

Interações Automatizadas para Apoio à Modelagem de Processos de Negócio

Fabio Bermudes Cabral, Bruno Cardoso Coutinho,
Fidelis Zanetti de Castro, Mateus Barcellos Costa

*Instituto Federal do Espírito Santo (IFES), Campus Serra
(e-mail: fabiocabral@hotmail.com, {bccout,fidelis,mcosta}@ifes.edu.br)*

Abstract:

Business Process Modeling is an activity increasingly adopted by organizations seeking to improve their operational performance. A common practice for this activity is to perform interviews and workshops by involving modeling professionals (modelers) and domain experts. The produced models, in this case, are built based on the interpretation of the views gathered from domain experts. Considering this context, this paper discusses an approach for automated interactions between a computational agent and domain experts with the aim of determining process' control flow constraints, preventing potential inconsistencies and identifying modeling alternatives. The interaction generation mechanism, which is based on the building of the process's discernment structure, provides to the computational agent the complete view of the modeling solution space. As a proof of concept, a prototype to support interactions was developed. The performed tests and experiments demonstrate that interactions produce valid and consistent models and allow the analysis of modeling alternatives.

Resumo: A Modelagem de Processos de Negócio é uma atividade cada vez mais adotada por organizações que buscam melhorar seu desempenho operacional. Uma prática comum para esta atividade consiste na identificação do comportamento dos processos e sua transcrição em modelos, ao longo de entrevistas e oficinas envolvendo profissionais de modelagem (modeladores) e especialistas de domínio. Os modelos produzidos, neste caso, são construídos por modeladores tendo como base sua interpretação das visões colhidas junto a especialistas de domínio. Considerando este contexto, este trabalho discute um modelo de interações automatizadas com especialistas de domínio para capturar restrições de controle do fluxo do processo. O modelo concebido para apoiar as interações baseou-se na construção de uma *estrutura de discernimento*, que fornece a visão de todo o Espaço de Solução para o processo. Como prova de conceito foi desenvolvido um protótipo para interações entre um agente computacional e um humano. Conforme verificado em testes e experimentos com o protótipo, as interações permitem a análise de alternativas de modelos, produzem modelos consistentes e avaliam todo o Espaço de Solução de modelagem do processo.

Keywords: Business Process Modeling; Process control flow; Constraint Graphs; Automated interactions; BPMN.

Palavras-chaves: Modelagem de processos de negócios; Controle de fluxo de processos; Grafos de restrição; Interações automatizadas; BPMN.

1. INTRODUÇÃO

Na Modelagem de Processos de Negócio, o conhecimento acerca dos processos tratados, muitas vezes manifestado apenas de modo tácito ou subjetivo (Verner, 2004), deve ser conhecido para depois ser transcrito em modelos utilizando-se alguma notação ou linguagem de modelagem, e.g., BPMN - *Business Process Modeling and Notation* ou EPC - *Event Process Chain*. É prática comum na

modelagem a participação de um especialista modelador, que constrói os modelos a partir de sua interpretação das visões colhidas junto a especialistas de domínio. Nestas interações determina-se o controle de fluxo do processo, o qual especifica em termos lógico-temporais a ordem de execução das atividades e eventos relacionados ao mesmo. Tal tarefa apresenta grandes desafios por estar sujeita a erros, tanto de concepções quanto de interpretações, os quais desviam os modelos produzidos da realidade que deveriam representar (Leyh et al., 2016; Kannengiesser and Oppl, 2015).

Neste trabalho, argumentamos que, para mitigar o eventual *bias* inerente às interações de modelagem, o modelador deve ter uma visão precisa do comportamento esperado para o processo e do Espaço de Solução da modelagem e

* Agradecemos à FAPES (Fundação de Amparo à Pesquisa e Inovação do Espírito Santo) e à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro dado por meio do PDPG (Parcerias Estratégicas nos Estados da CAPES). Também agradecemos ao Propós (Programa Institucional de Apoio à Pós-Graduação *Stricto Sensu*) do IFES pelo apoio financeiro.

suas alternativas. Considerando que o comportamento do processo é de conhecimento dos especialistas de domínio, apresentamos, como contribuição principal, uma abordagem para a determinação do controle de fluxo de processos, em termos de suas restrições lógico-temporais, baseada em interações automatizadas. No modelo proposto, uma sequência de perguntas, realizadas por um agente computacional e respondidas por um especialista de domínio, resulta em um modelo consistente e em conformidade com a realidade apresentada por este último.

O modelo de interações baseia-se no conceito *estrutura de discernimento* (Schubert, 2012; Zhou et al., 2017) para representar todas as hipóteses aceitáveis de conjuntos de restrições de controle de fluxo que são mapeáveis em modelos BPMN (Group, 2013). Embora a solução proposta tenha complexidade exponencial, estratégias de condução das interações permitem, mesmo assim, a obtenção bem-sucedida de modelos de restrição, conforme revelaram os experimentos realizados.

O restante deste artigo está organizado da seguinte forma: A Seção 2 discute aspectos fundamentais de modelagem e interações automatizadas. A Seção 3 discute a arquitetura e o modelo propostos. Na Seção 4 o modelo proposto é discutido por meio de um exemplo presente na literatura. A Seção 5 apresenta uma discussão dos resultados. A Seção 6 apresenta uma breve revisão de literatura de alguns trabalhos correlatos ao nosso. A Seção 7 conclui o artigo, discute suas limitações e trabalhos futuros.

2. CONCEITOS

2.1 Modelos de Restrição para Processos de Negócio

A modelagem de processos é um método para descrever como as organizações conduzem suas operações. Normalmente, utiliza-se representações capazes de descrever aspectos organizacionais e elementos inerentes ao processo como suas atividades e eventos, seus estados e sua lógica de controle de fluxo (Recker et al., 2009).

A lógica de controle de fluxo de processos pode ser representada em termos de restrições, por meio de notações ou linguagens declarativas (Fahland et al., 2009a). Abordagens declarativas visam expressar os limites comportamentais do processo de negócio, estabelecendo a construção de modelos por meio de declarações que especificam restrições lógico-temporais sobre o comportamento do processo (Pesic et al., 2007). Exemplos de abordagens declarativas são a linguagem Declare (Pesic et al., 2007) e a notação minimalista, baseada no conceito de *Situação*, proposta por Costa and Tamzalit (2017), que foi considerada neste trabalho.

Uma Situação, conforme definida pelos autores, é descrita por meio de tipos de relações entre os objetos de fluxo ativos que compõem o *Domínio do processo (D)*. A Tabela 1 apresenta o conjunto de tipos de Situação propostos pelos autores.

A partir de um modelo baseado em situações é possível a obtenção de modelos imperativos (Fahland et al., 2009b). Enquanto abordagens declarativas delegam ao tempo de execução do processo a definição do fluxo de execução a ser tomado (desde que respeitadas as restrições), abordagens

Situação	Descrição	Simbologia
Independência	relação explícita de independência temporal de um conjunto de objetos de fluxo	$*(a, b)$ (a e b podem ocorrer em qualquer ordem)
Dependência Estrita	relação de precedência estrita entre dois grupos de objetos de fluxo	$b \triangleleft a$ (b depende de a)
Dependência Circunstancial	relação de precedência entre dois grupos de objetos de fluxo, na presença do grupo dependido	$b \triangleleft a$ (b depende de a, na presença de a)
Não Coexistência	relação de exclusão mútua (xor) entre dois grupos de objetos de fluxo	$a \otimes b$ (a exclui b, e b exclui a em um mesmo fluxo)
União	Relação de união (ou lógico) entre dois grupos de objetos de fluxo	$a \oplus b$ (a, a e b, ou b, podem ocorrer em um mesmo fluxo)

Tabela 1. Tipos de Situações para a criação de Modelos de Controle de Fluxo

imperativas determinam explicitamente no modelo quais são estas agendas de execução. *Business Process Model and Notation - BPMN* (Group, 2013), por exemplo, é uma notação aderente ao paradigma imperativo. A título de exemplo e para comparação entre as duas abordagens, a Figura 2 ilustra um diagrama BPMN para um processo denominado Aprovação de Projeto.

Na Figura 1 é apresentado um modelo declarativo baseado em situações, do qual o Modelo Aprovação de Projeto, da Figura 2, foi derivado.

```

Domínio *****
E0: Chegada de Projeto
T1: Registrar projeto - T2: Recuperar Projeto
T3: Avaliar - T4: Registrar Relatório
T5: Emitir Alvará - T6: Pedir Mudanças
Situações *****
T1 DEP E0 T2 DEP E0
T3 DEPC T1 T3 DEPC T2
T1 XOR T2 T4 DEP T3
T5 DEP T3 T6 DEP T3
T5 XOR T6
    
```

Figura 1. Modelo de restrições para o processo Aprovar Projeto

Neste modelo, DEP, DEPC e XOR são rótulos usados para representar, respectivamente, a dependência estrita, circunstancial e a não coexistência. É possível observar que o modelo BPMN da Figura 2 é uma *instância* de modelo BPMN válida com relação ao modelo de restrições. Isso significa que, para o modelo de restrições apresentado, foi possível construir um modelo sintaticamente correto com relação à notação BPMN, e que não viola nenhuma das restrições do modelo declarativo.

2.2 Interações Automatizadas

Entrevistas e oficinas estão entre as principais ferramentas utilizadas para o levantamento de informações para modelagem de processos. De acordo com Leyh et al. (2016), entrevistas são usadas para coletar fatos detalhados sobre

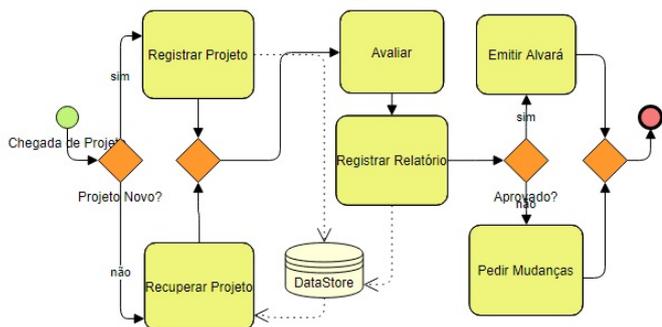


Figura 2. Modelo de processo em BPMN - Aprovação de Projeto.

os processos durante uma conversa. Dentre os desafios para a realização das mesmas inclui-se o alto grau de comprometimento que é exigido de todos os envolvidos. Problemas de agendamento e desgastes devido a longos períodos de duração são comuns. Outra questão acerca de entrevistas é que, para que não haja erros ou más interpretações, estas requerem um amplo conhecimento do domínio de possibilidades do processo.

Entrevistas para identificação do controle de fluxo do processo podem apresentar um grau maior de formalidade e estrutura, visto que as questões a serem levantadas são restritas ao estabelecimento de, basicamente, construções lógico-temporais. Entretanto, o levantamento de tais construções pode incorrer em um espaço de possibilidades muito grande, que pode levar o entrevistado a ter dificuldades em fornecer informações que se aplicam a toda a variabilidade do processo que está sendo elicitado.

Considerando estas dificuldades, e para evitar desvios (*bias*) por falta de compreensão plena do Espaço de Solução de modelagem e suas alternativas, consideramos um modelo para obtenção das restrições de controle de fluxo no qual o modelador pode delegar a um agente computacional a realização de interações com o especialista de domínio.

Para ilustrar essas interações, considere o exemplo do Processo Aprovar Projeto. Após identificar as atividades e eventos (E0, T1, ..., T6), o agente computacional realiza uma série de interações por meio de perguntas diretas, por exemplo, – *A atividade Registrar Projeto Depende do evento Chegada de Projeto?* – Neste caso, a resposta (sim ou não) introduz ou elimina a possibilidade de uma restrição do modelo. A interação pode também realizar perguntas indiretas, inferindo a ação a ser tomada a partir da resposta. Por exemplo, a pergunta – *Qual a primeira atividade do processo?* – sendo respondida pela indicação do Evento E0, permite inferir que E0 não possui dependência de execução com qualquer outro objeto de fluxo.

A automatização da entrevista baseou-se na determinação de perguntas a partir de buscas na *estrutura de discernimento* de modelos para o processo (*Frame of Discernment*) (Schubert, 2012). Esta estrutura corresponde ao conjunto de todas as hipóteses de modelos aceitáveis para o processo, o qual denominamos *Espaço de Solução* da modelagem. A estrutura de discernimento foi obtida por meio de árvores de subconjuntos do conjunto $\mathcal{P}(S)$ das

partes (*power set*) do conjunto S de todas as situações que podem ser formuladas para o domínio do processo tratado. Dessa forma, o problema assume complexidade de tempo e espaço com função teto $F(N_S) \propto 2^{N_S}$, em que N_S é o número de situações possíveis para um processo com domínio de tamanho n , para a linguagem adotada.

Para este trabalho, foi adotado um subconjunto das situações propostas por Costa and Tamzalit (2017), a saber: Dependência ($b \triangleleft a$), Não-coexistência ($a \otimes b$) e União ($a \oplus b$). Neste caso, sendo S o conjunto de todas as situações que podem ser formuladas para o domínio do processo e o domínio do processo possuindo cardinalidade igual a n , a cardinalidade de S é dada por

$$N_S = 2 \cdot \binom{n}{2} + \binom{n}{2} = 4 \cdot \binom{n}{2} = \frac{2n!}{(n-2)!} = 2n(n-1).$$

O crescimento de N_S segue portanto a progressão aritmética de segunda ordem correspondente à sequência $\{4, 12, 24, 40, 60, 84, 112, \dots\}$. Na hipótese de todos os modelos do Espaço de Solução serem consistentes, o tamanho do problema em tempo e espaço cresceria proporcionalmente a $2^{2n(n-1)}$, tornando-se intratável mesmo para um número pequeno de tarefas.

A Figura 3 ilustra o crescimento do Espaço de Solução com o aumento do número de atividades do processo. Em cada grafo estão um conjunto de modelos e cada modelo é representado por um conjunto de vértices interligados por arestas de mesma cor. Da esquerda para a direita, cada grafo apresenta este espaço para domínios de tamanho 3, 4 e 5, respectivamente. Percebe-se nesta figura, a explosão combinatorial exponencial do problema. Entretanto, conforme apresentado na Seção 4, as podas realizadas no espaço a partir das respostas obtidas nas interações reduzem este espaço também de forma exponencial. Observa-se ainda que os espaços de solução para todos os processos com o mesmo número de atividades, para o mesmo conjunto de situações, são equivalentes, a menos de um isomorfismo. Sendo assim, na prática, o Espaço de Solução precisa ser construído uma única vez para cada tamanho de domínio do processo.

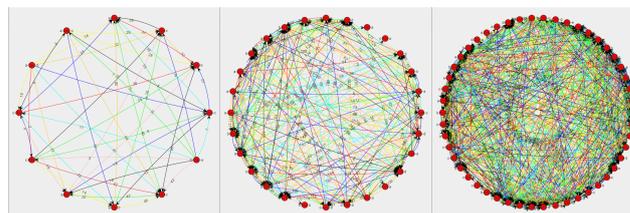


Figura 3. Grafos dos espaços de Solução para 3, 4 e 5 atividades.

3. MODELO PARA INTERAÇÕES AUTOMATIZADAS

A Figura 4 ilustra, por meio de um diagrama de classes UML, o modelo arquitetural para apoiar as interações automatizadas e obter modelos de controle de fluxo de processos. No modelo, o componente *Entrevistador* é

responsável por conduzir a entrevista e interagir com o especialista de domínio. Inicialmente, este elemento obtém informações do processo, que incluem os seus objetos de fluxo ativos. Posteriormente, o entrevistador passa a controlar as interações com o objetivo de obter as situações do processo que constituirão o seu modelo.

As perguntas são produzidas pelo componente *Gerador de Perguntas*, com o apoio de um raciocinador (componente *Reasoner*). O *Reasoner* é uma máquina de inferência baseada no estado da entrevista e no Espaço de Solução do Processo. O *Reasoner* infere as melhores perguntas a serem realizadas, tendo como parâmetro, a minimização do número de perguntas necessárias para a obtenção de um modelo satisfatório. Na solução implementada, a métrica utilizada para determinar estas perguntas é a capacidade do corte no Espaço de Solução, que ocorre por meio de podas baseadas nas respostas obtidas. Essa medida corresponde ao grau dos vértices no grafo do Espaço de Solução, sendo considerados os vértices correspondentes a situações válidas para o estado corrente do modelo. Por exemplo, considere o modelo da Tabela 1 e que seu estado corrente inclua apenas as declarações [T1 DEP E0], [T2 DEP E0] e [T3 DEPC T1]. Nesse caso, o *Reasoner* identificará no Espaço de Solução a situação válida para o modelo corrente que seja correspondente ao vértice de maior grau no grafo correspondente ao Espaço de Solução em seu estado corrente.

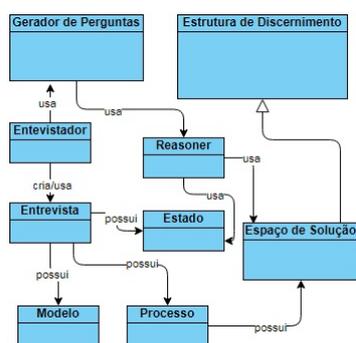


Figura 4. Arquitetura para um Sistema de Entrevistas Automatizadas

3.1 Dinâmica das Interações

Uma interação inicia-se identificando o conjunto preliminar de *Objetos de Fluxo Ativo* do processo. Para efeito de simplificação, eventos e atividades foram tratados indistintamente como atividades. Uma vez que as atividades são informadas, o entrevistador passa a realizar perguntas relacionando as atividades com base nas situações discutidas na Seção 2.1. Cabe ao entrevistado responder se a relação é verdadeira ou não.

A resposta a cada pergunta pode alterar o estado do modelo atual e reduzir a área de busca no Espaço de Solução, sendo preservada a consistência dos modelos. Uma vez interrompida a entrevista, os modelos resultantes são registrados, podendo a interação ser retomada ou considerada concluída. No caso de ser retomada, novas atividades podem ser adicionadas ao domínio do processo e o progresso da interação passará a considerar a nova configuração do domínio.

3.2 Validação de modelos

A validação de modelos pertencentes ao Espaço de Solução adota como critério parte da semântica da linguagem BPMN (Wong and Gibbons, 2008), de sorte que, para um processo P com domínio D , todo conjunto de situações formadas a partir dos elementos de D para as quais um modelo BPMN, contendo todos os elementos de D , possa ser construído, é um modelo válido.

Como exemplo, considere o modelo $P_1 = \{D = \{a, b, c\}, C = \{b \triangleleft a, a \otimes c\}\}$. Esta especificação é válida, pois é possível construir um modelo BPMN que inclui todos os elementos do domínio de P e não viola suas situações. Por outro lado, a especificação $P_2 = \{D = \{a, b, c\}, C = \{b \triangleleft a\}, a \otimes b\}$ não é válida, uma vez que não é possível construir um modelo BPMN que esteja em conformidade com as restrições do processo. Isso pode ser facilmente notado, já que não é possível b depender de a ao mesmo tempo que a seja não coexistente com b .

A semântica do subconjunto de interesse de BPMN, a saber, sequência, desvios exclusivos e desvios inclusivos, e suas combinações, foi traduzida em um conjunto de testes assertivos capazes de determinar a existência de um mapeamento válido entre modelos de situações e modelos BPMN. Cada asserção, quando verificada verdadeira, para um dado modelo de situações, indica a inexistência de tal mapeamento. Em termos práticos, o resultado é que não é possível criar um modelo BPMN válido a partir do modelo de situações considerado.

- (1) Operadores Equivalentes - A condição de Operadores equivalentes ocorre quando duas situações de tipos diferentes possuem os mesmos operadores. Por exemplo, $C = b \triangleleft a, a \oplus b$.
- (2) Dependência Cíclica - Uma dependência cíclica ocorre quando um conjunto de operadores entra em um ciclo de dependência causado pelas situações de um modelo. Por exemplo, um ciclo pode ser verificado na especificação $C = b \triangleleft a, a \triangleleft b$ ou $C = b \triangleleft a, a \triangleleft c, c \triangleleft b$.
- (3) Impedimento de Dependência Indireta - Impedimento de Dependência Indireta ocorre quando os operadores em uma relação de dependência indireta estão envolvidos em uma relação de escolha na mesma especificação. Neste caso, não existe um modelo BPMN válido onde os operadores possam depender uns dos outros ao mesmo tempo que atendem a uma relação de escolha. Na Especificação $C = a \triangleleft b, b \triangleleft c, a \oplus c$, ocorre um impedimento de dependência indireta uma vez que a depende indiretamente de c e ambos estão em uma relação de escolha.
- (4) Promiscuidade - A promiscuidade ocorre quando existe uma dependência entre dois operandos que estão, cada um, em escolhas diferentes com tipos diferentes. Por exemplo, $C = a \otimes c, a \triangleleft b, b \oplus c$ tem uma relação promíscua, já que dado que a depende de b , não é possível ter um modelo BPMN contendo uma não coexistência (xor) entre a e b e uma união (ou inclusivo) entre b e c .
- (5) Dependência Dual - A dependência dual ocorre quando ambos os operadores de uma relação de escolha são dependentes do mesmo operador na especificação. Por exemplo, na especificação $C =$

$$S_{1\dots 5} = \{a - dep - b, a - dep - c, c - dep - d, a - xor - b, c - uni - d\}$$

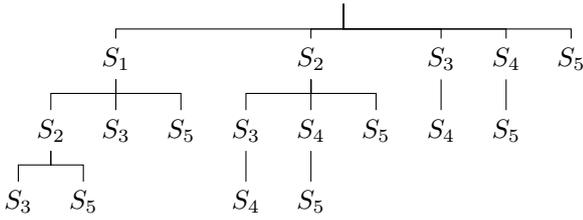


Figura 5. Árvore do espaço de solução para o conjunto de situações S

$a \otimes b$, $c \triangleleft a$, $c \triangleleft b$ há uma violação de dependência dual porque, uma vez que c depende de a e b que são não coexistentes, c não poderá ser executado.

3.3 Construção do Espaço de Solução

A Figura 5 mostra uma árvore N -ária representando um Espaço de Solução. A árvore denota o espaço correspondente ao conjunto de 5 situações do conjunto S . O caminhamento em profundidade na árvore, a partir de sua raiz até qualquer um de seus nós, sempre produz um caminho cujos vértices formam um modelo.

A construção de árvores de espaços de solução é baseada na geração do conjunto das partes (*power set*) do conjunto base $S = \{s_1..s_n\}$ de n situações. Para um dado Processo P , S é obtido considerando os tipos de situação da notação adotada. O algoritmo de geração opera construindo as combinações de elementos de S de ordem 1, 2, 3, ..., n . Elementos de ordem 1 possuem 1 elemento, de ordem 2, por dois elementos e assim por diante. O método de construção destas árvores é realizado pelos algoritmos 1 e 2. A função *Inserer Situação* é usada para validar a inserção de novos nós na árvore, tendo como base as asserções discutidas na Seção 3.2.

O Algoritmo 1 percorre o conjunto S e, para cada s_i , chama a função *Gera Árvore* que construirá árvores tentando inserir s_i como nó filho de todos os nós atuais da árvore.

Alg. 1 Gera Arvore De Caminhos

```

1: EnaryTree geraArvoreDeCaminhos(ListaRestrS)
   {
2: Caminho c;
3: EnaryTree a;
4: a.root := null;
5: for all  $s_i \in S$  do
6:   geraArvore(a.root,  $s_i$ , c);
7: }
```

4. EXPERIMENTOS

Foi desenvolvido um protótipo de um sistema de entrevista o qual foi testado com o intuito de (1) verificar a obtenção de modelos esperados em entrevistas e (2) verificar a capacidade da solução de processar entrevistas com o crescimento do domínio do processo. O contexto

Alg. 2 Gera Árvore

```

1: geraArvore(Node noCorrente, Situacao v, Caminho c)
2: Caminho nc ← c;
3: Caminho cCorrente;
4: if (insereSituacao(nc, v)) then
5:   Node[] children ← noCorrente.children;
6:   if (children ≠ null) then
7:     for all (Node n ∈ children) do
8:       cCorrente ← c;
9:       insereSituacao(cCorrente, n, Situacao);
10:      geraArvore(n, s, cCorrente);
11:   Node newNode;
12:   newNode.children ← null;
13:   newNode.Situacao ← s;
14:   noCorrente.children+ ← newNode;
```

de modelagem utilizado foi o apresentado por Tolfo et al. (2013), formado pelos processos de uma fábrica de artefatos de marcenaria. O processo *Realizar Marcenaria*, utilizado aqui para apresentar a utilização do protótipo, possui a seguinte descrição:

Processo Realizar Marcenaria. *Primeiramente, é analisada a existência de um produto. Caso ele não exista, ele deve ser projetado para que então o corte das peças que o compõe seja realizado. Havendo necessidade, de acordo com o produto, é feita a montagem do mesmo.* Para tanto uma série de atividades e um dado controle de fluxo necessita ser determinado.

Uma vez configurada uma nova entrevista, solicitando-se ao usuário que informe as atividades *a priori* do processo, as seguintes atividades foram definidas:

Atividade	ID
1- Abrir arquivo de produto preexistente	Ab2
2- Projetar o produto	Pr0
3- Cortar peças	Co3
4- Entregar produto montado	En2
5- Entregar produto desmontado.	En6

Iniciada a seção de perguntas, o sistema busca reduzir o tamanho do Espaço de Solução eliminando situações triviais, pela determinação de situações incompatíveis como modelos válidos. No exemplo, ao se aferir que a atividade Co3 depende da atividade Ab2, a situação [ab2 DEP Co3] pode ser eliminada do conjunto base. Nesta etapa da entrevista não é possível inferir as alternativas de modelos válidos, mas apenas determinar situações que nunca farão parte desses modelos. Eliminada uma situação, o conjunto base foi reduzido de 40 para 39 situações e o Espaço de Solução foi gerado contendo 428363 especificações válidas.

Em seguida o sistema passa a realizar podas no Espaço de Solução com base em perguntas respondidas com respostas afirmativas. A partir dessa etapa, perguntas com respostas negativas não foram consideradas para podas pois não implicam necessariamente em uma hipótese estrita, ao passo que perguntas respondidas afirmativamente determinam a presença da situação no modelo.

Quando a pergunta (*A atividade Entregar produto sem montagem depende da atividade Cortar peças para executar*

em algum fluxo?) é confirmada pelo usuário, 361553 especificações são eliminadas, indicando uma redução exponencial do Espaço de Solução. O processo segue e, após uma sequência de podas, a pergunta (*A atividade Abrir arquivo de produto preexistente é opcional em relação a atividade Cortar peças?*) ao ser respondida afirmativamente, elimina mais 412 especificações, concluindo a entrevista com 9 modelos válidos, mostrados a seguir:

- M1=[[En2-Dep-Co3], [En6-Dep-Co3], [En2-Dep-Pr0], [En6-Dep-Pr0], [Ab2-Xor-Pr0] [En2-Xor-En6], [Ab2-Uni-Co3]]
- M2=[[En2-Dep-Co3], [En6-Dep-Co3], [En2-Dep-Pr0], [Ab2-Xor-Pr0], [En2-Xor-En6], [En6-Xor-Pr0], [Ab2-Uni-Co3]]
- M3=[[En2-Dep-Co3], [En6-Dep-Co3], [En2-Dep-Pr0], [Ab2-Xor-Pr0], [En2-Xor-En6], [Ab2-Uni-Co3]]
- M4=[[En2-Dep-Co3], [En6-Dep-Co3], [En6-Dep-Pr0], [Ab2-Xor-Pr0], [En2-Xor-En6], [En2-Xor-Pr0], [Ab2-Uni-Co3]]
- M5=[[En2-Dep-Co3], [En6-Dep-Co3], [En6-Dep-Pr0], [Ab2-Xor-Pr0], [En2-Xor-En6], [Ab2-Uni-Co3]]
- M6=[[En2-Dep-Co3], [En6-Dep-Co3], [Ab2-Xor-Pr0], [En2-Xor-En6], [En2-Xor-Pr0], [En6-Xor-Pr0], [Ab2-Uni-Co3]]
- M7=[[En2-Dep-Co3], [En6-Dep-Co3], [Ab2-Xor-Pr0], [En2-Xor-En6], [En2-Xor-Pr0], [Ab2-Uni-Co3]]
- M8=[[En2-Dep-Co3], [En6-Dep-Co3], [Ab2-Xor-Pr0], [En2-Xor-En6], [En6-Xor-Pr0], [Ab2-Uni-Co3]]
- M9=[[En2-Dep-Co3], [En6-Dep-Co3], [Ab2-Xor-Pr0], [En2-Xor-En6], [Ab2-Uni-Co3]]

Todos os modelos foram verificados como válidos e em conformidade. Ou seja, os modelos são todos mapeáveis em diagramas BPMN e estão de acordo com a visão do especialista de domínio. Conforme pode ser percebido, o sistema foi também capaz de apresentar ao usuário as alternativas de modelos diante da visão colhida junto ao especialista. Assim, o Modelador pode optar com segurança por um modelo imperativo que atenda a uma dessas visões. Ao todo foram realizadas 30 perguntas para a obtenção da especificação esperada, a partir de um Espaço de Solução total de 428363 especificações possíveis. Na Figura 6, vemos o espaço da solução inicial gerado logo após a inclusão das atividades, e o Espaço de Solução final após as podas realizadas baseadas nas respostas do entrevistado.

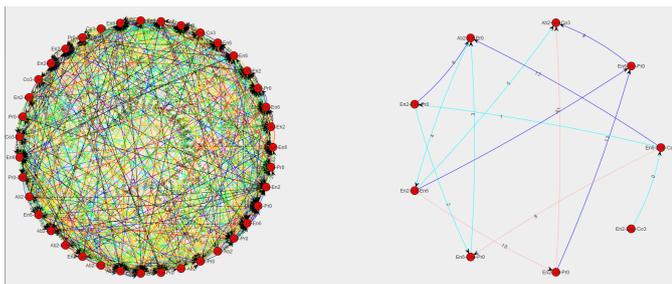


Figura 6. Espaço de Solução e modelos resultantes após a interação.

O estudo continua com acréscimo de novas atividades devido a mudanças no contexto do processo. É acrescentado o seguinte comportamento ao processo:

Versão 2 - Registrar entrada do pedido no setor de Marcenaria.

Assim, a atividade *Registrar entrada do pedido* é adicionada ao processo, ficando este com 6 atividades. As novas situações, construídas como combinações da nova atividade e das anteriormente existentes, foram incluídas, e a interação realizou as perguntas com relação às mes-

mas, considerando o Espaço de Solução já podado e os novos modelos gerados pela combinação dos modelos já existentes com as novas situações. Após a inclusão, foi dado início uma nova rodada de interações que produziu novamente modelos válidos e em conformidade. Um dos modelos produzidos foi M9=[[Ab2-Dep-Re2], [p20-Dep-Re2] [En2-Dep-Co3], [En6-Dep-Co3], [Ab2-Xor-Pr0], [En2-Xor-En6], [Ab2-Uni-Co3]], onde Re2 é o identificador da atividade Registrar entrada do pedido. A título de ilustração, a Figura 7 ilustra um diagrama BPMN válido para este modelo.

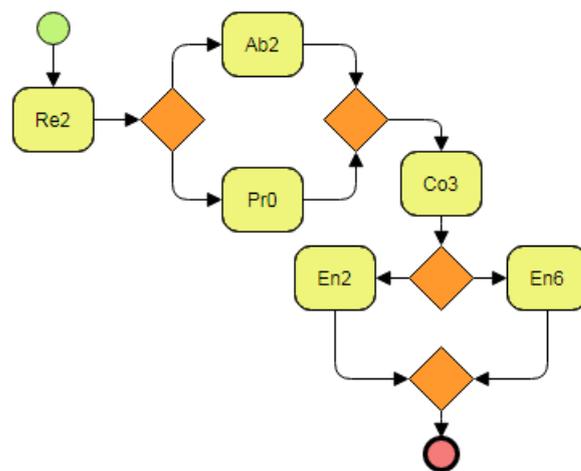


Figura 7. Modelo BPMN mapeado a partir de um modelo M9, resultante da interação.

5. RESULTADOS E DISCUSSÃO

Para a avaliação do modelo proposto, o processo Realizar Marcenaria foi novamente testado mas agora já iniciando com 6 atividades. Neste novo cenário, o Espaço de Solução passou a ter tamanho proporcional a 2^{60} . Testes na tentativa de geração deste espaço foram realizados utilizando uma máquina com *chipset* Intel Z170XP, processador Intel Core I7 7700 com quatro núcleos e 8 *threads* com 16 gigabytes de memória RAM. No caso de 5 atividades, o Espaço de Solução completo foi obtido facilmente em 35.755s, sendo este formado por 513554 especificações válidas. Para 6 tarefas, a explosão combinatorial comprometeu a construção do espaço completo.

Foram realizados testes de desempenho tendo como parâmetros o número de situações do conjunto base. A Figura 8 ilustra a quantidade de caminhos gerados para cada subárvore da árvore da solução para um conjunto de Situações $S = \{s_1, \dots, s_{33}\}$. Já a Figura 9 ilustra o incremento de tempo, obtido com a geração de cada subárvore. Note-se que este incremento diminui e tende a zero visto que a cada nova subárvore a situação raiz da árvore anterior é excluída.

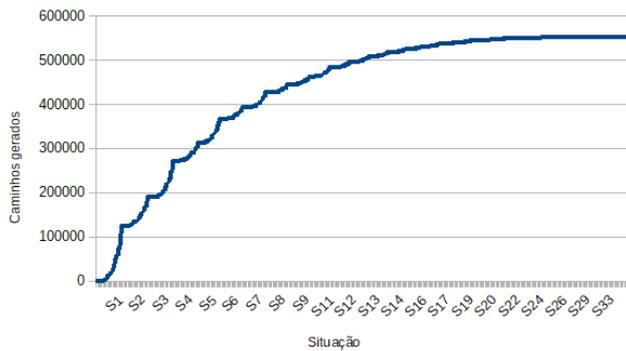


Figura 8. Quantidade de caminhos acumulada com a geração de cada subárvore de S.

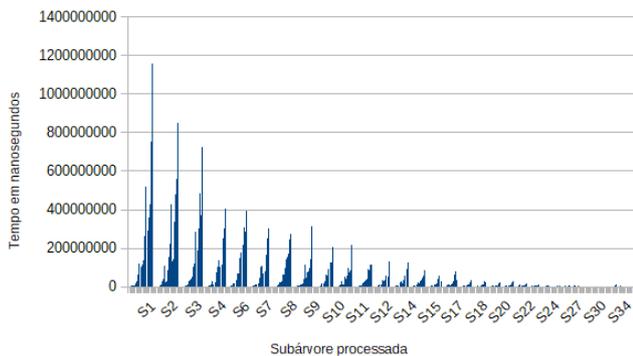


Figura 9. Tempo de processamento para geração de cada subárvore de S.

Embora seja possível explorar espaços um pouco maiores, por meio de soluções como paralelismo e mudança de plataforma, os resultados permitem concluir que o problema de se gerar todo o Espaço de Solução se torna intratável para um número mais expressivo de atividades, e.g. 20, ou para o aumento de tipos de situações. Considerando este aspecto, a alternativa encontrada foi a construção iterativa de modelos, discutida na Seção 4. Esta solução compreende a criação de modelos com um conjunto menor de tarefas. Posteriormente, novas tarefas são adicionadas iterativamente e, a cada passo, uma nova rodada de perguntas podem ser realizadas e os modelos podem ser obtidos com custo computacional baixo.

6. TRABALHOS RELACIONADOS

Alman et al. (2020) apresentam uma proposta de *chatbot* para definir restrições declarativas usando instruções de linguagem natural fornecidas na forma de entrada vocal ou textual por meio de interações com usuários. As restrições suportadas utilizam *Multi-Perspective Declare* (MP-Declare), complementando as restrições de fluxo de controle com dados e perspectivas temporais.

Já van der Aa et al. (2019), (Van der Aa et al., 2018), desenvolveram técnicas personalizadas de Processamento de Linguagem Natural (PNL) que identificam atividades e suas inter-relações a partir de descrições de restrições textuais. Apesar dos avanços na área da PNL, Os autores afirmam que aplicá-la ao BPM apresenta vários desafios

para os pesquisadores de PNL. Em nossa proposta, a perspectiva da linguagem natural não é considerada podendo esta ser vista de forma complementar, visto que restrições obtidas dessa forma podem ser submetidas a validação tendo como base o Espaço de Solução proposto.

Wiśniewski et al. (2018) apresentam um método de composição de processos de negócios onde um conjunto de restrições predefinidas é obtida para gerar um modelo sintético do processo, fornecendo uma especificação simplificada e não reordenada, que descreve as atividades a serem realizadas.

Tendo também como objetivo a determinação de modelos de controle de fluxo de processos, Milani et al. (2013) apresentam um método para extrair modelos a partir de um conjunto de diagramas existentes, que representam diferentes variantes de execução do fluxo de trabalho e propõem uma abordagem denominada consolidação orientada à decomposição. Essa abordagem suporta a modelagem de processos reduzindo inconsistências e redundâncias, possíveis de ocorrer em modelos criados manualmente, mas apresenta limitações inerentes à confiabilidade e validade. Nosso estudo elimina esses problemas através da geração do Espaço de Solução contendo todas as hipóteses de modelos aceitáveis para o processo.

Pode-se observar também a complementaridade do trabalho proposto com outras abordagens de modelagem, tal como Mineração de Processos (Mendling, 2016), que utilizam análise de restrições para geração de modelos a partir arquivos de *logs* de sistemas de informação. Embora a proposta desse trabalho esteja apresentada no contexto de interações com os especialistas de domínio, as técnicas de geração e de validação de modelos propostos podem ser aplicadas também no contexto da mineração, por exemplo, na determinação de restrições em processos a partir da análise de *logs* considerando os testes de asserção propostos na Seção 3.

7. CONCLUSÃO

Este artigo discutiu o apoio à Modelagem de Processos de Negócio por meio de interações com especialistas de domínio, automatizadas por meio de um agente computacional. As interações têm como objetivo determinar modelos descritivos de controle de fluxo de processos de negócio com vistas a auxiliar o trabalho de especialistas modeladores. Considerando este objetivo, o trabalho apresentou um modelo que permite a realização dessas interações. Um protótipo baseado no modelo proposto foi desenvolvido e testado em cenários de modelagem. Os testes e experimentos com o protótipo permitiram verificar a efetividade da proposta que se mostrou capaz de produzir modelos de controle de fluxo válidos, com relação ao modelo de validação semântica estabelecido, e em conformidade, com relação à visão expressa pelo especialista de domínio. Foi também possível verificar a geração de alternativas de modelos, as quais permitem ao modelador um melhor julgamento das possibilidades de modelagem. Em adição, foi estabelecida uma abordagem para mitigar o problema da explosão combinatorial do Espaço de Solução de modelagem, por meio de iterações no processo de modelagem. Na aborda-

gem proposta, novas atividades vão sendo incorporadas ao processo ao longo da interação. Uma limitação do trabalho em seu estado corrente é a composição das perguntas que não refletem a naturalidade de uma entrevista com agentes humanos. Como trabalho futuro, está planejada a análise de alternativas baseadas em técnicas de processamento da linguagem natural, por exemplo, utilizando o ambiente RASA.

AGRADECIMENTOS

Este trabalho foi realizado no contexto do Laboratório de Inteligência Computacional e Sistemas de Informação - LAICSI, do campus Serra do Ifes, e contou com o apoio do Laboratório NERA, o qual gentilmente cedeu acesso ao servidor de processamento adquirido no contexto do projeto *Bibliometria Automatizada: Um estudo de caso sobre o uso de lamas de beneficiamento rochas ornamentais*, coordenado pelo Prof. Dr. Francisco de Assis Boldt, e onde parte dos experimentos foi realizada.

REFERÊNCIAS

- Alman, A., Balder, K.J., Maggi, F.M., and Aa, H.v.d. (2020). Declo: A chatbot for user-friendly specification of declarative process models. In *CEUR Workshop Proceedings*, volume 2673, 122–126. RWTH.
- Costa, M.B. and Tamzalit, D. (2017). Recommendation patterns for business process imperative modeling. In *Proceedings of the Symposium on Applied Computing, SAC '17*, 735–742. ACM, New York, NY, USA. doi: 10.1145/3019612.3019619. URL <http://doi.acm.org/10.1145/3019612.3019619>.
- Fahland, D., Lübke, D., Mendling, J., Reijers, H., Weber, B., Weidlich, M., and Zugal, S. (2009a). Declarative versus imperative process modeling languages: The issue of understandability. In *Enterprise, Business-Process and Information Systems Modeling*, 353–366. Springer.
- Fahland, D., Lübke, D., Mendling, J., Reijers, H., Weber, B., Weidlich, M., and Zugal, S. (2009b). Declarative versus imperative process modeling languages: The issue of understandability. In *Enterprise, Business-Process and Information Systems Modeling*, 353–366. Springer.
- Group, O.M. (2013). Bpmn 2.01 specification. <https://www.omg.org/spec/BPMN/2.0.1>.
- Kannengiesser, U. and Oppl, S. (2015). Business processes to touch: Engaging domain experts in process modelling. In *BPM (Demos)*, 40–44.
- 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*, 42–56. Springer.
- Mendling, J. (2016). A knowledge perspective on business process management. *Wissen im Prozess*, 33.
- Milani, F., Dumas, M., and Matulevičius, R. (2013). Decomposition driven consolidation of process models. In *International Conference on Advanced Information Systems Engineering*, 193–207. Springer.
- Pesic, M., Schonenberg, H., and Van der Aalst, W.M. (2007). Declare: Full support for loosely-structured processes. In *Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International*, 287–287. IEEE.
- Recker, J., Rosemann, M., Indulska, M., and Green, P. (2009). Business process modeling—a comparative analysis. *Journal of the association for information systems*, 10(4), 1.
- Schubert, J. (2012). Constructing and evaluating alternative frames of discernment. *International Journal of Approximate Reasoning*, 53(2), 176–189.
- Tolfo, C., Medeiros, T.S., and Mombach, J.G. (2013). Modelagem de processos com bpm em pequenas empresas: um estudo de caso. *XXXIII Encontro Nacional de Engenharia De Produção, Salvador*.
- Van der Aa, H., Carmona Vargas, J., Leopold, H., Mendling, J., and Padró, L. (2018). Challenges and opportunities of applying natural language processing in business process management. In *COLING 2018: The 27th International Conference on Computational Linguistics: Proceedings of the Conference: August 20-26, 2018 Santa Fe, New Mexico, USA*, 2791–2801. Association for Computational Linguistics.
- van der Aa, H., Di Ciccio, C., Leopold, H., and Reijers, H.A. (2019). Extracting declarative process models from natural language. In *International Conference on Advanced Information Systems Engineering*, 365–382. Springer.
- Verner, L. (2004). The challenge of process discovery. *BPM Trends, May*, 05–04.
- Wiśniewski, P., Kluza, K., and Ligęza, A. (2018). An approach to participatory business process modeling: Bpmn model generation using constraint programming and graph composition. *Applied Sciences*, 8(9). doi: 10.3390/app8091428. URL <https://www.mdpi.com/2076-3417/8/9/1428>.
- Wong, P.Y. and Gibbons, J. (2008). A process semantics for bpmn. In *International Conference on Formal Engineering Methods*, 355–374. Springer.
- Zhou, Z.J., Hu, G.Y., Zhang, B.C., Hu, C.H., Zhou, Z.G., and Qiao, P.L. (2017). A model for hidden behavior prediction of complex systems based on belief rule base and power set. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(9), 1649–1655.