

Uma Ferramenta para Modelagem de Processos de Negócios com base em Padrões de Recomendação

Arthur Chisté Lucas¹, Karin S. Komati¹, Mateus Conrad Barcellos da Costa¹

¹Programa de Pós-graduação em Computação Aplicada (PPComp)
Campus Serra do Instituto Federal do Espírito Santo (IFES)

arthur@aclucas.net, {kkomati,mcosta}@ifes.edu.br

Abstract. *The Business Process Model and Notation (BPMN) is the core OMG specification for the development of business processes models based on the imperative modeling paradigm. The creation of BPMN diagrams remains an activity-dependent on human performance and knowledge on the Application Domain. In this context, this work proposes a modeling support tool for recommendation systems based on the concept of Recommendation Pattern. This tool supports the fulfillment of recommendation personalization requirements and makes the recommendation process an integrated part of the modeling. The case study presents that each recommendation suggested is sensitive to the current state of the model. The final model is syntactically and semantically correct.*

Resumo. *A Notação de Modelagem de Processos de Negócios (BPMN) é a especificação desenvolvida pela OMG para representar processos de negócios no paradigma de modelagem imperativa. A criação de diagramas BPMN continua sendo uma atividade dependente do desempenho humano e do conhecimento do domínio da aplicação. Nesse contexto, este trabalho propõe uma ferramenta de apoio à modelagem de processos baseada no conceito de Padrão de Recomendação com o objetivo de gerar recomendações posicionais em um processo interativo integrado à atividade de modelagem. O estudo de caso mostra que cada recomendação sugerida é sensível ao estado atual do modelo. O modelo final é sintático e semanticamente correto.*

1. Introdução

Sistemas de recomendação vêm sendo propostos no domínio de processos de negócio e *Workflow* em diferentes contextos, tais como na implementação de *Workflows* Científicos [Zhang et al. 2009], no apoio à agilidade de negócios em ambientes que fazem uso intensivo de processos [Fellmann et al. 2018], em processos de configuração de linha de produto [Pereira et al. 2018], dentre outros. Sua aplicação se dá, principalmente, no apoio à construção de modelos e na configuração e alocação de recursos para a execução de processos.

Na modelagem de processos, esses sistemas têm por objetivo reduzir os esforços de modelagem, a presença de erros e apoiar a produção de modelos que estejam em conformidade com os requisitos de negócio do processo [Koschmider and Oberweis 2010]. A variabilidade e a complexidade, bem como a dependência de conhecimento tácito e a fragmentação da informação, comuns aos ambientes organizacionais, são obstáculos à

modelagem que podem ser mitigados por sistemas de recomendação [Leyh et al. 2016] capazes de determinar recomendações personalizadas de fragmentos de modelos [Fellmann et al. 2018]. Para atenderem esses requisitos, tais sistemas devem conhecer as opções de modelagem adequadas ao estado corrente do modelo bem como seus elementos faltantes.

Literatura e prática de modelagem de processos apresentam uma plethora de métodos e técnicas de representação. Dentre estas, destacam-se as notações gráficas [Andaloussi et al. 2020], como BPMN (*Business Process Model and Notation*), Redes de Petri, EPC (*Event-driven Process Chains*), DECLARE e DCR (*Dynamic Condition Response Graphs*). Recomendações para notações e ambientes aderentes ao paradigma imperativo, como, por exemplo, editores BPMN, podem ser personalizadas com base no estado do modelo [Barba et al. 2013]. O conceito de estado do modelo, utilizado neste trabalho, denota o conjunto de informações sobre a estrutura do modelo em construção. A personalização, neste caso, requer um processo iterativo no qual cada passo da modelagem neste estado seja reportado ao sistema de recomendação para que esse, por sua vez, encaminhe recomendações consistentes com o estado corrente.

Este artigo apresenta uma ferramenta baseada no conceito de “Padrão de Recomendação” [Costa and Tamzalit 2017], que permite a identificação de recomendações no formato de fragmentos BPMN a partir de restrições especificadas para um dado processo. A proposta consiste em um conjunto de algoritmos capazes de produzir recomendações de modelagem a partir de uma especificação preliminar do processo. As opções de modelagem são geradas a partir da análise dessas regras considerando o estado corrente do modelo e posições específicas do mesmo (pontos de recomendação). Para avaliar a viabilidade do sistema, o protótipo desenvolvido de um sistema de recomendação para modelagem em BPMN é aplicado a um estudo de caso.

O restante deste artigo está organizado da seguinte maneira: a Seção 2 apresenta outras publicações sobre recomendações na modelagem de processos de negócios. Na Seção 3, apresenta-se a proposta de [Costa and Tamzalit 2017], os Padrões de Recomendação na modelagem de processos, especificamente para o BPMN. Na Seção 4 são apresentados os fundamentos e algoritmos da ferramenta; e na Seção 5 é apresentado um estudo de caso do funcionamento da ferramenta; sendo a conclusão apresentada na Seção 6.

2. Trabalhos relacionados

Na Modelagem de Processos de Negócios, considera-se o auxílio de sistemas de recomendação para sugerir fragmentos e propriedades de modelo. Uma das primeiras iniciativas nessa direção é o sistema FlowRecommender, proposto por Zhang e outros [Zhang et al. 2009], baseado em proveniência de Workflow (*Workflow Provenance*). A Proveniência de Workflow consiste no registro histórico de fluxos de trabalho que foram executados e de resultados intermediários e finais desta execução. O termo Proveniência de Processos é adotado na Gerência de Processos de Negócio para referir-se à coleta sistemática das informações necessárias para reconstruir o que ocorre na execução dos processos [van der Aalst 2010].

O FlowRecommender tem como objetivo apoiar a construção de *workflows* científicos quando não há conhecimento sobre o domínio do problema ou quando os fluxos

de trabalho são construídos por modeladores com pouco conhecimento. Nessa proposta, a questão da personalização e adequação das recomendações é observada pelos autores. Estes argumentam que, nas recomendações típicas obtidas por meio de proveniência, a previsão de um nó sucessor no *workflow* em construção é dependente apenas de um de seus subcaminhos adjacentes antecessores. Nesse sentido, eles propõem um método para se determinar as recomendações que investigam a correlação entre cada nó possível para o fluxo de trabalho, juntamente com todos os seus caminhos adjacentes antecessores, e os dados de proveniência.

Iniciativas de recomendação para modelagem de processos têm se fixado no mesmo princípio dos sistemas baseados em proveniência. Isto é, tais sistemas buscam encontrar soluções aceitas, baseadas, por exemplo, em modelos de referência, repositórios de modelos e dados históricos de execução. As recomendações incluem, por exemplo, sugestões de fragmentos de modelo baseadas em padrões de processo [Dijkman et al. 2011, Wang and Wu 2011], e soluções comprovadas [Schonenberg et al. 2008, Koschmider et al. 2011, Li et al. 2014].

Embora tais propostas possam fornecer percepções úteis para a tomada de decisão na modelagem, a adoção das mesmas de forma integrada a ambientes de modelagem pode requerer a adaptação e integração das recomendações ao modelo em construção. Conforme observado por Fellmann e outros [Fellmann et al. 2018], na proposta de um catálogo de requisitos para estes sistemas, as recomendações devem ser contextualizadas. Um dos requisitos funcionais do catálogo proposto pelos autores refere-se à necessidade de recomendações personalizadas: *“O sistema deve fornecer recomendações personalizadas adaptadas às necessidades do usuário específico em seu ou sua situação específica de modelagem”* [Fellmann et al. 2018]. Há portanto uma necessidade de se sistematizar a interação entre a obtenção das recomendações e o processo de modelagem em si.

Algoritmos de recomendação definem uma metodologia de auxílio ao modelador durante a atividade de criação de diagramas baseados em histórico de processos, que podem ser obtidos a partir de repositórios. Wang, Gui e Cao [Wang et al. 2019] propõem a decomposição dos diagramas em fragmentos e uma metodologia de cálculo que permite apurar o percentual de similaridade do diagrama que está sendo criado com outros diagramas armazenados em um repositório. Jannach e Fischer [Jannach and Fischer 2014] e Jannach, Jugovac e Lerche [Jannach et al. 2015] propõem a utilização de aprendizado de máquina para alimentação de um repositório para se obter recomendações por critérios de ocorrência.

O presente trabalho se diferencia dos algoritmos de recomendação descritos anteriormente pelo fato de não utilizar uma base com processos previamente cadastrados (repositório) para realizar as recomendações, e o mecanismo proposto realiza uma análise conjunta de todas as restrições a serem atendidas pelo processo para a geração de recomendações. A Figura 1 apresenta a proposta deste trabalho no contexto dos sistemas de recomendação. Os dados a serem recomendados têm sua origem em repositórios de processos e no domínio do processo que está sendo modelado. Desses dados, são extraídos padrões de acordo com os critérios do sistema em questão e a recomendação é realizada. A proposta deste trabalho acrescenta a parte tracejada da figura, na qual a tabela de padrões origina um conjunto de restrições que precisam ser respeitadas, chamado de modelo declarativo. À medida que o modelador navega pelo sistema, são realizadas

então recomendações posicionais de fragmentos que podem ser inseridos naquele ponto do modelo em acordo com as regras. Após a inserção de um fragmento do modelo, o modelo declarativo é reconstruído com base no novo cenário e novas recomendações são propostas.

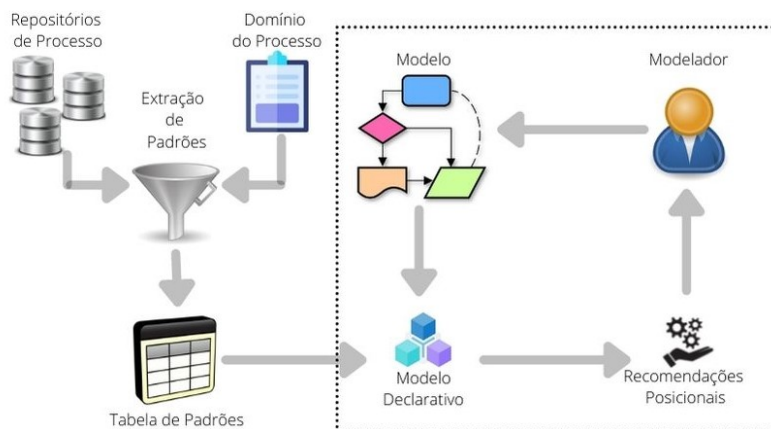


Figura 1. Arquitetura de recomendação com Composição de Padrões.

A geração do modelo baseado em padrões de recomendação é declarativa, sendo usado em uma notação BPMN imperativa. No paradigma imperativo, os modelos resultantes tendem a representar o comportamento esperado dos processos de uma forma determinística na qual todas as possibilidades de execução para o processo são explicitadas [Goedertier et al. 2015]. O paradigma declarativo tende a representar processos por meio de restrições ou regras sobre o seu comportamento, sendo que os fluxos de trabalho não são explicitados, devendo apenas estar em conformidade com as restrições e regras definidas no modelo [Fahland et al. 2009].

3. Padrões de Recomendação como Construções BPMN

BPMN é uma notação imperativa utilizada para modelagem de processos, padrão OMG (*Object Management Group*). BPMN é constituída de quatro categorias básicas de elementos de modelo: objetos de fluxo (*gateways*, atividades e eventos), objetos de conexão (*sequence flows*, *message flows* e *associations*), *swimlanes* e artefatos. As atividades constituem os principais elementos do BPMN. Eles denotam tarefas que devem ser realizadas e são representados por retângulos com cantos arredondados. O fluxo de sequência entre as atividades, o fluxo de controle, é representado por arcos. As direções dos arcos representam a ordem em que as atividades devem ser realizadas. Eventos, representados por círculos, denotam algo que acontece durante o tempo de vida do processo. O ícone dentro do círculo indica o tipo de evento, por exemplo, envelope para evento de mensagem, relógio para evento de tempo. *Gateways*, representados por formas de diamante, determinam bifurcação e fusão do fluxo de sequência entre as tarefas em um processo, dependendo de algumas condições.

Os diagramas BPMN, por serem aderentes ao paradigma imperativo, são prescritivos ou mandatórios. No entanto, recomendações podem ser tanto declarativas quanto imperativas. Quando uma recomendação é declarativa, essa pode ser mapeada em mais de um fragmento de modelo em BPMN. Por exemplo, considere um processo com atividades a, b e c, e a recomendação do tipo restrição <<executar b após a>>. Neste

caso existem três fragmentos BPMN que atendem a esta recomendação: $a \rightarrow b \rightarrow c$, $c \rightarrow a \rightarrow b$ e $a \rightarrow c \rightarrow b$. No caso de recomendações prescritivas, isto é, que definem um fragmento de modelo específico para ser adicionado ao diagrama, o mapeamento para BPMN é direto (injetor), e requer apenas a sua transcrição em termos de elementos de modelo da notação.

Os Padrões de Recomendação, ou *Recommendation Patterns*, apresentados por Costa e Tamzalit [Costa and Tamzalit 2017], propõem um estudo das relações entre as tarefas de um processo de negócio antes mesmo da criação de diagramas. Essas relações, uma vez estabelecidas, seriam utilizadas pelos sistemas de modelagem para auxiliar o modelador na criação de diagramas. Antes de realizar uma ação durante a construção ou manutenção de um diagrama, métodos baseados nos padrões de recomendação propõem que o modelador tenha acesso ao conjunto de situações do estado do modelo. Várias definições de Padrões de Recomendação são mapeadas no modelo BPMN: Conjunto de Tarefas, Relação de Escolha, Relação de Dependência, Ponto de Recomendação e Estado do Modelo.

- Um Estado do Modelo (S , *modeling State*) abrange todas as informações do modelo em um dado estágio de elaboração de um diagrama. Transições de estados de modelagem ocorrem quando uma ação é tomada pelo modelador. O objetivo do processo de recomendação é apoiar a evolução do modelo de um estado inicial (S_0) para um estado final (S_f) em que todas as situações do modelo sejam atendidas.
- Um Ponto de Recomendação (RP , do inglês *Recommendation Point*) refere-se a um ponto do diagrama. Cada RP possui recomendações específicas, que retratam para o modelador as possibilidades para a continuidade do diagrama respeitando as restrições dos padrões de recomendação no contexto do mesmo.
- Um Conjunto de Tarefas (TS , do inglês *Task Set*) é um subconjunto de tarefas que participam de alguma relação. Por exemplo, $[a, b]$ representa um conjunto formado pelas tarefas a e b .
- Uma Relação de Escolha (CR , do inglês *Choice Relation*) estabelece uma escolha entre fluxos de execução do processo. Essa escolha será entre dois TS s. É representada pelo símbolo \otimes dividindo os TS s à esquerda e à direita. Por exemplo, $(a, b \otimes c)$ representa uma relação de escolha entre dois conjuntos de tarefas: (a, b) e (c) . Na notação BPMN, quando houver uma relação de escolha, o sistema recomendará a abertura de uma bifurcação no diagrama, sendo representada por um *gateway* exclusivo.
- Uma Relação de Dependência (DR , do inglês *Dependency Relation*) representa uma ordem temporal de execução entre um TS denominado dependente e um TS denominado dependido. Uma dependência pode ser do tipo estrita, representado pelo símbolo \triangleleft , ou circunstancial, representado pelo símbolo \trianglelefteq . Na relação de dependência estrita $(a, b \triangleleft c)$, as tarefas a e b somente podem ser inseridas no modelo após a inserção da tarefa c . Na relação de dependência circunstancial $(a, b \trianglelefteq c)$, é permitida a inserção das tarefas a e b no modelo, desde que depois de c ou em um diagrama em que c não faça parte.

As situações consideradas nos padrões de recomendação formam um conjunto preliminar constituído de 5 situações: independência, dependência estrita, dependência circunstancial, união e não coexistência.

Uma situação de Independência é uma relação sobre um conjunto de objetos e indica que este conjunto não possui qualquer restrição de comportamento entre si. Em uma situação de dependência, um conjunto de tarefas, denominado dependente, depende de outro, denominado dependido, para ser executado. Ela se subdivide em dois grupos: dependência estrita e dependência circunstancial. Na dependência estrita, o conjunto de tarefas dependente somente pode ser inserido no diagrama após todas as tarefas do grupo dependido terem sido inseridas. Na dependência circunstancial, em um caminho do fluxo onde o conjunto de tarefas dependido é realizado, o conjunto de tarefas dependentes somente pode ser realizado após ele, porém, o conjunto dependente também pode ser realizado em um caminho do fluxo onde o conjunto de tarefas dependido não é realizado.

A situação de não coexistência é uma relação em que dois conjuntos de tarefas não podem fazer parte do mesmo caminho, sendo obrigatória a execução de apenas um dos conjuntos por vez. Das relações entre grupos de tarefas surge o conjunto independência, que é o conjunto de tarefas que pode ser inserido em determinado ponto do diagrama. As tarefas independentes podem ser realizadas em paralelo, e daí surge a relação de união entre elas.

4. A Proposta

Durante a criação de diagramas para representar o modelo, com base nas restrições existentes no estado atual, o sistema de modelagem baseado nos padrões de recomendação apresentará ao modelador as opções de modelagem existentes. À medida que o modelador escolher uma das opções apresentadas, o sistema irá computar a escolha e apresentar opções que retratem o novo estado do sistema, e esse ciclo se repetirá enquanto houver tarefas. Quando não houver mais tarefas a serem inseridas no diagrama, o sistema recomendará o evento de encerramento. Nas subseções, são descritas as regras e os algoritmos de recomendação para geração das opções, mantendo a corretude do modelo.

4.1. Regras

Durante a criação de *TSs* e Relações (*CRs* ou *DRs*), o algoritmo de apoio à recomendação atenta-se para as seguintes regras para garantir a consistência das recomendações:

- Uma tarefa não pode fazer parte dos dois *TSs* de uma *CR* ou *DR*;
- Duas tarefas não podem estar contidas simultaneamente em *TSs* diferentes, de duas relações distintas. Por exemplo, se um modelo possuir uma relação de escolha $CR_1 = (a, b \otimes c)$, este mesmo modelo não poderá possuir a relação de dependência $DR_1 = (a \triangleleft c)$, pois a *CR* obriga que somente uma das tarefas sejam executadas, *a* ou *c*; no entanto, a *DR* obriga que *a* seja executada depois de *c*, violando a *CR*. O mesmo princípio se aplicaria se a *DR* fosse circunstancial.

União entre relações (*CR* ou *DR*): Durante a criação das tarefas e restrições para o modelo, o sistema fará união de relações conforme os seguintes critérios:

- Relações de escolha (*CR*): Quando duas relações de escolha (CR_1 e CR_2) possuírem *TSs* contendo as mesmas tarefas, pode-se unir as relações, gerando uma nova (CR_f) entre o *TS* que é comum às duas relações e a união das demais tarefas dos dois *TSs*. Isto é, a relação $CR_1 = (a \otimes c, d)$ pode ser unida à relação $CR_2 = (b \otimes c, d)$, gerando a relação $CR_s = (a, b \otimes c, d)$.

- Relações de dependência (DR): Quando duas relações de dependência, circunstancial ou estrita, (DR_1 e DR_2) forem do mesmo tipo e possuírem TS s dependentes com as mesmas tarefas, pode-se unir as relações, gerando uma relação DR_s . Isto é, a relação $DR_1 = (x, y \triangleleft a, b)$ pode ser unida à relação $DR_2 = (z, w \triangleleft a, b)$, gerando a relação $DR_s = (x, y, z, w \triangleleft a, b)$. Da mesma forma, $DR_1 = (x, y \trianglelefteq a, b)$ pode ser unida à relação $DR_2 = (z, w \trianglelefteq a, b)$, gerando a relação $DR_s = (x, y, z, w \trianglelefteq a, b)$.

Dependência entre relações (CR ou DR): Durante a criação das restrições do modelo, o algoritmo será capaz de identificar a dependência entre uma relação R_1 que está sendo inserida no modelo e as demais, formando uma hierarquia de relações para saber recomendar somente tarefas independentes. Isto é, dadas as relações $CR_1 = (x, y, z \otimes a, b, c)$, $CR_2 = (x, \otimes y, z)$ e $DR_3 = (a, b \trianglelefteq c)$, o algoritmo reconhece que CR_2 e DR_3 estão contidos dentro de TS s de CR_1 , ou seja, as relações CR_2 e DR_3 somente entrarão em pauta na recomendação após o modelador fazer a escolha CR_1 .

4.2. Algoritmo de Recomendação

O algoritmo de recomendação recebe como entrada um modelo declarativo de processo formado por um conjunto de tarefas e situações. O algoritmo permite a implementação de mecanismos interativos nos quais são apresentados ao modelador, a cada estágio da modelagem, uma lista de recomendações de fragmentos de processos que podem ser inseridos no diagrama.

Para realizar as recomendações, o algoritmo realiza os seguintes passos: obter relações independentes, obter tarefas independentes, obter escolhas independentes e realizar recomendações. Em um dado estado do modelo, são obtidas as relações independentes, que servirão de base para as recomendações, pois relações contidas dentro de outras relações não podem ser recomendadas. O Algoritmo 1 ilustra esta etapa.

Algoritmo 1: Get Independent Relations

Data: List<Relation> relations

Result: List<Relation>

List<Relation> ret = new List<Relation>();

for each relation in relations do

 List<Relation> others = relations.FindAll(x => x.Id != relation.Id);

 if (!isContained(relation, others)) ret.Add(relation);

Neste algoritmo, em um dado estado do modelo, são obtidas as tarefas que podem ser inseridas sozinhas no modelo por serem independentes, ou por não estarem em nenhuma relação, ou por fazerem parte de uma relação independente na qual fazem parte do TS dependido, ou ainda por fazerem parte do TS dependente de uma relação independente de dependência circunstancial. Esta etapa é desempenhada pelo Algoritmo 2. As escolhas independentes são as CR s que fazem parte das relações independentes.

Após obter as tarefas e escolhas independentes de um dado estado do modelo, o algoritmo de recomendação (Algoritmo 3) recomenda uma escolha de tarefa para cada tarefa e um *gateway* exclusivo para cada escolha independente. Caso o ponto de recomendação esteja dentro de um *gateway* e não haja mais recomendações para o RP

atual e para o outro *RP* dentro do *gateway*, o sistema recomenda o fechamento do *gateway* em ambos os pontos. Caso o *RP* corrente seja o único ativo e não haja mais recomendações, o sistema recomendará o evento de encerramento do diagrama.

Algoritmo 2: Get Independent Tasks

Data: List<Task> tasks, List<Relation> relations
Result: List<Task>
List<Task> ret = new List<Task>();
for each task in tasks do
 bool isIndependent = true;
 for each relation in relations do
 RelationRole role = getRole(task, relation);
 if role == Role.Dependent AND relation.Type ==
 RelationType.StrictDependency then
 | isIndependent = false;
 | break;
 else if role != Role.NotIncluded AND !isIndependent(relation) then
 | isIndependent = false;
 | break;
 if isIndependent then
 | ret.Add(task);

Algoritmo 3: Make Recommendations

Data: RecommendationPoint rp
for each task in IndependentTasks do
 | rp.Recommendations.Add(new SelectTaskAction(task));
for each relation in IndependentRelations do
 | rp.Recommendations.Add(new OpenExclusiveGatewayAction(task));
if isInsideGateway(rp) AND rp.Recommendations.Count == 0 then
 | OpenExclusiveGateway gateway = getOpeningGateway(rp);
 | RecommendationPoint relatedRP = getRelatedRP(gateway, rp);
 if relatedRP.Recommendations.count == 0 then
 | rp.Recommendations.Add(new
 CloseExclusiveGatewayAction(gateway));
 | related.Recommendations.Add(new
 CloseExclusiveGatewayAction(gateway));
if rp.Recommendations.Count == 0 AND
 state.ActiveRecommendationPoints.All(x => x.Recommendations.Count ==
 0) **then**
 | rp.Recommendations.Add(new EndEvent());

4.3. Suporte à escolha de recomendações

Quando o modelador escolher um item recomendado, o algoritmo (Algoritmo 4) fará o tratamento adequado, gerando um novo estado para o diagrama de acordo com a recomendação escolhida. Quando uma tarefa é escolhida, ela é removida dos *TSs* dos *DRs* dos quais faz parte. Caso seja a tarefa dependente de uma *DR* circunstancial, a

tarefa dependida também deixa de fazer parte das escolhas possíveis para o modelador. Quando um *gateway* é aberto, a *CR* a ele associada é removida da lista de relações e são criados dois novos *RPs*, um para cada fluxo do *gateway*, contendo os respectivos *TSs* e as relações nele contidas. Quando um *gateway* é fechado, o algoritmo faz as recomendações padrão.

Algoritmo 4: Handle Modeler Action

Data: RecommendationPoint rp, IModelerAction action

switch action **do**

case *SelectTaskAction* *selectTask* **do**

 RecommendationPoint newRP = DeepCopy.Copy(rp);
 newRP.Tasks.RemoveAll(x => x.Id == selectTask.Task.Id);
 newRP.Relations.Foreach(x =>
 x.RemoveTask(selectTask.Task.Id)); newRP.Relations =
 newRP.Relations.RemoveAll(x => x.IsEmpty());
 MakeRecommendations(newRP); return newRP;

case *OpenGatewayAction* *openGateway* **do**

 RecommendationPoint leftRP = DeepCopy.Copy(rp);
 RecommendationPoint rightRP = DeepCopy.Copy(rp);
 leftRP.Tasks = openGateway.Relation.TaskSetA; right.Tasks =
 openGateway.Relation.TaskSetB;
 leftRP.Relations = leftRP.Relations.FindAll(x =>
 x.IsContained(leftRP.Tasks)); right.Relations =
 right.Relations.FindAll(x => x.IsContained(right.Tasks));
 MakeRecommendations(leftRP); MakeRecommendations(rightRP);
 return leftRP, rightRP

case *CloseGatewayAction* *closeGateway* **do**

 RecommendationPoint newRP = DeepCopy.Copy(rp);
 RecommendationPoint openingRP =
 GetOpeningRP(closeGateway.Relation); newRP.tasks =
 openingRP.Tasks.RemoveAll(x =>
 closeGateway.Relation.ContainsTasks(x)); newRP.Relations =
 openingRP.Relations; newRP.Relations.Foreach(x =>
 x.RemoveTask(selectTask.Task.Id)); newRP.Relations =
 newRP.Relations.RemoveAll(x => x.IsEmpty());
 MakeRecommendations(newRP); return newRP;

5. Estudo de Caso

Como estudo de caso, a solução proposta foi aplicada no cenário de Marketing de Conteúdo, elaborado pelos próprios autores, conforme descrição apresentada a seguir:

“O *marketing* de conteúdo pode ser uma bagunça muito agitada, a menos que você o organize em processos de negócios claros. A menos que você trabalhe nisso, você acabará tendo 20 artigos de trabalho em andamento perdidos no limbo, um designer muito confuso e uma base de seguidores confusa - esperando desesperadamente por seu artigo semanal. Dependendo da sua estratégia, existem muitas maneiras diferentes de estabelecer fluxos de trabalho de marketing de conteúdo. Para dar um exemplo, vamos passar por

um processo de publicação muito básico: o redator elabora o rascunho de um artigo, inclui descrições de quaisquer imagens personalizadas que devem ser usadas. Caso o artigo seja publicável, o editor o revisa, faz pontos sobre gramática, estilo, ortografia, etc. Caso contrário, ele descarta o artigo. Com o artigo aprovado pelo editor, o designer cria imagens personalizadas conforme solicitado, enviando-as ao redator de conteúdo. O redator inclui as imagens criadas pelo designer no artigo e as disponibiliza para publicação. O profissional de *marketing* reúne informações de contato de *influencers*, para serem usadas para publicidade e marketing assim que o artigo for publicado. O especialista em SEO (*Search Engine Optimization*) garante que o artigo atenda às melhores práticas de otimização e publica o artigo.”

Para utilizar a ferramenta é necessário, inicialmente, identificar as tarefas de um processo e a relação entre estas tarefas. Do modelo descritivo, extraem-se sete tarefas: *t1* - elaborar rascunho; *t2* - revisar artigo; *t3* - criar imagens; *t4* - incluir imagens; *t5* - reunir informações de publicidade; *t6* - publicar artigo; e *t7* - reprovar artigo.

Adicionalmente, é possível também extrair relações entre os conjuntos de tarefas:

- *t2* depende de *t1*, ou seja, para que o editor revise um artigo, esse precisa ter sido elaborado pelo redator;
- *t3* depende de *t2*, ou seja, somente após a revisão e aprovação pelo editor é que o designer irá criar imagens;
- *t4* depende de *t3*, ou seja, após receber as imagens do designer, o redator irá incluí-las no artigo;
- *t5* e *t6* dependem de *t4*, ou seja, quando o artigo for disponibilizado, o marketing e o SEO realizarão a publicação e a publicidade;
- Após a revisão pelo editor (*t2*), o artigo pode ser descartado (*t7*) ou encaminhado para o designer criar as imagens (*t3*).

Com as tarefas e as restrições, o usuário gera o modelo, com as tarefas e restrições apresentadas já inseridas na ferramenta conforme a Figura 2, na qual no quadro à esquerda é apresentada a lista de tarefas que compõem o modelo e à direita é apresentada a lista de restrições entre as tarefas.

Registered tasks	Relations																												
<table border="1"> <thead> <tr> <th>Task Name</th> <th></th> </tr> </thead> <tbody> <tr> <td>t7 - reprovar artigo</td> <td>Delete</td> </tr> <tr> <td>t2 - revisar artigo</td> <td>Delete</td> </tr> <tr> <td>t1 - elaborar rascunho</td> <td>Delete</td> </tr> <tr> <td>t3 - criar imagens</td> <td>Delete</td> </tr> <tr> <td>t4 - incluir imagens</td> <td>Delete</td> </tr> <tr> <td>t6 - publicar artigo</td> <td>Delete</td> </tr> <tr> <td>t5 - reunir informações de publicidade</td> <td>Delete</td> </tr> </tbody> </table>	Task Name		t7 - reprovar artigo	Delete	t2 - revisar artigo	Delete	t1 - elaborar rascunho	Delete	t3 - criar imagens	Delete	t4 - incluir imagens	Delete	t6 - publicar artigo	Delete	t5 - reunir informações de publicidade	Delete	<table border="1"> <thead> <tr> <th>Relation Text</th> <th></th> </tr> </thead> <tbody> <tr> <td>(t7 - reprovar artigo @ t3 - criar imagens)</td> <td>Delete</td> </tr> <tr> <td>(t2 - revisar artigo <= t1 - elaborar rascunho)</td> <td>Delete</td> </tr> <tr> <td>(t4 - incluir imagens <= t3 - criar imagens)</td> <td>Delete</td> </tr> <tr> <td>(t6 - publicar artigo ,t5 - reunir informações de publicidade <= t4 - incluir imagens)</td> <td>Delete</td> </tr> <tr> <td>(t3 - criar imagens ,t7 - reprovar artigo <= t2 - revisar artigo)</td> <td>Delete</td> </tr> </tbody> </table>	Relation Text		(t7 - reprovar artigo @ t3 - criar imagens)	Delete	(t2 - revisar artigo <= t1 - elaborar rascunho)	Delete	(t4 - incluir imagens <= t3 - criar imagens)	Delete	(t6 - publicar artigo ,t5 - reunir informações de publicidade <= t4 - incluir imagens)	Delete	(t3 - criar imagens ,t7 - reprovar artigo <= t2 - revisar artigo)	Delete
Task Name																													
t7 - reprovar artigo	Delete																												
t2 - revisar artigo	Delete																												
t1 - elaborar rascunho	Delete																												
t3 - criar imagens	Delete																												
t4 - incluir imagens	Delete																												
t6 - publicar artigo	Delete																												
t5 - reunir informações de publicidade	Delete																												
Relation Text																													
(t7 - reprovar artigo @ t3 - criar imagens)	Delete																												
(t2 - revisar artigo <= t1 - elaborar rascunho)	Delete																												
(t4 - incluir imagens <= t3 - criar imagens)	Delete																												
(t6 - publicar artigo ,t5 - reunir informações de publicidade <= t4 - incluir imagens)	Delete																												
(t3 - criar imagens ,t7 - reprovar artigo <= t2 - revisar artigo)	Delete																												

Figura 2. Regras do modelo de processo

A partir desse modelo, serão derivadas as recomendações. O estado definido pelas tarefas e restrições listadas configura o estado inicial S_1 do modelo para a criação de diagramas. Como recomendação, haverá o RP_1 contendo o conjunto independência desse modelo, que é constituído pela tarefa t_1 , conforme demonstrado pela Figura 3.

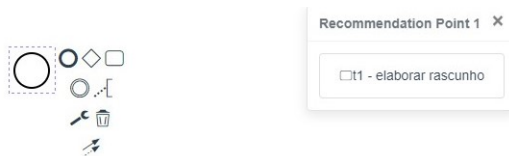


Figura 3. RP1

Quando a recomendação de RP_1 for escolhida, será criado então o estado S_2 , contendo o conjunto de tarefas $[t_2, t_3, t_4, t_5, t_6, t_7]$, com as restrições: (i) $(t_3, t_7) \triangleleft t_2$; (ii) $t_4 \triangleleft t_3$; (iii) $(t_5, t_6) \triangleleft t_4$; (iv) $t_7 \otimes t_3$. Na tarefa t_1 que foi inserida no diagrama, será criado o ponto de recomendação RP_2 , contendo uma única recomendação de escolha, a tarefa t_2 , conforme apresentado pela Figura 4.



Figura 4. RP2

Quando a tarefa t_2 for escolhida, será criado o estado S_3 , contendo o conjunto de tarefas $[t_3, t_4, t_5, t_6, t_7]$, com as restrições: (i) $t_4 \triangleleft t_3$; (ii) $(t_5, t_6) \triangleleft t_4$; (iii) $t_7 \otimes t_3$. Na tarefa t_2 inserida no diagrama, será criado o ponto de recomendação RP_3 . Nesse ponto, as tarefas t_3 e t_7 são independentes e possuem uma relação de não coexistência entre si, ocasionando a recomendação de abertura de um *gateway*, conforme apresentado pela Figura 5.



Figura 5. RP3

Quando a recomendação do *gateway* for escolhida, será criado o estado S_4 , contendo o conjunto de tarefas $[t_3, t_4, t_5, t_6, t_7]$, com as restrições: (i) $t_4 \triangleleft t_3$; (ii) $(t_5, t_6) \triangleleft t_4$. Em cada um dos dois pontos de bifurcação do *gateway*, será criado um ponto de recomendação, conforme apresentado pela Figura 6.

- RP_4 : tarefas $[t_3, t_4, t_5, t_6]$ e as restrições: $t_4 \triangleleft t_3$ e $(t_5, t_6) \triangleleft t_4$;
- RP_5 : tarefa $[t_7]$ e nenhuma restrição.

Após o modelador escolher a tarefa t_7 em um lado do *gateway* e ter escolhido todas as tarefas do outro lado, será então recomendado o fechamento deste *gateway*, conforme apresentado pela Figura 7. Após a escolha pelo fechamento, como não há mais

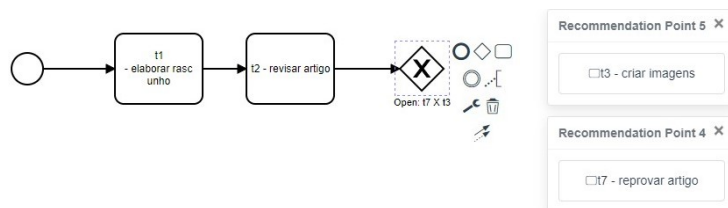


Figura 6. RPs 4 e 5

tarefas disponíveis para serem incluídas no diagrama, será então recomendado o evento de encerramento, conforme apresentado pela Figura 8.

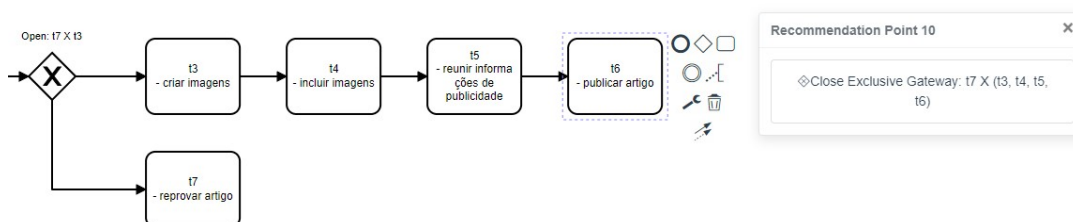


Figura 7. RP 10

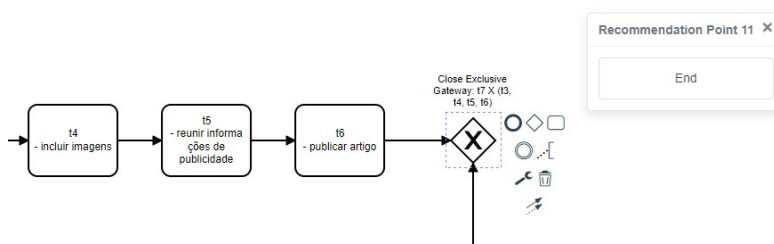


Figura 8. RP 11

Os experimentos tiveram o propósito de verificar a corretude do modelo e avaliar a utilização da ferramenta. Ao utilizar as recomendações, pode-se observar que o modelador é conduzido a diagramas corretos, com esforço de modelagem reduzido. O trabalho maior passa a ser a configuração das restrições entre as atividades, seja por meio de interpretação de descrições textuais, entrevistas ou utilizando outras técnicas.

5.1. Aspectos de implementação

O algoritmo foi desenvolvido e implementado em uma ferramenta tendo como base o projeto **bpmn.io**¹, que é uma plataforma *open source* de modelagem e possui uma ferramenta de criação de diagramas seguindo a notação BPMN, chamado **bpmn-js**². Foi adicionada então uma estrutura para suporte a modelos de usuários, e esses modelos contém a configuração de tarefas e restrições que servirão como base para a criação de diagramas. Existem duas telas que podem ser acessadas, uma no menu **Config**, onde o usuário

¹<https://bpmn.io/>

²<https://bpmn.io/toolkit/bpmn-js/>

criará as tarefas e restrições. Nesta versão da ferramenta, não está implementada ainda a questão de dependência circunstancial do algoritmo, e toda dependência é considerada como sendo estrita. O outro menu é o menu **Diagram**, onde o usuário criará seus diagramas. Abaixo do nome do usuário, existe a opção **Change Model**, que permite ao usuário configurar diversos processos (tarefas e restrições) para posteriormente criar diagramas.

6. Conclusão

Neste trabalho foi discutido o apoio à criação de diagramas BPMN por meio de recomendações. A proposta apresentada baseou-se no conceito de Padrões de Recomendação, o qual elabora uma técnica para obtenção de fragmentos de modelo BPMN a partir de restrições comportamentais do processo denominadas situações. Com base nesses padrões, foi desenvolvido um conjunto de algoritmos para conduzir um processo interativo em que as recomendações de modelagem são sugeridas de acordo com o estado corrente do modelo. As recomendações também são organizadas segundo a topologia do modelo. Isto é, em um mesmo estado de modelagem, para um dado ponto de modelo existe um conjunto específico de recomendações. No estudo de caso, as opções de modelagem recomendadas são consistentes com a especificação declarativa previamente fornecida e conduzem a modelagem à elaboração de modelos sintática e semanticamente corretos.

Como trabalhos futuros, serão realizados mais experimentos e testes para uma completa validação. Pretende-se melhorar a ferramenta com a funcionalidade de ajuste automático do *layout* de modelos de processo. Além disso, planeja-se a oferta de minicursos de BPMN usando a ferramenta, tanto para o público-alvo de modeladores iniciantes quanto o de experientes, e desta forma coletar a impressão de usuários finais, a produtividade, bem como o estudo da curva de aprendizado com relação à geração do modelo e à notação das restrições.

Referências

- Andaloussi, A. A., Burattin, A., Slaats, T., Kindler, E., and Weber, B. (2020). On the declarative paradigm in hybrid business process representations: A conceptual framework and a systematic literature study. *Information Systems*, 91:101505.
- Barba, I., Del Valle, C., Weber, B., and Jimenez, A. (2013). Automatic generation of optimized business process models from constraint-based specifications. *International Journal of Cooperative Information Systems*, 22(02):1350009.
- Costa, M. B. and Tamzalit, D. (2017). Recommendation patterns for business process imperative modeling. In *Proceedings of the Symposium on Applied Computing*, pages 735–742. ACM.
- Dijkman, R., Dumas, M., Van Dongen, B., Käärik, R., and Mendling, J. (2011). Similarity of Business Process Models: Metrics and Evaluation. *Information Systems*, 36(2):498–516.
- Fahland, D., Lübke, D., Mendling, J., Reijers, H., Weber, B., Weidlich, M., and Zugal, S. (2009). Declarative versus imperative process modeling languages: The issue of understandability. In *Enterprise, Business-Process and Information Systems Modeling*, pages 353–366. Springer.

- Fellmann, M., Metzger, D., Jannaber, S., Zarvic, N., and Thomas, O. (2018). Process modeling recommender systems. *Business & Information Systems Engineering*, 60(1):21–38.
- Goedertier, S., Vanthienen, J., and Caron, F. (2015). Declarative business process modeling: principles and modelling languages. *Enterprise Information Systems*, 9(2):161–185.
- Jannach, D. and Fischer, S. (2014). Recommendation-based modeling support for data mining processes. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 337–340.
- Jannach, D., Jugovac, M., and Lerche, L. (2015). Adaptive recommendation-based modeling support for data analysis workflows. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pages 252–262.
- Koschmider, A., Hornung, T., and Oberweis, A. (2011). Recommendation-based Editor for Business Process Modeling. *Data & Knowledge Engineering*, 70(6):483–503.
- Koschmider, A. and Oberweis, A. (2010). Designing business processes with a recommendation-based editor. In *Handbook on Business Process Management 1*, pages 299–312. Springer.
- Leyh, C., Bley, K., and Seek, S. (2016). Elicitation of processes in business process management in the era of digitization—the same techniques as decades ago? In *International Conference on Enterprise Resource Planning Systems*, pages 42–56. Springer.
- Li, Y., Cao, B., Xu, L., Yin, J., Deng, S., Yin, Y., and Wu, Z. (2014). An Efficient Recommendation Method for Improving Business Process Modeling. *Industrial Informatics, IEEE Transactions on*, 10(1):502–513.
- Pereira, J. A., Matuszyk, P., Krieter, S., Spiliopoulou, M., and Saake, G. (2018). Personalized recommender systems for product-line configuration processes. *Computer Languages, Systems & Structures*, 54:451–471.
- Schonenberg, H., Weber, B., van Dongen, B., and van der Aalst, W. (2008). Supporting Flexible Processes through Recommendations Based on History. In *Business Process Management*, pages 51–66. Springer.
- van der Aalst, W. M. (2010). Business process simulation revisited. In *Workshop on Enterprise and Organizational Modeling and Simulation*, pages 1–14. Springer.
- Wang, H. J. and Wu, H. (2011). Supporting Process Design for e-Business via an Integrated Process Repository. *Information Technology and Management*, 12(2):97–109.
- Wang, J., Gui, S., and Cao, B. (2019). A process recommendation method using bag-of-fragments. *International Journal of Intelligent Internet of Things Computing*, 1(1):32–42.
- Zhang, J., Liu, Q., and Xu, K. (2009). Flowrecommender: a workflow recommendation technique for process provenance. In *Proceedings of the Eighth Australasian Data Mining Conference-Volume 101*, pages 55–61. Australian Computer Society, Inc.