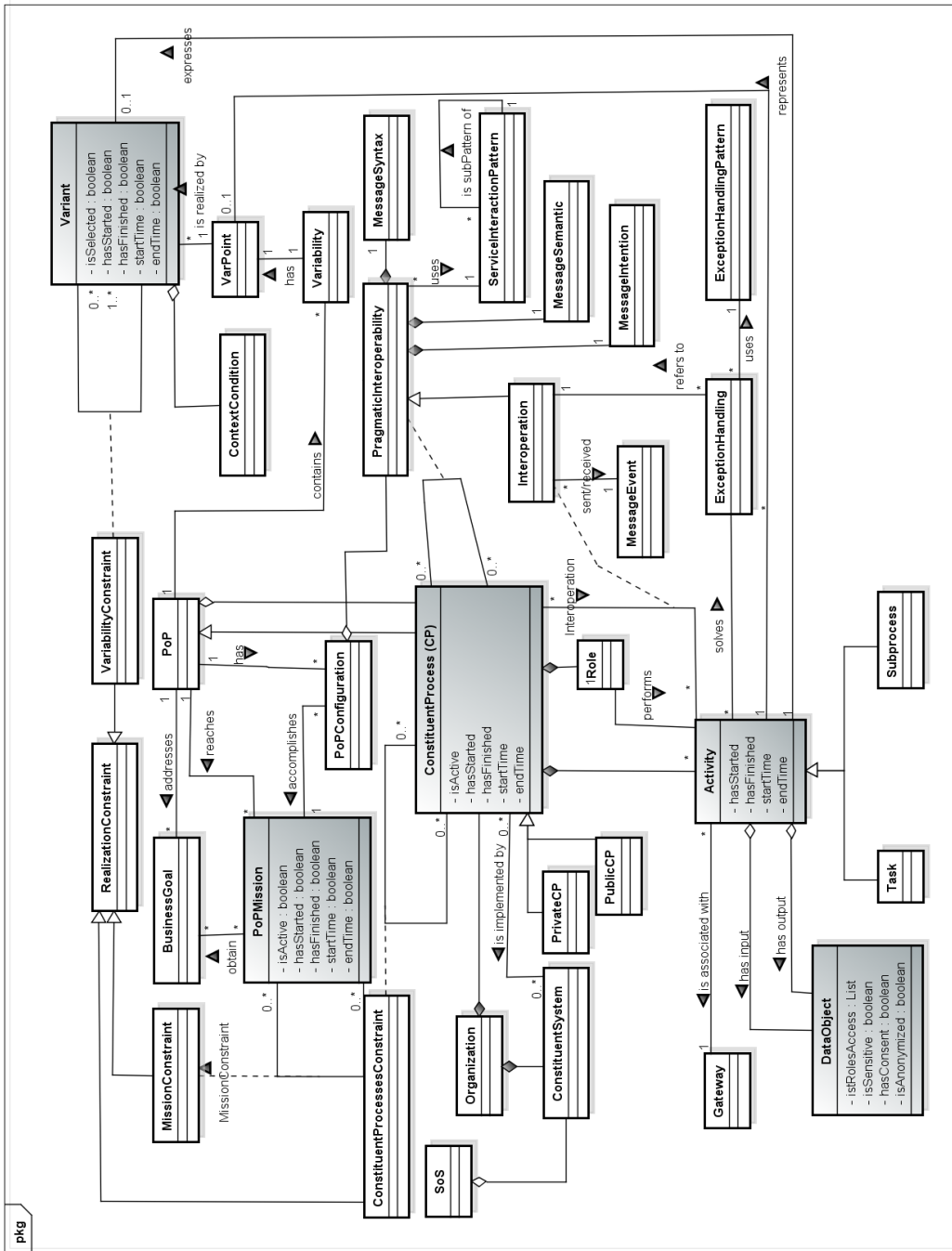


Figura 6. PoP metamodel (adaptado de [Cagnin and Nakagawa 2022])

EclipseOCL

Após a instalação da ferramenta EclipseOCL, foi elaborado o modelo *Ecore*⁶ com classes e atributos representando os elementos dos modelos do PoP Saúde, conforme ilustrado na Figura 7.

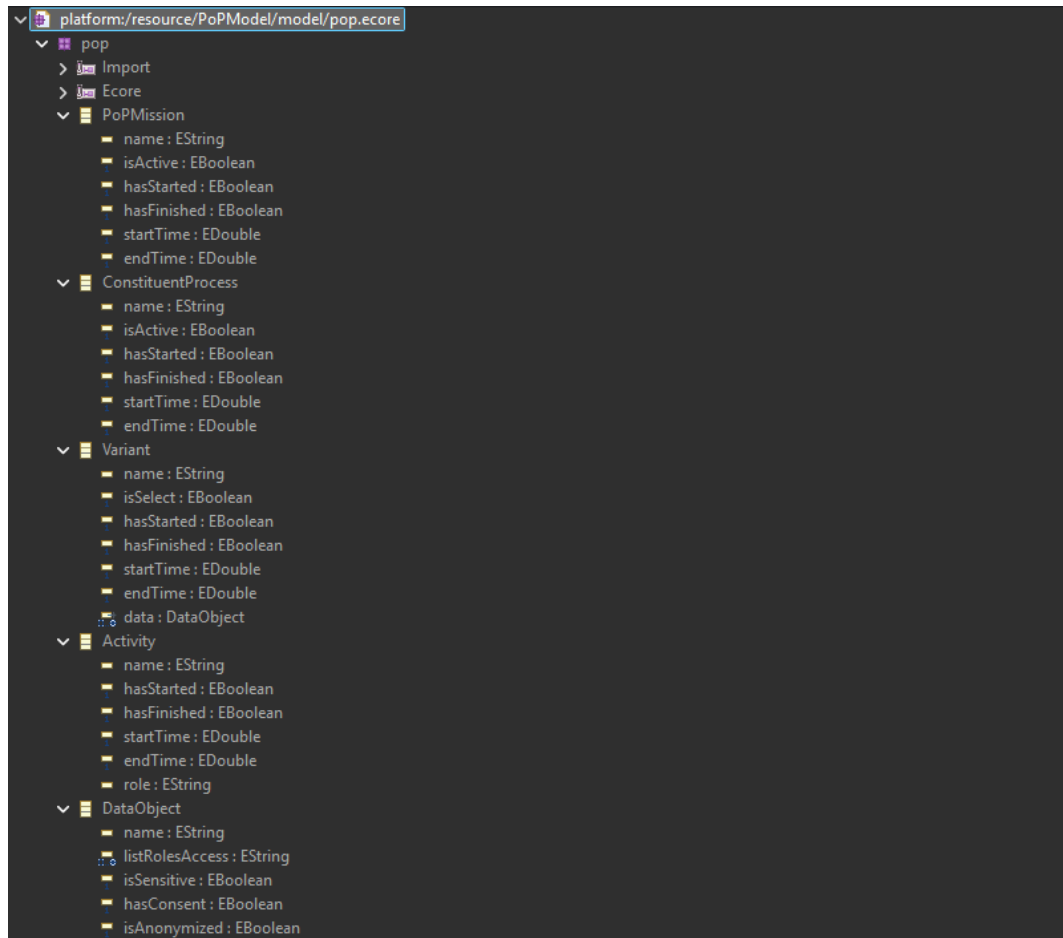


Figura 7. Modelo *Ecore* do PoP Saúde

Posteriormente, as especificações de restrição do PoP Saúde, apresentadas nos passos 2 a 6, foram criadas em um arquivo *ocl*⁷. Esse arquivo construído por meio do *CompleteOCL Editor*, um editor próprio do EclipseOCL voltado para a construção e validação de regras OCL externas ao metamodelo. Essa abordagem mostrou-se adequada, pois permite definir um conjunto de restrições sobre um metamodelo e validá-las de forma independente. A Figura 8 apresenta as especificações de restrições referentes ao passo 2, já devidamente validadas e nenhum erro foi identificado. Similarmente, as especificações de restrição dos passos 3 a 6 também foram validadas sem identificação de erro de sintaxe.

⁶Um Ecore é um arquivo (extensão .ecore) que define classes, atributos, relacionamentos, tipos, pacotes e restrições básicas. Embora semelhante a um diagrama de classes, é estruturado de forma que o Eclipse o reconheça como um metamodelo executável.

⁷Arquivo externo utilizado para especificar restrições em OCL sem a necessidade de alterar o modelo Ecore.

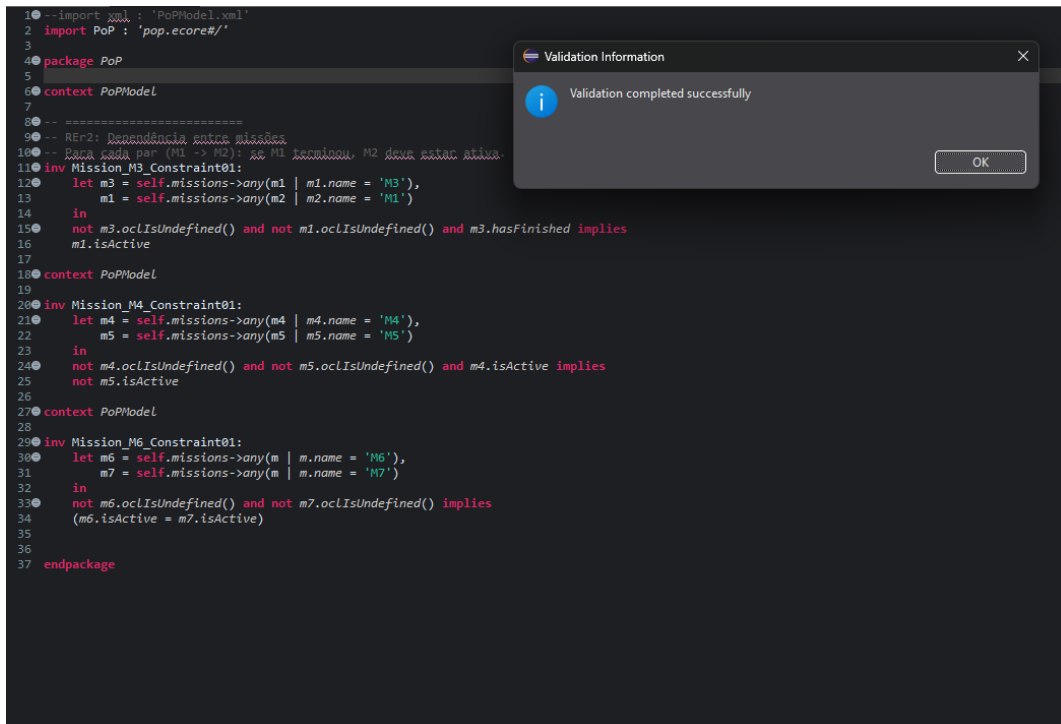


Figura 8. Validação com EclipseOCL das especificação das restrições do PoP Saúde - Passo 2

USE: UML-based Specification Environment

Após a instalação da ferramenta USE, elaborou-se um arquivo *use*⁸ contendo as classes, atributos e associações necessárias para representar o PoP Saúde, conforme ilustrado na Figura 9.

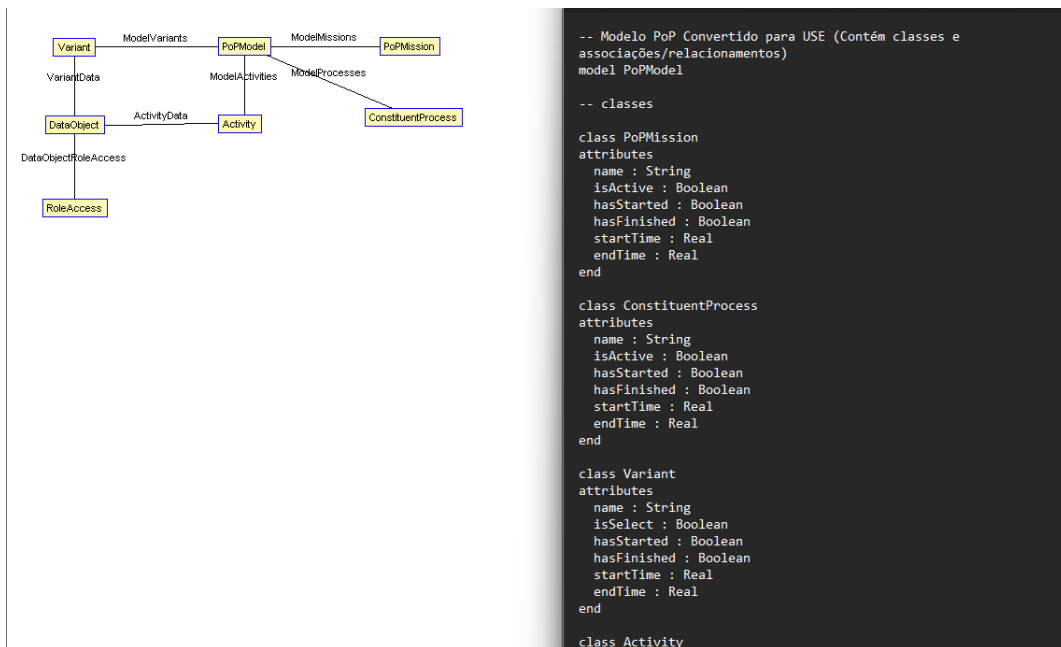


Figura 9. Modelo *USE* do PoP Saúde

Na etapa seguinte, foi criado um arquivo *soil*⁹ com todos os objetos necessários para validar as regras definidas nos passos de 2 a 6. A Figura 10 apresenta o diagrama dos objetos gerados.

⁸Formato textual utilizado pela ferramenta USE para especificar modelos UML enriquecidos com restrições OCL, incluindo classes, atributos, associações e invariantes. É o tipo de arquivo carregado pela ferramenta para permitir a validação formal das regras.

⁹script utilizado pela ferramenta USE para construir um estado do modelo, permitindo criar objetos, atribuir valores e inserir ligações em associações. É usado para realizar testes e validar invariantes OCL sobre instâncias do modelo.

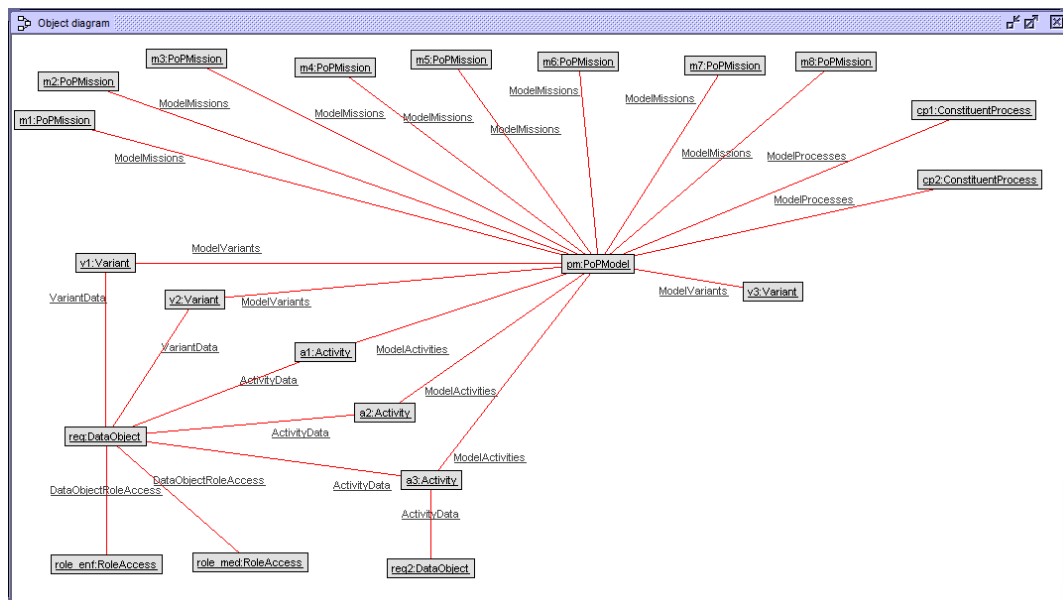


Figura 10. Instâncias de objetos criados para validação das especificações do passos 2 a 6 com a ferramenta USE

Por fim, elaborou-se um arquivo *use* contendo todas as *constraints* definidas nos passos 2 a 6. Como os objetos já haviam sido criados e possuíam valores atribuídos, a validação pôde ser realizada imediatamente. Para fins ilustrativos, empregou-se a regra REr11 (Definir restrição Necessity to Know (NtK) ou controle de acesso) com o objetivo de demonstrar os casos nos quais a restrição é satisfeita ou violada. A Figura 11 apresenta as restrições definidas e seus respectivos resultados de validação. Todas as especificações de restrições definida neste trabalho também foram validadas com o apoio da ferramenta USE e nenhum erro de sintaxe foi detectado.

Class invariants					Object properties	
Invariant	Loaded	Active	Negate	Satisfied	a2	
PoPModel: Mission_M2_Constraint01	☒	☒	☐	true	name : String	'AtualizarRegistroPaciente'
PoPModel: Mission_M3_Constraint01	☒	☒	☐	true	hasStarted : Boolean	true
PoPModel: Mission_M3_Constraint02	☒	☒	☐	true	hasFinished : Boolean	false
PoPModel: Mission_M4_Constraint01	☒	☒	☐	true	startTime : Real	2.0
PoPModel: Mission_M6_Constraint01	☒	☒	☐	true	endTime : Real	2.5
PoPModel: Mission_M8_Constraint01	☒	☒	☐	true	role : String	'Medico'
PoPModel: Process_RCU_Constraint01	☒	☒	☐	true		
PoPModel: Restricao_Anonimizacao_AnaliseEpidemiologica_RegistroPaciente	☒	☒	☐	true		
PoPModel: Restricao_Consentimento_RegistrarOuAtualizarRegistroPaciente	☒	☒	☐	true		
PoPModel: Restricao_NIK_AtualizarRegistroPaciente	☒	☒	☐	true		
PoPModel: Variability_SearchBedAutoSpec_Constraint01	☒	☒	☐	true		
PoPModel: Variability_V1_Constraint_01	☒	☒	☐	true		
Constraints ok. (1ms)					Apply Reset	
Class invariants					Object properties	
Invariant	Loaded	Active	Negate	Satisfied	a2	
PoPModel: Mission_M2_Constraint01	☒	☒	☐	true	name : String	'AtualizarRegistroPaciente'
PoPModel: Mission_M3_Constraint01	☒	☒	☐	true	hasStarted : Boolean	true
PoPModel: Mission_M3_Constraint02	☒	☒	☐	true	hasFinished : Boolean	false
PoPModel: Mission_M4_Constraint01	☒	☒	☐	true	startTime : Real	2.0
PoPModel: Mission_M6_Constraint01	☒	☒	☐	true	endTime : Real	2.5
PoPModel: Mission_M8_Constraint01	☒	☒	☐	true	role : String	'Recepcionista'
PoPModel: Process_RCU_Constraint01	☒	☒	☐	true		
PoPModel: Restricao_Anonimizacao_AnaliseEpidemiologica_RegistroPaciente	☒	☒	☐	true		
PoPModel: Restricao_Consentimento_RegistrarOuAtualizarRegistroPaciente	☒	☒	☐	true		
PoPModel: Restricao_NIK_AtualizarRegistroPaciente	☒	☒	☐	false		
PoPModel: Variability_SearchBedAutoSpec_Constraint01	☒	☒	☐	true		
PoPModel: Variability_V1_Constraint_01	☒	☒	☐	true		
1 constraint failed. (2ms)					Apply Reset	

Figura 11. Resultado da validação com a ferramenta USE das especificações de restrições do PoP Saúde

5. Conclusão

Este trabalho apresentou um conjunto de regras para apoiar a especificação de restrições em PoP. Como principais contribuições deste trabalho destacam-se: (i) levantamento sistemático de tipos de restrições em BPMN, permitindo compreender como diferentes estudos tratam dependências, temporalidades, variabilidades, responsabilidades e requisitos de privacidade; (ii) análise comparativa, baseada em critérios bem definidos, de linguagens e notações de restrições utilizadas em modelos BPMN, culminando na seleção da OCL como a mais adequada para representar restrições no PoP; (iii) definição de um conjunto estruturado de regras para facilitar a especificação de restrições nos três modelos do M-PoP — missões, processos constituintes e variabilidade — além de regras voltadas para restrições temporais e de privacidade; (iv) aplicação prática das regras definidas em um PoP complexo e real, o PoP de Saúde Pública, demonstrando sua viabilidade e clareza; e (v) atualização do metamodelo do PoP para se adaptar às regras de restrições definidas.

Além disso, em [de Mattia et al. 2024], foi desenvolvida a *PoP Modeler*, que é uma ferramenta web para viabilizar a prática do método M-PoP [Cagnin and Nakagawa 2022], apoiando a gestão e a modelagem de PoP. No entanto, essa ferramenta ainda não oferece suporte para a especificação formal de restrições. Embora permita criar modelos de restrições nas três visões do PoP, ela não verifica se as restrições estão escritas corretamente em uma linguagem formal, como OCL, nem possui um mecanismo para validar conflitos ou inconsistências entre as restrições definidas. Dessa forma, o presente trabalho contribui para suprir parcialmente essa lacuna, oferecendo regras formais que podem orientar futuras melhorias na ferramenta *PoP Modeler*.

Algumas limitações também foram identificadas e contribuirão para a evolução do trabalho realizado. A aplicação das regras ocorreu em apenas um PoP, o que restringe a generalização dos resultados para outros domínios organizacionais. Além disso, a ausência de um ambiente de suporte para a edição e validação das expressões OCL pode levar à formulação de restrições sintaticamente incorretas ou inconsistentes, o que reforça a necessidade de evolução da ferramenta *PoP Modeler* com base nas regras de especificação de restrição propostas.

Com base nos resultados obtidos, algumas direções de trabalhos futuros são recomendadas para ampliar e fortalecer esta linha de pesquisa: (i) conduzir estudos de caso em PoP reais de diferentes domínios, permitindo avaliar a aplicabilidade, clareza e completude das regras de especificação de restrição em diferentes cenários; (ii) evoluir a ferramenta *PoP Modeler* para auxiliar a inserção, validação sintática e verificação automática das restrições em OCL, com base nas regras definidas neste trabalho; (iii) investigar mecanismos para tratar restrições dinâmicas, que possam se alterar durante a execução do PoP; e (iv) explorar o uso de linguagens ou extensões que permitam representar incertezas, exceções e aspectos probabilísticos, pois em [Ocampo-Pineda et al. 2022] é destacada a importância desse tratamento em processos distribuídos e dinâmicos. Assim, incorporar essas ideias ao PoP representa um avanço significativo para aumentar sua capacidade de lidar com cenários reais mais complexos.

Espera-se que as contribuições alcançadas sirvam de base para a evolução do M-PoP e para o aprimoramento de práticas de modelagem de processos em ambientes colaborativos e corporativos, fornecendo meios mais claros, padronizados e verificáveis para lidar com restrições em processos interorganizacionais de diferentes domínios.

Referências

Arévalo Maldonado, C., Escalona Cuaresma, M. J., Ramos Román, I., and Domínguez Muñoz, M. (2016). A metamodel to integrate business processes time perspective in bpmn 2.0. *Information and Software Technology*, 77 (september 2016), 17-33.

- Bergman, A., Rebmann, A., and Kampik, T. (2023). Bpmn2constraints: breaking down bpmn diagrams into declarative process query constraints. In *21st International Conference on Business Process Management (BPM)*, pages 137–141. Aachen University.
- Cagnin, M. I. and Nakagawa, E. Y. (2021). Towards dynamic processes-of-business processes: a new understanding. *Business Process Management Journal*, 27(5):1545–1568.
- Cagnin, M. I. and Nakagawa, E. Y. (2022). M-PoP: Leveraging the systematic modeling of processes-of-business processes. *Business Process Management Journal*, 28(5/6):1412–1445.
- de Mattia, W. G., Franco, I. C. T., Costa, M. G. N., Molina, S. d. A., Paiva, D. M., and Cagnin, M. I. (2024). Pop modeler: Uma ferramenta para facilitar a modelagem e compreensão de processos-de-processos de negócio. In *VI Workshop em Modelagem e Simulação de Sistemas Intensivos em Software (MSSiS) em conjunto com o XIV Congresso Brasileiro de Software*, pages 31–40. SBC.
- Fattouch, N., Lahmar, I. B., and Boukadi, K. (2022). Towards a meta-modeling approach for an iort-aware business process. In *ECMS*, pages 29–35.
- Grefen, P. (2024). Business models, business processes and information systems: A dynamic network view. In *30th International Conference on Cooperative Information Systems*, pages 3–17. Springer.
- Hegedüs, Á., Horváth, Á., Ráth, I., Branco, M. C., and Varró, D. (2011). Quick fix generation for dsmls. In *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 17–24. IEEE.
- Kitchenham, B. and Charters, S. (2007). Guidelines for Performing Systematic Literature Reviews in Software Engineering. Technical Report 2007-001, Keele University and Durham University, UK.
- Kumar, V. and Sharma, P. (2019). *An insight into mergers and acquisitions: A growth perspective*. Springer Nature Singapore.
- Labda, W., Mehandjiev, N., and Sampaio, P. (2014). Modeling of privacy-aware business processes in bpmn to protect personal data. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 1399–1405.
- Ma, Q., Kelsen, P., and Glodt, C. (2015). A generic model decomposition technique and its application to the eclipse modeling framework. *Software & Systems Modeling*, 14:921–952.
- Object Management Group (2013). Business Process Model and Notation (BPMN). <http://www.omg.org/spec/BPMN/2.0.2/>. Acesso em: 28/02/2021.
- Object Management Group (2014). Object Constraint Language (OCL) – Version 2.4. <http://www.omg.org/spec/OCL/2.4/>. Acesso em: 21/09/2022.
- Ocampo-Pineda, M., Posenato, R., and Zerbato, F. (2022). Timeawarebpmn-js: An editor and temporal verification tool for time-aware bpmn processes. *SoftwareX*, 17:100939.

A. Apêndice A

As Tabelas 10, 11, 12, 15, 13, 14, 16 descrevem justificativas para o caso da linguagem de especificação que não atende ou atende parcialmente a um determinado critério.

Tabela 10. OCL (Object Constraint Language)

Critério	Atende Parcialmente (AP) ou Não Atende (NA)	Justificativa
CS2	AP	Pode expressar restrições estruturais relacionadas a missões, mas não oferece suporte direto para modelar regras operacionais complexas de missão em processos de negócio.
CS3	AP	Pode ser usado para definir restrições em feature models, controlando a configuração de variabilidade, mas não tem suporte nativo para modelagem dinâmica de variabilidade.
CS4	AP	Permite definir restrições temporais simples, mas não tem suporte avançado para verificações temporais.
CS5	AP	Pode expressar regras de privacidade, mas não inclui mecanismos nativos para controle de acesso dinâmico.
CS6	NA	Não suporta modificações dinâmicas das restrições em tempo de execução. Suas expressões são estáticas e não podem ser alteradas sem recompilar o modelo. (Nenhuma das linguagens avaliadas suporta modificações dinâmicas das restrições em tempo de execução)
CS7	AP	Tem uma curva de aprendizado moderada, exigindo conhecimento de UML e lógica formal.

Tabela 11. DECLARE

Critério	Atende Parcialmente (AP) ou Não Atende (NA)	Justificativa
CS2	NA	Foca em restrições de conformidade e não suporta restrições operacionais complexas.
CS3	NA	Não há suporte direto para modelagem de variabilidade.
CS6	NA	Linguagem estática, não suporta modificações dinâmicas das restrições em tempo de execução.
CS7	AP	Embora seja baseada em lógica declarativa, o que facilita a modelagem flexível, DECLARE pode ser desafiador para usuários iniciantes devido à falta de uma representação visual tão intuitiva quanto outras notações. Além disso, entender como as regras declarativas interagem e influenciam o comportamento do modelo requer prática.
CS9	AP	Não há suporte nativo para integração com ferramentas BPMN, mas existem abordagens para mapeamento manual ou uso de ferramentas externas para combinar as duas notações. Isso exige esforço adicional por parte do usuário.
CS12	AP	A validação formal é possível, mas depende da ferramenta de apoio utilizada. Não é um recurso intrínseco à notação, mas habilitado por ferramentas externas.

Tabela 12. LTLF (Finite-trace Linear Temporal Logic)

Critério	Atende Parcialmente (AP) ou Não Atende (NA)	Justificativa
CS1	AP	Expressa restrições nos processos, mas exige modificações para aplicação no BPMN.
CS2	AP	Pode ser usada no contexto de modelar objetivos de missões baseados no fluxo de tempo.
CS3	NA	não oferece suporte para restrições de privacidade.
CS6	NA	Embora a lógica temporal seja poderosa para modelar propriedades dinâmicas, a adaptação às mudanças em tempo de execução não é um recurso nativo.
CS7	NA	A lógica temporal é intrinsecamente complexa, especialmente para modeladores sem experiência em formalismos matemáticos ou computacionais. Isso limita sua acessibilidade para iniciantes.
CS9	NA	Não há suporte direto ou integração padrão com ferramentas BPMN. A adaptação ou uso conjunto exige mapeamento manual ou desenvolvimento adicional.

Tabela 13. SIGNAL

Critério	Atende Parcialmente (AP) ou Não Atende (NA)	Justificativa
CS1	AP	Embora SIGNAL seja eficaz para modelar eventos reativos e dependências de tempo, sua aplicação em processos de negócios não é direta, exigindo abstração ou adaptação do modelo.
CS2	NA	Não é possível representar missões do PoP.
CS3	NA	Não há suporte para modelagem de variabilidade.
CS5	NA	Não há mecanismos para especificação de restrições de privacidade.
CS6	NA	Não há recurso para mudança em tempo de execução.
CS7	NA	Exige conhecimento prévio de modelagem reativa, o que dificulta o aprendizado.
CS9	NA	Não tem suporte nativo para integração.

Tabela 14. CSTNU (Constraint Systems with Time, Numerical Values, and Uncertainty)

Critério	Atende Parcialmente (AP) ou Não Atende (NA)	Justificativa
CS1	AP	Pode expressar restrições, mas exige um modelo baseado em sinais e eventos contínuos, o que dificulta a aplicação direta no BPMN.
CS2	NA	Não é adequado para restrições de missões.
CS3	NA	CSTNU não lida com variabilidade.
CS5	NA	Não inclui mecanismos para restrições de privacidade.
CS6	NA	Não permite modificações dinâmicas de restrições.
CS7	NA	CSTNU exige familiaridade com redes temporais e conceitos de incerteza, o que dificulta seu aprendizado para modeladores que não têm formação em lógica formal ou áreas afins.
CS9	AP	Não possui integração nativa com ferramentas BPMN, e a adaptação depende de abordagens personalizadas ou ferramentas intermediárias.

Tabela 15. SWRL (Semantic Web Rule Language)

Critério	Atende Parcialmente (AP) ou Não Atende (NA)	Justificativa
CS2	AP	Pode expressar restrições de missões via regras semânticas, mas carece de integração nativa com BPMN.
CS3	NA	Não há suporte para variabilidade.
CS4	NA	Não tem suporte para restrições temporais.
CS6	NA	Não suporta modificações dinâmicas das restrições em tempo de execução.
CS7	AP	Requer conhecimento de OWL e lógica de regras, o que pode ser complexo para modeladores sem experiência prévia na Web Semântica.
CS9	AP	Pode ser integrado ao BPMN via anotações semânticas, mas sem suporte nativo.
CS10	AP	A criação de ontologias pode ser influenciada pelo domínio específico.
CS12	AP	A validação formal é limitada ao contexto de inferência semântica. Verificações mais complexas ou dinâmicas requerem mecanismos externos.

Tabela 16. EssentialOCL

Critério	Atende Parcialmente (AP) ou Não Atende (NA)	Justificativa
CS2	AP	Similar ao OCL, pode definir restrições sobre missões, mas não de forma operacional.
CS3	NA	Linguagem simplificada em relação ao OCL, não tem suporte para variabilidade.
CS4	AP	Pode representar restrições temporais, mas sem suporte avançado.
CS5	AP	Não inclui mecanismos para restrições de privacidade.
CS6	NA	Assim como o OCL padrão, EssentialOCL é mais adequada para especificações estáticas e não suporte para variações dinâmicas.
CS7	AP	Requer conhecimento prévio de OCL e metamodelos.
CS12	AP	Suporta verificação formal, mas depende de ferramentas externas.