

Modelagem de processos não-estruturados: uma comparação entre BPMN, DCR Graphs e DECLARE

Ana Carolina Claudino (PUC-PR) ana.claudino@pucpr.edu.br

Rafael Araujo Kluska (PUC-PR) rafael@kluska.com.br

Rafael Duarte (PUC-PR) r.duarte@pucr.br

Fernando Deschamps (PUC-PR) fernandodeschamps@pucr.br

Resumo:

O objetivo deste estudo é aprofundar os conhecimentos de modelagem de processos não estruturados, ou seja, processos pouco especificados e geralmente modelados por um conjunto de tarefas e outro de restrições, comparando os três métodos envolvidos para melhor compreender quais são suas principais diferenças. Para isso, foi desenvolvida uma breve revisão bibliográfica sobre métodos e ferramentas de modelagem de processos declarativos e imperativos, seguida de um estudo comparativo constituído pela modelagem de um processo genérico de desenvolvimento de um produto em três ferramentas distintas: BPMN, DCR Graphs e DECLARE. Como resultado, é possível observar que o BPMN dentre as notações de modelagem analisadas, mostra-se mais intuitiva, porém apresenta pontos restritivos do ponto de vista de modelagem para processos não estruturados, que não restringem as atividades de modelagem de processos de negócios, mas requer adaptações para processos não estruturados. Já as ferramentas DCR Graphs e DECLARE, são semelhantes em seus recursos, porém o DCR possibilita expressar restrições com apenas quatro tipos de relações, enquanto com o DECLARE é necessário combinar operadores lógicos e temporais para construí-las.

Palavras chave: Modelagem de processos, Ferramentas de modelagem, Modelagem declarativa, Processos não estruturados

Modeling of unstructured processes: a comparison of BPMN, DCR Graphs and DECLARE

Abstract

The purpose of this study is to deepen the knowledge of modeling unstructured processes, that is, processes that are not specified and are usually modeled by a set of tasks and another of constraints, comparing the three methods involved to better understand their main differences. For this, a brief literature review was developed on methods and tools for modeling declarative and imperative processes, followed by a comparative study of the modeling of a generic product development process in three different tools: BPMN, DCR Graphs and DECLARE. As a result, it is possible to observe that BPMN among the notations of the analyzed model is more intuitive, but presents restrictive points of view from the modeling point for unstructured processes, which don't restrict business process modeling activities, but requires adaptations to unstructured processes. The DCR Graphs and DECLARE tools are similar in their capabilities, however, DCR allows you to express constraints with only four types of relationships, whereas with DECLARE you need to combine logical and temporal operators to construct them.

Key-words: Process modeling, Modeling tools, Declarative modeling, Loosely structured process

1. Introdução

A modelagem de processos é utilizada para representar as atividades, fatores, ambientes, execuções e responsabilidades dentro de um determinado processo. Através desta ferramenta, é possível controlar e prever possíveis ocorrências dentro de um cenário modelado, inserindo atividades sequenciais e restrições que direcionam o que pode ou não ser executado. De acordo com Campos (2013), a modelagem é de grande relevância para estabelecer e melhorar a capacidade operacional e de controle das organizações visando tornar possível o atingimento dos objetivos estratégicos de uma organização ou iniciativa de negócio.

Existem dois tipos de modelos de processos, os imperativos e os declarativos, sendo que o primeiro é o mais comum. Também existem dois tipos distintos de processos que podem ser modelados: processos estruturados e não estruturados. De acordo com Giacomo et al. (2015), processos não estruturados são comumente constituídos por um conjunto de tarefas e outro de restrições, sendo que as sequências de eventos só serão permitidas caso o conjunto de restrições seja cumprido (CARVALHO et al., 2016).

Os pesquisadores Reijers, Slaats e Stahl (2013), apontam que os modelos imperativos expressam sequências de comandos que realizam transformações sobre os dados através de uma abordagem de dentro para fora do processo, desta forma todas sequências de execução possíveis são modeladas explicitamente. Enquanto modelos declarativos, tendem adotar uma abordagem mais externa, descrevendo apenas características essenciais do processo. Na pesquisa de Hildebrandt e Mukkamala (2011) também se ressalta que a principal diferença entre processos declarativos e imperativos é que o fluxo de controle na linguagem declarativa ocorre de maneira implícita, na forma de um conjunto de restrições ou regras, enquanto nos modelos imperativos tal fluxo ocorre de maneira mais explícita, como uma sequência de comandos.

Devido ao alto grau de detalhamento exigido pela linguagem imperativa, os modelos que a utilizam tendem a ser mais suscetíveis à detalhamentos excessivos, tornando-se sistemas pouco flexíveis, o que faz com que este tipo de linguagem seja incapaz de suportar processos pouco estruturados. Esta lacuna foi uma oportunidade para que a modelagem utilizando linguagem declarativa se desenvolvesse, este fato adicionou alguns diferenciais às atividades de modelagem, como: um sistema mais fácil de modelar, menos detalhado, incluindo apenas características importantes inerentes ao processo, proporcionando maior flexibilidade ao permitir a manutenção e alteração do modelo, trazendo maior conforto e expressividade à modelagem.

Este trabalho é parte de um projeto de pesquisa na área de gerenciamento de projetos, busca-se aprofundar nossos conhecimentos referentes às tais notações de modelagem de processos. Para isso, optamos pela utilização de um modelo não estruturado simples para aplicar às três notações escolhidas: BPMN, DCR Graphs e DECLARE. Estas notações apresentam uma linguagem gráfica, tendo seus recursos representados por elementos gráficos.

Para desenvolver o referido estudo, optou-se pela metodologia de pesquisa-ação. Primeiramente uma revisão bibliográfica foi realizada para a aquisição de conhecimento teórico sobre os assuntos e, em seguida, as aplicações práticas tiveram início. O mesmo exemplo foi modelado para cada notação estudada e as diferenças observadas foram registradas. De modo geral, foi possível observar que o BPMN é a notação de mais fácil compreensão dentre as três, pois é composta por fluxogramas que facilitam o entendimento. As notações utilizadas no DCR Graphs e DECLARE são semelhantes entre si, sendo que a principal diferença entre as mesmas é que o DCR possui marcação de tempo e possibilita que as restrições sejam criadas com apenas quatro relações, enquanto o DECLARE possui operadores temporais e lógicos que demandam mais conexões para expressar as mesmas restrições.

2. Metodologia

Para o desenvolvimento das atividades de pesquisa optou-se pela utilização do método de pesquisa conhecido como pesquisa-ação. Assim como sugerido por Turriani e Mello (2012) este método pode ser utilizado quando o pesquisador interfere de forma participativa no desenvolvimento da pesquisa. Nosso problema em questão está relacionado com o entendimento da aplicação dos métodos para a modelagem de processos não estruturados. Alguns pesquisadores afirmam que esta técnica pode ser considerada uma variação do estudo de caso (WESTBROOK, 1995). Porém, diferente do estudo de caso, onde o pesquisador não interfere no objeto de estudo, a pesquisa-ação utiliza de abordagem participativa, ao associar uma ação à resolução de um problema coletivo, os pesquisadores e os participantes estão envolvidos de modo cooperativo ou participativo, na busca pela solução (THIOLLENT, 2007).

É neste exato ponto que a pesquisa foi conduzida. Trabalhando por meio de um processo cíclico, a pesquisa-ação é composta por cinco fases e uma metáfase. Estas fases estão divididas e organizadas de forma a suprir as demandas geradas em um projeto que envolva a construção de conhecimento (COUGHLAN e COUGHLAN, 2002). Na Figura 1 é possível observar a disposição dessas fases.

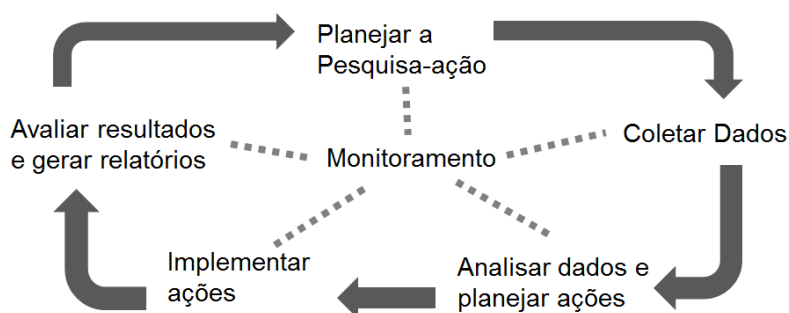


Figura 1 – Estruturação para condução da pesquisa-ação

Fonte: Adaptado de Coughland and Coughlan (2002), p.154., p.154.

As fases consistem em: planejar; coletar dados; analisar dados e planejar ações; implementar ações; avaliar resultados e gerar relatórios. A meta-fase é conhecida como monitoramento. Segundo Coughlan e Coughlan (2002) e Thiollent (2007), esta meta-fase ocorre em todos os ciclos e tem por função o acompanhamento e a documentação do aprendizado.

Planejamento da pesquisa: esta fase ocorre para alinhar os interesses entre os pesquisadores através da realização de um estudo de viabilidade que, segundo Thiollent (2007), permite aos mesmos tomarem a decisão de aceitarem a pesquisa sem criar falsas expectativas. Para isso, um diagnóstico é desenvolvido baseado nos contatos com as partes interessadas, nas características da pesquisa, nas expectativas e nos problemas do estudo.

Coleta de dados: para esta pesquisa os dados foram obtidos de artigos científicos relacionados com a área, além da modelagem de um exemplo desenvolvido em cada notação estudada. De acordo com Thiollent (2007), independente das técnicas utilizadas para esta obtenção, os pesquisadores e participantes procuram apenas as informações que são julgadas necessárias para o andamento da pesquisa.

Analisar dados e planejar ação: por se tratar de uma pesquisa-ação de cunho educacional, a análise dos dados obtidos está diretamente relacionada à aprendizagem. Ainda de acordo com Thiollent (2007), para corresponder ao conjunto de seus objetivos, a pesquisa-ação deve se concretizar de alguma forma de ação planejada, sendo que a formulação de um plano de ação

constitui uma exigência fundamental. Para esta pesquisa, o plano de ação é constituído pela elaboração de um quadro comparativo entre as três notações estudadas.

Implementat as ações: Esta fase foram executadas as ações planejadas. Realizou-se revisões de literatura referente às motações estudadas (sessão 3 deste artigo) e na sequencia aplicoou-se um processo exemplo à notações (sessão 4, deste artigo).

Avaliação dos resultados e geração de relatórios: Após a conclusão dos trabalhos bucou-se avaliar as notações estudadas e foi desenvolvido um quadro comparativo entre as três notações estudadas, o resultado obtido está declarado na sessão 5 deste artigo.

3. Referencial teórico

Nesta seção são apresentadas as notações estudas a para representar processos não estruturados, são elas: BPMN (Business Process Model and Notation), DCR Graphs (Dynamic Condition Response) e DECLARE. Para facilitar a compreensão de cada notação buscou-se, de forma sucinta, apresentar os principais elementos de cada notação, juntamente com sua descrição e aplicações.

3.1 BPMN

O BPMN é uma notação para modelagem de processos cujo objetivo principal é fornecer uma linguagem de simples entendimento para a modelagem de negócios. Atualmente está na versão 2.0 e é mantida pela OMG – Object Management Group, sendo esta a única entidade que pode autorizar desenvolvedores, fornecedores e vendedores de softwares para certificação (OMG, 2011). Esta notação utiliza-se de elementos gráficos para a contrução de um workflow que representa por meio de seus elementos: o que, onde, quando, por que, como e por quem o processo é realizado (CBOCKTM, 2014)., este é um facilitador que possibilita até mesmo os usuários mais novos consigam se adaptar rapidamente. Por outro lado, uma de suas restrições é a modelagem de processos de softwares, sendo necessário realizar algumas adaptações no processo. De acordo com Kluska et al. (2015), existem cinco elementos principais que pertencem ou estão ligados ao BPMN: negócios, melhoria contínua, processos de negócios, modelagem de processos e automação de processos.

Para modelar, os utilizadores desta notação contam com diversos recursos, tais como: piscinas (swin lanes), objetos de fluxo (atividades, gateways e eventos), objetos de conexão (fluxo de mensagens, sequência e associação), objetos de dados, grupos e anotações. Tais recursos estão brevemente descritos abaixo:

a) Atividades fazem parte do grupo de objetos de fluxo e possuem algumas variações para atender melhor as demandas dos projetos.

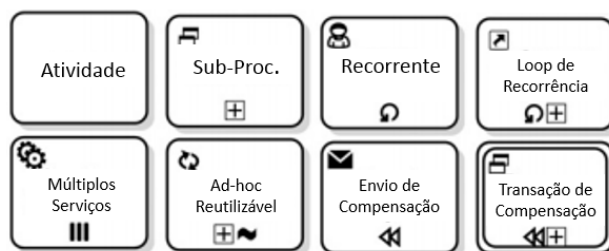


Figura 2 – Objetos de Fluxo: Atividades.

Fonte: Adaptado de CHINOSI e TROMBETTA (2012)

Como pode ser observado na Figura 2, as tarefas apresentam variações devido às suas aplicações, visto que são empregadas quando o trabalho no processo não pode ser quebrado até

um nível mais detalhado. Uma tarefa é considerada uma atividade dentro de um fluxo de processo (ABPMP, 2014).

b) Eventos também pertencem ao grupo dos objetos de fluxo. São classificados como iniciais, intermediários e finais e possuem diversas utilizações, como pode ser observado na Figura 3. Todo processo dentro do BPMN deve iniciar e finalizar com um evento;

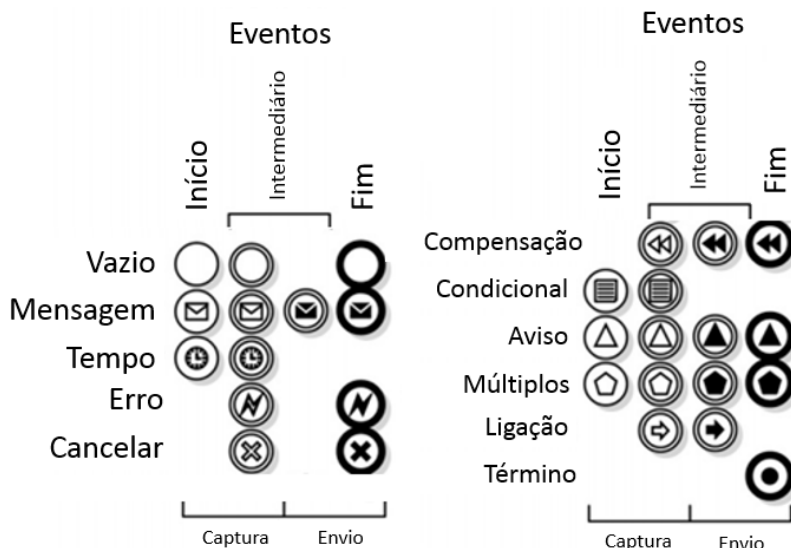


Figura 3 – Objetos de Fluxo: Eventos
Fonte: Adaptado de CHINOSI e TROMBETTA (2012)

Eventos de captura referem-se ao início do processo e ao recebimento de eventos intermediários, enquanto os eventos de envio representam os resultados (saídas do processo), como exposto na Figura 3. (ABPMP, 2014).

c) O último componente do grupo dos objetos de fluxo é o conjunto de Gateways, representado na Figura 4. Estes elementos são utilizados para tomadas de decisão dentro de um processo;



Figura 4 – Objetos de Fluxo: Gateways
Fonte: Adaptado de CHINOSI e TROMBETTA (2012)

As interações entre as sequências de fluxos são controladas pelos gateways expostos na Figura 4. Se o fluxo do processo não precisa ser controlado, então a utilização de gateways também não é necessária. Os gateways são mecanismos de entrada que permitem ou não a passagem de dados pelo fluxo. (ABPMP, 2014).

d) Abaixo estão apresentados os outros três grupos principais do BPMN. Piscinas são a base do modelo do processo, é neste elemento que todas as atividades, eventos, gateways, conexões e artefatos são incluídos. Já o grupo dos Objetos de Conexão é composto por três tipos de ligações (sequencial, de mensagem ou de associação) que conectam eventos às atividades, gateways e artefatos.

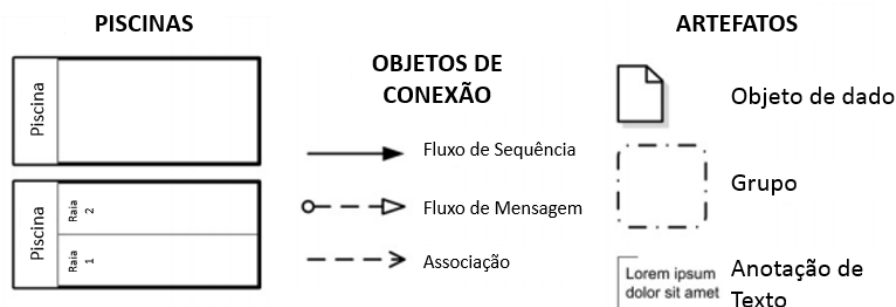


Figura 5 – Piscinas, Objetos de Conexão e Artefatos
Fonte: Adaptado de CHINOSI e TROMBETTA (2012)

Por fim, há o grupo dos Artefatos, indicado na Figura 5, composto por objetos de dados, grupos e anotações, permitindo que o usuário detalhe e represente dados e/ou informações importantes do processo.

3.2 DCR Graphs

O DCR Graphs nada mais é do que uma notação que possui linguagem gráfica, baseada em acontecimentos, comumente aplicada para modelagem de processos declarativos. Neste modelo, os processos possuem atividades que podem ser conectadas por quatro tipos de relações (condição, resposta, inclusão e exclusão), ilustradas abaixo:

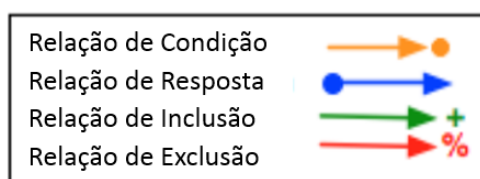


Figura 6 – Notações no DCR Graphs
Fonte: Adaptado de HILDEBRANDT e MUKKAMALA (2011)

De acordo com Schaidt (2017), As estruturas definidas no DCR Graphs permitem definir uma sequência entre conjuntos de eventos e um conjunto de relações. Sendo que as sequências de tais eventos devem obedecer ao conjunto de relações exposto na Figura 6, ou seja, apenas as sequências que obedecem as relações podem ocorrer.

A relação de condição (*condition*) estabelece que o evento A deve ocorrer ao menos uma vez antes do evento B. Já a relação de resposta (*response*) indica que se o evento A ocorrer, então o evento B precisa ocorrer em seguida. A relação de inclusão (*include*) é empregada para tornar uma atividade novamente disponível após uma exclusão. Por outro lado, a relação de exclusão (*exclude*) define que uma vez que a atividade A tenha sido executada, a atividade B não poderá mais acontecer.

3.3 DECLARE

De acordo com Pesic, Schonenberg e Aalst (2007), o Declare é um protótipo de sistema de gerenciamento de fluxo de trabalho que utiliza uma modelagem de processos baseada em restrições para a construção de modelos declarativos de processos pouco estruturados. O maior diferencial deste recurso é que com o Declare é possível criar restrições para cada tipo de relação dentro do processo. Cada modelo de restrição possui um nome único, semântica especificada e representação gráfica, sendo que a lógica empregada (*Linear Temporal Logic – LTL*) utiliza vários operadores temporais - além dos operadores lógicos clássicos.

Grupo	Relação	Definição
Existência	Ao menos	A atividade deve ser executada ao menos x vezes
	No Máximo	A atividade pode ser executada no máximo x vezes
	Exatamente	A atividade deve ocorrer exatamente x vezes
	Posição	Define a posição da atividade (final ou inicial)
Relação	Não ordenado	Para relações de coexistência ou resposta que não precisam necessariamente seguir uma ordem
	Ordenado	Para relações de coexistência que necessitam seguir uma ordem acontecimento
Escolha	Simple	Ao menos uma das atividades envolvidas na decisão deve ser executada
	Exclusiva	Somente uma das atividades envolvidas na decisão deve ser executada
Relação de Negação	Não ordenado	Relação de não coexistência
	Ordenado	Relação de não sucessão
	Cadeia	Duas atividades não podem ser executadas próximas

Tabela 1 – Principais relações no Declare.

Fonte: O autor, 2017

Como pode ser observado na Tabela 1, de maneira resumida, as relações no Declare são classificadas em quatro grupos principais: Existência (*Existence*), Relação (*Relation*), Escolha (*Choice*) e Negação (*Negative*). Os dados para a construção deste comparativo foram retirados do DECLARE (versão 2.2.0).

4. Desenvolvimento

Nesta seção será apresentada a modelagem de um processo não estruturado usando-se cada uma das três notações apresentadas no referencial teórico. O processo escolhido foi o processo de desenvolvimento de produto. Este processo consiste basicamente em um levantamento dos dados de mercado por parte do Marketing e Diretoria, que repassam uma solicitação de desenvolvimento de produto para o setor de Engenharia de Produto. Este, por sua vez, é responsável por analisar a viabilidade do projeto, elaborar os desenhos, enviar ordens de produção de peças para teste, montar e testar um modelo protótipo, replanejar as peças caso seja necessário e, por fim, após o produto teste ter sido validado, criar um catálogo de peças e distribuir para a produção.

4.1 BPMN

Para modelar o processo de desenvolvimento de um novo produto no BPMN, foram utilizadas duas piscinas (uma para os setores de Pesquisa e Desenvolvimento, Marketing e Diretoria, e outra para o processo de desenvolvimento do item), sendo que uma foi dividida em duas raias (Engenharia de Produto e Produção). Para representar as etapas do processo, foram inseridas atividades convencionais, sem alterações. Já as decisões foram expressadas por gateways inclusivos que fornecem ao sistema duas opções para determinadas situações. Os elementos de conexão foram empregados segundo os critérios sequenciais (linha cheia) – para fluxos dentro da mesma swimlane – e de mensagem (tracejado) – para fluxos de mensagem e que conectam diferentes swimlanes. Por fim, o único artefato utilizado foi um objeto de dado para representar o Catálogo de peças para fabricação.

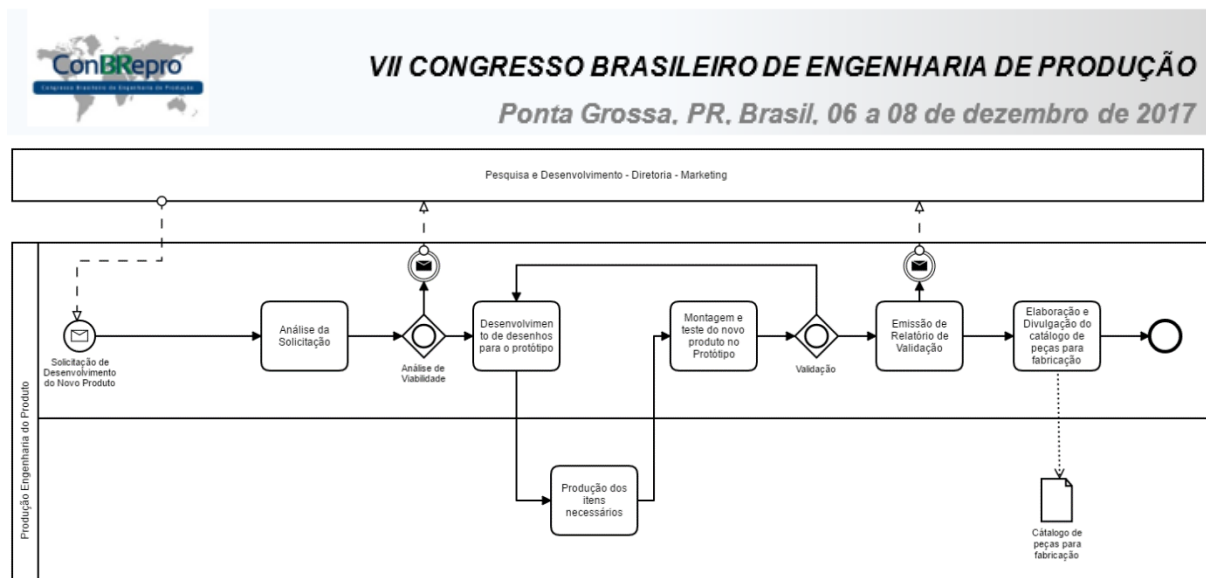


Figura 7 – Modelagem do Processo em BPMN

Fonte: O autor, 2017

Brevemente, o processo representado na Figura 7 acontece da seguinte forma: uma mensagem solicitando o desenvolvimento de um novo item é encaminhada da swimlane “Diretoria” para o pool da “Engenharia do Produto”. É neste pool que todas as atividades ocorrem, exceto a “Produção dos Itens Necessários”, que ocorre no pool da Produção. A partir da análise da solicitação, a Engenharia do Produto pode aceitar ou rejeitar a proposta através do primeiro gateway. Caso rejeite, uma mensagem é enviada para os responsáveis pela solicitação (primeira swimlane). Caso aceite, segue para as atividades de desenvolvimento de desenhos, envio para a produção e montagem/teste no protótipo. Após a etapa de testes há mais um gateway, chamado “Validação”, onde é possível decidir se o produto está pronto para ser validado ou se necessita de mais alguns ajustes. Uma vez validado, um relatório de validação é elaborado e enviado para a gerência e o catálogo de peças para fabricação é confeccionado e emitido para a produção. Caso contrário, os desenhos deverão ser replanejados, enviados novamente para a produção e testados mais uma vez no protótipo.

4.2 DCR Graphs

Para exemplificar o processo de desenvolvimento de um novo produto, foram utilizadas três das quatro notações apresentadas. Abaixo é possível conferir o modelo em questão e as descrições das relações empregadas:

Relação de condição (*condition*) foi aplicada nos seguintes casos: A Solicitação de Novo Produto deve ocorrer ao menos uma vez antes do Estudo de Viabilidade. Assim como o Estudo deve ser executado ao menos uma vez antes do Replanejamento e da Elaboração dos desenhos, a Elaboração antes da Produção dos itens, a Produção antes da Montagem e os Testes antes da Criação e Distribuição do catálogo de Peças.

Relação de resposta (*response*) utilizada da seguinte maneira: sempre que a Solicitação de um Novo Produto for requerida, um Estudo de Viabilidade deverá ser executado. Assim como a Montagem e o teste do Protótipo deve ocorrer após a Produção dos itens, que, por sua vez, ocorre após a Elaboração dos Desenhos.

Relação de exclusão (*exclude*) aplicada com o seguinte propósito: Uma vez que o Estudo de Viabilidade tenha sido concluído, não é possível solicitar um novo produto. Também ao Elaborar os Desenhos para a Produção, não é mais possível Estudar a Viabilidade. Por último, após a Distribuição do Catálogo de peças para a produção, não é mais possível Replanejar, Elaborar desenhos, Produzir Itens e Montar/Testar no protótipo.

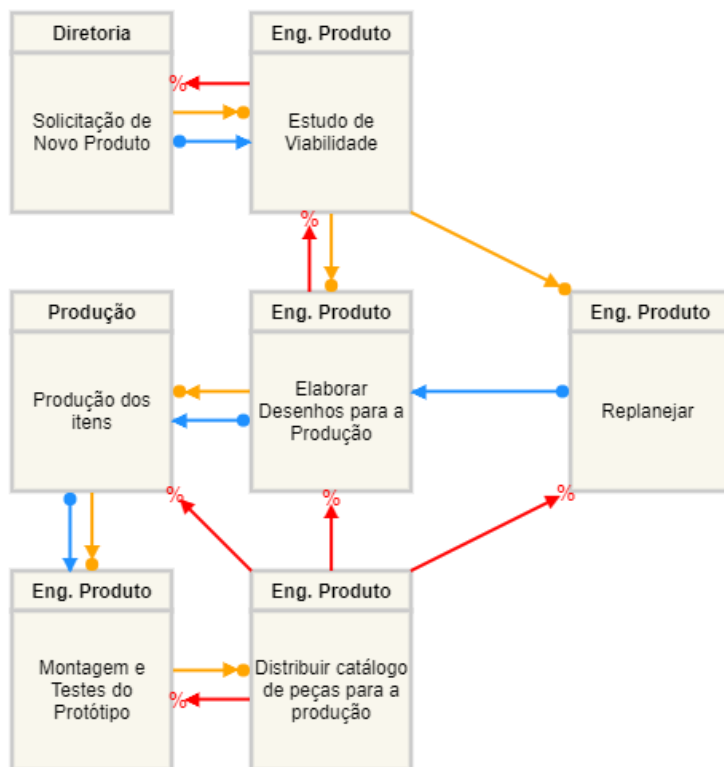


Figura 8 – Modelagem do Processo com o DCR Graphs
Fonte: O autor, 2017

Como pode ser observado na Figura 8, as atividades são representadas por retângulos com os respectivos nomes dos grupos responsáveis, sendo conectadas pelas relações descritas anteriormente, de maneira a construir um fluxo lógico para o processo.

4.3 DECLARE

Para o exemplo de modelagem do processo de desenvolvimento de um novo produto utilizando o DECLARE, foram utilizadas as seguintes notações:

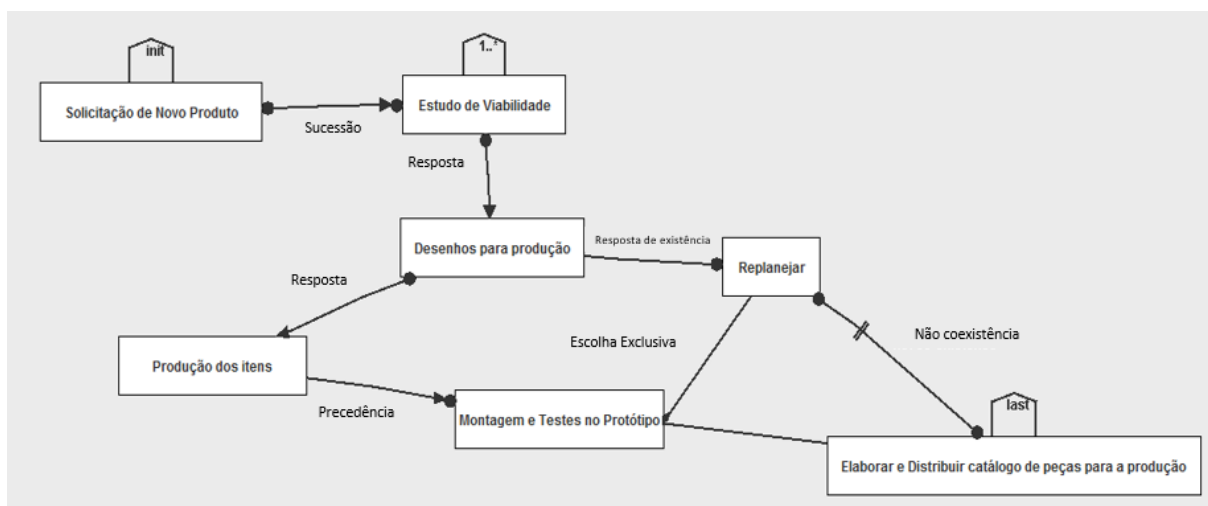


Figura 9 – Modelagem do Processo com o DECLARE
Fonte: O autor, 2017

Uma das maiores diferenças na modelagem com o Declare é o fato de podermos utilizar operadores temporais, como os indicadores de início e fim sobre as atividades “Solicitação de

Novo Produto” e “Elaborar e Distribuir catálogo de peças para a produção” e como a condição de existência sobre a atividade “Estudo de Viabilidade”, que indica que a mesma deve ocorrer ao menos uma vez durante o processo exposto na Figura 9.

As ligações de sucessão (*succession*) indicam que a atividade A é sucedida pela B, de modo que B pode ocorrer apenas se A já tiver acontecido. No exemplo, a ligação é empregada para restringir o acontecimento da atividade “Estudo de Viabilidade”, que só pode ocorrer após a “Solicitação de Novo Produto”. Seguindo a mesma lógica, existem as ligações de precessão (*precedence*) indicando que uma atividade B deve ser precedida pela atividade A, não podendo ocorrer antes disso. Este é o caso da “Montagem e Testes no Protótipo” e “Produção dos Itens”.

Ligações de resposta (*response*) são similares às ligações de precedência, diferindo apenas no fato de que em relações precedentes a atividade B necessita ser executada logo após a atividade A, enquanto em relações de resposta, a atividade B deve ser executada após A, mas não necessariamente imediatamente após A.

A relação que existe entre as atividades “Replanejar” e “Desenhos para Produção” é chamada de resposta de existência (*responded existence*). Tal ligação ocorre quando para que uma atividade A seja executada (ao menos uma vez), uma atividade B deve ter acontecido antes (ao menos uma vez). Ou seja, para que se possa executar a atividade “Replanejar”, a atividade “Desenhos para a Produção” deve ter acontecido ao menos uma vez. Por outro lado, quando ocorre uma ligação de não coexistência (*not co-existence*), significa dizer que duas atividades não podem ser executadas ao mesmo tempo. Como é o caso da atividade “Replanejar” e “Elaborar e Distribuir catálogo de peças para a produção”.

Por fim, a ligação de escolha exclusiva (*exclusive choice*) do modelo significa que apenas uma das atividades da sequência pode ser escolhida, como é o caso das atividades “Replanejar” e “Elaborar e Distribuir catálogo de peças para a produção” após “Montagem e Testes no Protótipo”.

5. Análises e discussões

A fim de expor melhor as principais diferenças observadas durante este estudo, as observações foram dispostas na Tabela 2. As ferramentas foram comparadas no que diz respeito à descrição geral, desenvolvimento de relações e restrições e diferenciais inerentes à cada uma.

Considerações	BPMN	DCR Graphs	Declare
Descrição	Ferramenta de simples entendimento, possui o sistema mais intuitivo de modelar	Modelo composto por atividades, relações e marcação de tempo	Composto por atividades e restrições
Relações e Restrições	Expressas em fluxograma por meio de objetos de conexão, fluxo e artefatos	Expressa as restrições utilizando apenas suas quatro relações básicas	Necessário combinar operadores temporais e lógicos para obter as ligações desejadas
Diferenciais	Único entre os três modelos que permite modelagem por fluxograma. Porém, não define tão explicitamente as restrições entre as atividades	Modelo mais acessível para modelar, visto que está disponível online (dcrgraphs.net) sem necessidade de realizar download	Seus operadores lógicos e temporais permitem um modelo mais assertivo, incluindo limitações de tempo de ocorrência

Tabela 2 – Resumo Comparativo das Ferramentas
Fonte: O autor, 2017

Como pode ser observado na Tabela 2, apesar de possuir muitos objetos, O BPMN realmente cumpre com seu objetivo de ser uma ferramenta de simples entendimento. Comparado aos

outros dois modelos, o BPMN possui o sistema mais instintivo de se desenvolver, visto que seu fluxograma é um processo muito lógico. Por outro lado, neste tipo de modelagem não é possível definir explicitamente quais são os tipos de relação entre as atividades (precedência, sucessão, coexistência, etc.), como ocorre nos outros modelos.

De acordo com Reijers, Slaats e Stahl (2013), o modelo utilizado no Declare é composto por atividades e restrições, enquanto o DCR é composto por atividades, relações e marcação de tempo de execução. Ao modelar o mesmo processo nos três sistemas diferentes (Declare, DCR e BPMN) é possível notar que as diferenças vão além de simples definições.

A princípio, o Declare e o DCR podem parecer ferramentas muito similares, mas possuem diferenças significativas no que diz respeito à construção de restrições. Enquanto no DCR é possível expressar todas as restrições utilizando apenas as quatro relações básicas apresentadas, no Declare é necessário combinar operações temporais e lógicas para obter as ligações desejadas. Por outro lado, a possibilidade de utilizar tais operadores temporais e lógicos é um ponto exclusivo do Declare, o que permite a construção de um modelo mais exato no que diz respeito à execução de atividades, como a opção de marcar quantas vezes uma atividade deve ocorrer ao mínimo ou ao máximo no processo.

6. Conclusão

O presente artigo foi elaborado como resultado de uma pesquisa pertencente ao Programa Institucional de Bolsas em Desenvolvimento Tecnológico e Inovação (PIBITI) da Pontifícia Universidade Católica do Paraná (PUC-PR).

O objetivo principal deste estudo é proporcionar a aquisição de novos conhecimentos acerca das notações representadas. Apesar do resultado ter sido considerado satisfatório pela equipe de pesquisa, pois atingiu o objetivo de desmistificar as notações e possibilitar um entendimento melhor acerca da modelagem de processos não estruturados, alguns recursos das notações deixaram de ser explorados, como por exemplo, a marcação de tempo no DCR Graphs. Um dos fatores limitantes foi a inexperience com as notações em questão, o que exigiu um estudo inicial dos detalhes de cada tipo de modelagem. Porém, a partir deste primeiro resultado é possível modelar processos mais complexos, aplicando mais recursos para explorar os benefícios de cada notação em seu todo.

Dentre as mais variadas formas de representação de processos, as notações de representação de processos são um conjunto de conhecimento que permite a padronização da linguagem de representação. Possibilitando um maior entendimento sobre o processo modelado. Sabe-se que as notações são estruturas limitadas, pois apresentam um número finito de elementos de representação. Porém essa característica é um fator positivo que facilita o processo de representação dos processos e também o entendimento dos processos modelados.

Referências

ABPMP. *BPM CBOKTM V3.0.* Guide to the business process management common body of knowledge. 2. ed. Brasil: Association of Business Process Management Professionals Brasil, 2014. OMG, O. M. G. Business process model and notation (BPMN) OMG, 2011. Disponível em: <http://www.omg.org/spec/BPMN/2.0>

CAMPOS, A. L. N. *Modelagem de processos com BPMN.* Rio de Janeiro: Brasport Livros e Multimídia, 2013

CARVALHO, R. M. de et al. *Comparing condec to cmmn: Towards a common language for flexible processes.* In: IEEE. Model-Driven Engineering and Software Development (MODELSWARD), 2016 4th International Conference on. [S.l.], 2016. p. 233–240.

CHINOSI, M.; TROMBETTA, A. *BPMN: An Introduction to the Standard*. Computer Standards & Interfaces, [s.l.], v.34, n.1, p. 124-134, jan. 2012. Elsevier BV. <http://dx.doi.org/10.1016/j.csi.2011.06.002>.

COUGHLAN, P.; COUGHLAN, D. *Action research for operations management*. International Journal Of Operations & Production Management, [s.l.], v. 22, n. 2, p.220-240, fev. 2002. Emerald. <http://dx.doi.org/10.1108/01443570210417515>.

GIACOMO, G. D. et al. *Declarative process modeling in bpmn*. In: SPRINGER. International Conference on Advanced Information Systems Engineering. [S.l.], 2015. p. 84–100.

HILDEBRANDT, T.T.; MUKKAMALA, R.R. *Declarative Event-Based Workflow as Distributed Dynamic Condition Response Graphs*. Eletronic Proceedings In Theoretical Computer Science,[s.l.], v.69, p.59-73, 18 out. 2011. Open Publishing Association. <http://dx.doi.org/10.4204/eptcs.69.5>.

KLUSKA, R. A.; LIMA, E. P.; COSTA, S. E. G. *Uma Proposta de Estrutura E Utilização Do Gerenciamento de Processos de Negócio (BPM)*. Revista Produção Online, 2015 (3): 886–913. doi:10.14488/1676-1901.v15i3.1867.

PESIC, M.; SCHONENBERG, H.; AALST, W.M. *Full Support for Loosely-Structured Processes*. 11th Ieee International Enterprise Distributed Object Computing Conference (edoc 2007), [s.l.], p.287-298, out. 2007. IEEE. <http://dx.doi.org/10.1109/edoc.2007.4384001>.

REIJERS, H. A.; SLAATS, T.; STAHL C. *Declarative Modeling-An Academic Dream or the Future for BPM?* Lecture Notes In Computer Science, [s.l.], p.307-322, 2013. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-40176-3_26.

SCHADT, S. *Variability management in constraint-based processes: contributions to selection of process variants*. 2017. 239 f. Tese (Doutorado) - Curso de Engenharia de Produção e Sistemas, Programa de Pós-graduação em Engenharia de Produção e Sistemas, Pontifícia Universidade Católica do Paraná - Pucpr, Curitiba, 2017.

THIOLLENT, M. *Metodologia da Pesquisa-Ação*. 2007.São Paulo – SP: Cortez.

TURRONI, J. P.; MELLO, C. H. P. *Pesquisa-Ação na Engenharia de Produção*. In Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações, 2 ed, 149-166. 2012. São Apaulo – SP.

Variabilit

WESTBROOK, R. *Action research: a new paradigm for research in production and operations management*. International Journal Of Operations & Production Management, [s.l.], v. 15, n. 12, p.6-20, dez. 1995. Emerald. <http://dx.doi.org/10.1108/01443579510104466>.