

Capítulo

1

Introdução à Modelagem de Processos de Negócio em BPMN 2.0 e à Automação em BPMS

Lucineia Heloisa Thom, Diego Toralles Avila

Resumo

O gerenciamento de processos de negócio (Business Process Management - BPM) é uma disciplina que visa a redução de tempo, erros e redundância na execução dos processos, além de maior controle destes. O ciclo de vida de BPM inclui as fases de identificação, descoberta, análise, redesenho, implementação, monitoramento e controle de processos. Em particular, a fase de descoberta de processos é fundamental não apenas para a automatização, mas também para a documentação de processos cujo conhecimento está centrado principalmente naqueles que os executam. O presente capítulo introduz BPM, conceitos fundamentais da disciplina, assim como seu ciclo de vida. São descritos os elementos básicos e avançados da Notação e Modelo de Processos de Negócio (Business Process Model and Notation - BPMN 2.0), incluindo tópicos sobre qualidade em modelagem de processo. Também será discutida a automação de modelos de processo de negócio em Sistemas de Gerenciamento de Processos de Negócio (Business Process Management Systems - BPMS), com ênfase na arquitetura de BPMS, transformação de modelos ao nível conceitual para modelos de execução e tipos de BPMS existentes.

Abstract

Business Process Management (BPM) aims to reduce time, errors and redundancy in the execution of processes, in addition to an increasing control of these. The BPM lifecycle includes phases for identifying, modeling, executing and monitoring processes. In particular, the process modeling phase is fundamental not only for automatization but also to the documentation of processes whose knowledge is centered mainly on those who execute them. This chapter introduces BPM, presenting the main concepts of the discipline as well as the BPM lifecycle. It also discusses the automation of business process models through Business Process Management Systems (BPMS), with emphasis in the architecture of the BPMS, the transformation of process models in executable models and the existing types of BPMS.

1.1. Gerenciamento de Processos de Negócio

O Gerenciamento de Processos de Negócio (BPM - *Business Process Management*) é uma disciplina relacionada as áreas de computação, administração de empresas, engenharia de produção e sistemas de informação. BPM tem sido adotado gradativamente pelas organizações que objetivam maior documentação e melhoria de seus processos de negócio, além da automação destes [Dumas et al. 2018]. Através da sua aplicação, BPM torna os processos de negócio mais efetivos, eficientes e adaptáveis, o que, por sua vez, melhora a produtividade e reduz custos [Van Der Aalst 2013].

BPM é um conjunto de métodos, técnicas e ferramentas para descoberta, análise, redesenho, execução e monitoramento de processos de negócio [Dumas et al. 2018]. Um processo de negócio é o trabalho que uma organização realiza quando ela manufatura os seus produtos ou oferece os seus serviços. A figura 1.1 apresenta um processo relacionado ao serviço de atendimento emergencial de um hospital. Neste contexto, o pronto-atendimento de um hospital analisa o nível de risco do paciente. Após, o atendimento deste paciente é classificado de acordo com uma prioridade ao chegar no hospital. Em seguida, o paciente aguarda até ser chamado por uma enfermeira que realiza a triagem do paciente. Realizada a triagem, o paciente é encaminhado, pela enfermeira a um médico geral, o qual realiza uma consulta que resulta no diagnóstico do paciente, finalizando o processo de atendimento.

Este capítulo descreve a disciplina de BPM. Em particular, descreve o ciclo de vida do BPM, respectivos papéis envolvidos, assim como principais aplicações práticas de BPM. O capítulo ainda apresenta aspectos de modelagem, com ênfase na descoberta de processos, na Notação e Modelo de Processos de Negócio (BPMN). Após, discute aspectos relacionados a qualidade de modelos de processo. Finalmente, o capítulo explora a automação de processos, discutindo a transformação de modelos de negócio em modelos de execução, assim como os principais Sistemas de Gerenciamento de Processos (BMPS - *Business Process Management Systems*) existentes.

1.1.1. Ciclo de Vida de BPM

BPM é frequentemente estudado através do seu ciclo de vida, que determina uma forma estruturada para se analisar como um processo pode ser gerenciado e de que maneira e em que momento a tecnologia pode ajudar nessa tarefa [Dumas et al. 2012]. O ciclo de vida objetiva melhor documentação e padronização de processos, além do aumento da eficiência e da qualidade de sua execução [Thom et al. 2009]. A literatura de BPM apresenta ciclos de vida clássicos ([Weske 2019]) e ciclos de vida com suporte a adaptação dinâmica ([Weber et al. 2009]). Entretanto, neste capítulo é apresentado o ciclo de vida definido por [Dumas et al. 2018] (Figura 1.3) que inclui seis fases, definidas nesta seção.

Na fase de *identificação de processo*, um problema de negócio é definido. Com base neste problema, os principais processos executados na organização, relacionados a este problema são identificados. O resultado desta fase é uma *arquitetura de processos* que resume estes processos e as relações entre eles. Ou seja, a identificação de processos envolve duas etapas: i) definição da arquitetura do processo e; ii) seleção de processos. A definição da arquitetura do processo (também denominada designação) objetiva compreender os processos em que uma organização está envolvida, bem como as relações entre

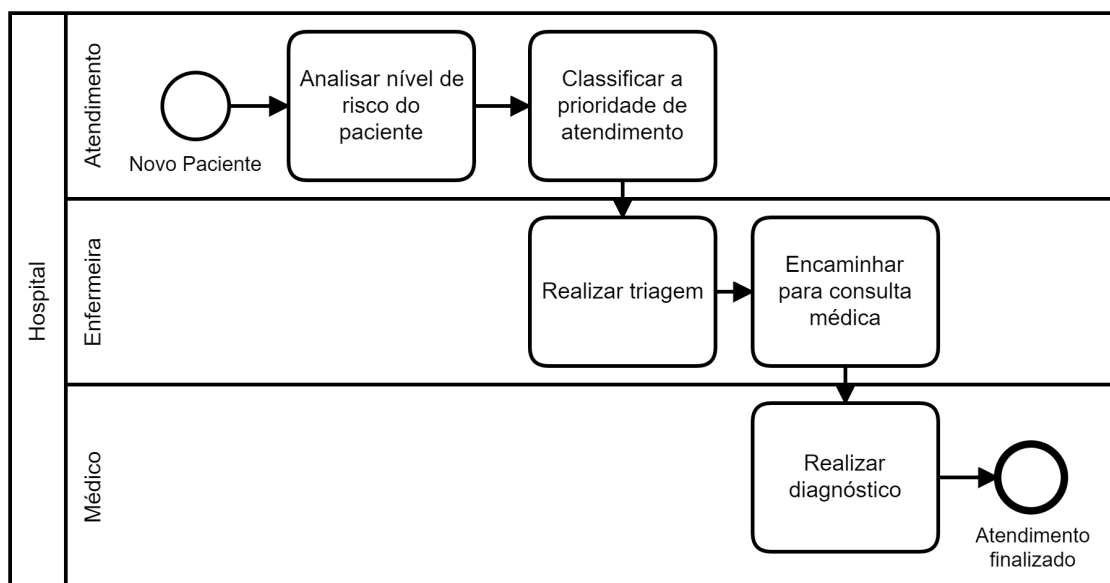


Figura 1.1. Exemplo de processo de atendimento emergencial.

estes processos. A seleção de processos, por sua vez, prioriza os processos para as demais atividades do ciclo de vida de BPM. Importante observar que a definição da arquitetura de processos geralmente ocorre de maneira descendente, conforme ilustrado pela pirâmide na Figura 1.2. No primeiro nível da pirâmide (*process landscape*) aparecem os processos que constituem a cadeia de valor da organização. Este nível, geralmente não inclui modelos de processo em uma notação, tal como a BPMN 2.0. Já os níveis 2 e 3, respectivamente, envolvem modelagem de processos em BPMN 2.0 e em diferentes níveis de detalhamento.

Na fase de *descoberta de processo*, com base na arquitetura de processos criada na fase de *identificação de processos*, geralmente o analista de processos modela, em alguma notação para modelagem de processos (ex.: BPMN 2.0) os processos-chave da organização ou que apresentam algum problema em sua execução (ex.: *deadlocks*, gargalos). O resultado desta fase é o modelo *as-is* do processo que representa como o processo é executado na organização. Esta fase também é denominada na literatura *modelagem de processos*. Entretanto, conforme [Dumas et al. 2018] o termo descoberta de processos é mais apropriado, pois os processos já existem na organização, mesmo que sem documentação. Caso seja uma organização nova e nenhum processo exista, as fases de descoberta e análise não são necessárias. Neste caso o processo deve ser desenhado ao invés de redesenhado.

Na fase de *análise de processo* são identificados problemas e aspectos de melhoria, relacionados ao processo (ex.: tarefas que nunca são executadas, mas que existem no modelo; sobrecarga de tarefas em papéis específicos do processo). Tais problemas e aspectos de melhoria são priorizados, conforme seu impacto para o processo e, em alguns casos, em relação ao tempo e esforço necessários para solucioná-los. O resultado desta fase é de fundamental importância para a etapa de redesenho do processo.

As melhorias identificadas na fase de análise do processo são consideradas na fase

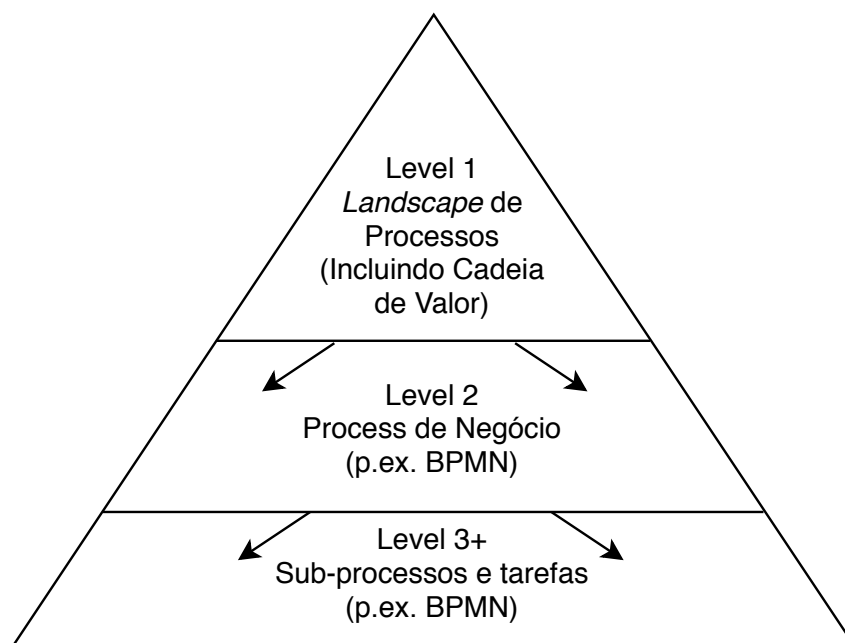


Figura 1.2. Arquitetura de processos, adaptado de [Dumas et al. 2018]

de *redesenho de processo*, onde um novo modelo de processo melhorado é desenhado, resultando no modelo *to-be*. Conforme [Dumas et al. 2018], o processo de redesenho considera três níveis de abstração: métodos (coleção de abordagens para solução de um problema), técnicas (ex.: técnicas para modelar e validar o processo, tal como IDEF3, Atos da Fala, Petri Nets, etc), e ferramentas (um software que suporta uma ou mais técnicas). Em [Biazus et al. 2019], os autores apresentam uma abordagem para recomendar recursos (ex.: software para executar uma tarefa) em tempo de modelagem. Os autores propõem que a recomendação dos recursos seja realizada no final da fase de redesenho do processo, tendo como base outros modelos de processo existentes na organização e sobre os quais existem recursos disponíveis na organização. O objetivo da recomendação ocorrer na fase de redesenho é possibilitar que o analista de processo possa verificar se a organização possui os recursos necessários para executar o processo e, posteriormente implementá-lo. Se os recursos necessários não estiverem disponíveis, a organização pode considerar adquiri-los ou recursos alternativos precisam ser considerados. Ou seja, o modelo de processo não deve seguir para a fase de implementação, caso os recursos necessários para a sua execução não estejam disponíveis.

Na fase de *implementação*, são realizadas alterações no modelo *as-is* para transformá-lo em um modelo *to-be*. Esta fase foca em dois aspectos: gerenciamento de mudança organizacional e automação de processo. O gerenciamento da mudança organizacional se refere a mudança do modo de trabalho dos participantes do processo. Automação de processo se refere ao desenvolvimento e a execução de sistemas de informação para suportar o modelo *to-be*. O modelo *to-be* é automatizado através de um sistema de gerenciamento de processos de negócio (*Business Process Management System - BPMS* [Dumas et al. 2018], resultando em um modelo executável.

Finalmente, na fase de *controle e monitoramento*, são extraídas métricas de perfor-

mance, referentes a execução do processo [Weske 2012]. Tais métricas são fundamentais para verificar a performance do processo e podem ser utilizadas pelos gestores das organizações para acompanhar a execução dos processos. As análises realizadas nesta etapa são fundamentais para definir se o ciclo de vida de BPM será executado novamente.

O ciclo de vida de BPM envolve diversos *stakeholders* [Dumas et al. 2018]:

- *Gerente de Time*. O gerente de time pode ser representado por diversos representantes da organização, incluindo o diretor-executivo que é responsável pelo sucesso do negócio da organização, o chefe de operações, o qual define a maneira como as operações na organização são executadas ou, pelo diretor de informações responsável pela infra-estrutura de sistemas de informação, etc.
- *Proprietário do Processo*. O proprietário do processo é responsável pela execução eficiente do processo. Também é responsável pelo planejamento, organização do processo, assim como pela definição de métricas para medir a eficiência do processo. Ele pode ainda decidir por projetos de melhoria do processo e guiar os participantes do processo na correção de erros e solução de exceções no processo. O proprietário do processo participa de todas as etapas do ciclo de vida de BPM.
- *Participante do Processo*. São participantes humanos que executam atividades do processo, seguindo as regras da organização. Participantes do processo são coordenados pelo proprietário do processo. Participam da etapa de descoberta de processo como especialistas de domínio. Finalmente, auxiliam no redesenho de atividades e na implementação do processo.
- *Analista de Processo*. O analista de processo realiza a identificação, descoberta, análise e redesenho de atividades. Além disso, coordena a implementação do processo, assim como o controle e monitoramento.
- *Engenheiro de Sistema*. Os engenheiros de sistema participam das fases de redesenho e implementação de processos. Eles interagem com os analistas de processo para entender os requisitos do processo. Além disso, os engenheiros de sistema traduzem os requisitos e processo em um sistema, o qual eles devem implementar, testar e executar.
- *Grupo de BPM*. Organizações que se dedicaram a BPM por anos, normalmente têm uma vasta documentação sobre seus processos. O grupo de BPM é responsável por garantir que esta documentação está sendo usada para atingir os objetivos organizacionais. Em particular, o grupo de BPM é responsável por manter a arquitetura de processos e pela priorização de processos que devem passar pela fase de redesenho. O grupo de BPM é ainda responsável por manter a cultura de BPM na organização.

1.1.2. Aplicabilidade de BPM

BPM vem sendo aplicado em diversos domínios de aplicação, incluindo educação, meio ambiente, saúde e entretenimento. Na educação, novos métodos vêm sendo propostos para o ensino de BPM [Sánchez-Ferreres et al. 2020], tal como métodos de ensino mais

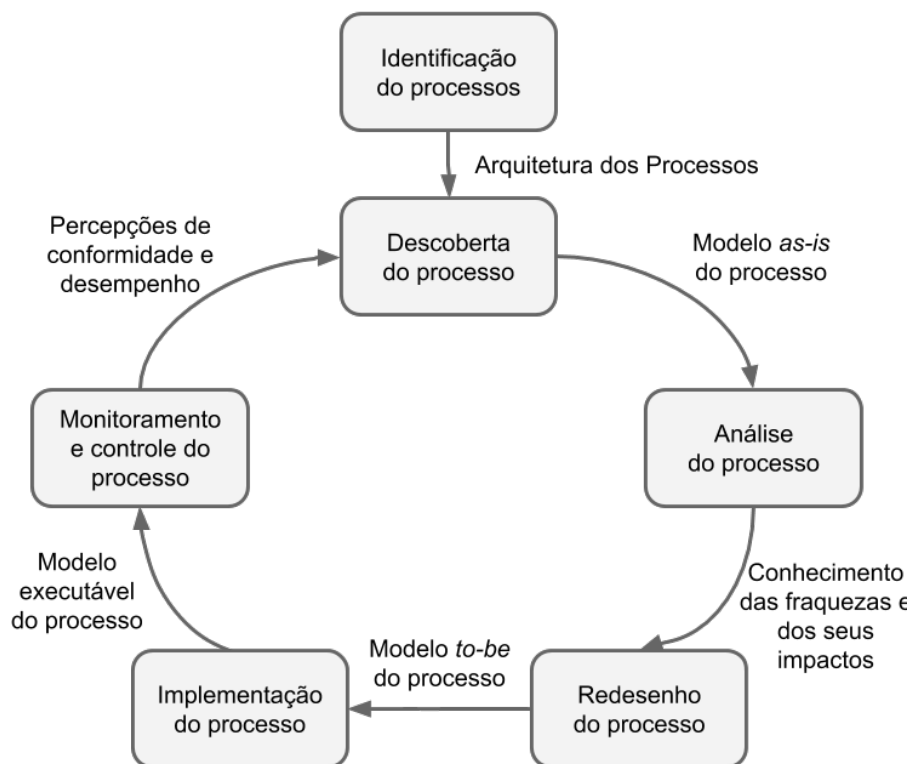


Figura 1.3. Ciclo de vida de BPM, adaptado de [Dumas et al. 2018]

dinâmicos [Enríquez et al. 2019]. Em [Drăgan et al. 2014], os autores propõem um modelo de gerenciamento de processos de negócio, descrevendo a importância de entender os requisitos dos graduados, suas necessidades e expectativas. Em [Vuksic and Bach 2012], os autores motivam a introdução de BPM como parte do currículo na educação ao nível universitário. Em [Enríquez et al. 2019], os autores realizaram uma análise empírica para avaliar as vantagens da aplicação do BPM na implementação de atividades de ensino inovadoras e dinâmicas. Com base neste estudo, desenvolveram uma ferramenta chamada RubricaSoft, um sistema de BPM focado em fornecer processos educacionais dinâmicos. A ferramenta automatiza várias tarefas, incluindo avaliação por pares, integração de informações e gerenciamento de prazos. Conforme os autores, os resultados do uso da ferramenta são promissores em termos de satisfação dos alunos, melhoria dos resultados acadêmicos e aumento da produtividade dos professores.

Em [vom Brocke et al. 2012], é discutido como os processos de negócio podem ser gerenciados e aprimorados objetivando a sustentabilidade. Os autores enfatizam que as organizações podem contribuir para as emissões de gases de efeito estufa e da produção de resíduos no contexto de seus processos de negócio. Conforme os autores, *Green BPM* compreende a documentação, modelagem, análise, simulação, execução e melhoria contínua de processos de negócio, considerando às consequências ambientais desses processos. Os autores apresentam respostas para questionamentos em diferentes níveis. No nível estratégico, eles sugerem a elaboração de estratégias de processo que reflitam adequadamente os objetivos ambientais, determinando quais são os principais indicadores

estratégicos para definir uma empresa sustentável de sucesso e como a sustentabilidade se relaciona com outros objetivos estratégicos de uma organização. Já ao nível de Governança, os autores sugerem a identificação de papéis necessários para implementar processos ambientalmente sustentáveis em uma organização. Eles também propõem identificar, quais sistemas de incentivo podem promover ou cumprir metas ambientais nos processos de negócios.

BPM também vem sendo aplicado na saúde. Conforme [Lenz et al. 2012] [Reichert and Pryss 2017], processos de assistência médica não são totalmente previsíveis nem repetitivos e, portanto, eles não podem ser totalmente pré-definidos na fase de modelagem de processos. A imprevisibilidade parcial desses processos, exige flexibilidade. Em [Thom et al. 2010] os autores apresentam um estudo detalhado sobre práticas clínicas e as exceções que podem ocorrer nos processos. Em particular, os autores buscam identificar a origem de uma exceção em um processo, classificando em exceções ocorridas devido a erros médicos (ex.: informações inconsistentes em uma carta médica), exceções relacionadas ao paciente (ex.: paciente se recusa a continuar um tratamento), exceções relacionadas a regras organizacionais (ex.: alteração no horário padrão para realização de um exame) e exceções relacionadas a contingências técnicas (ex.: um equipamento que falha durante a realização de um exame).

1.2. Modelagem de Processos de Negócio

A descoberta de processo é uma das fases mais importantes do ciclo de vida de BPM. Esta fase visa facilitar a concepção do processo e auxiliar no conhecimento sobre o processo entre os envolvidos em sua execução [Priego-Roche et al. 2012]. Conforme Dumas et al. [Dumas et al. 2018], a modelagem de processos é pré-requisito para a análise, redesenho e automação de processos de negócio. Uma modelagem de processos incorreta compromete as próximas fases do ciclo de vida de BPM, ou seja, quanto maior a precisão na modelagem de processos maior é a correção da automação [Ferreira et al. 2017]. A presente seção discute modelagem de processos. Em particular, apresentam-se as técnicas para descoberta de processos, os elementos básicos e avançados da BPMN 2.0, assim como exemplos ilustrativos de modelos de processo.

1.2.1. Técnicas para Descoberta de Processos

A *identificação de processos*, fase que precede a *descoberta de processos* é realizada em uma organização com o objetivo de identificar processos que criam valor de relevância estratégica para a organização ou processos que apresentam problemas substanciais em sua execução [Dumas et al. 2018]. É importante observar que a identificação de processos é uma tarefa contínua. Ou seja, após a resolução dos problemas de um processo, possíveis novos processos com problemas devem ser identificados. O mesmo se aplica a identificação de processos estratégicos em uma organização.

Em particular, uma organização que se dedica a uma iniciativa de BPM, precisa ter um mapa dos seus processos, assim como dos processos que requerem maior foco. Para tanto, precisa responder a duas importantes questões [Dumas et al. 2018]: i) “quais processos são executados na organização?” e (ii) ” em quais processos a organização deve se concentrar?” Após responder estas questões, a organização pode iniciar a fase de des-

coberta de processo, isto é a modelagem dos seus processos em uma notação, tal como a BPMN 2.0.

Conforme [Dumas et al. 2018] e [Leopold and Mendling 2012] um dos principais problemas na fase de *descoberta de processos* é a dificuldade para identificar e compreender os processos-chave executados na organização. Essa dificuldade se deve, principalmente, à complexidade de termos de domínio, às ambiguidades na descrição verbal dos processos por parte de seus participantes [Gassen et al. 2014] e aos variantes que um mesmo processo pode apresentar quando executado em contextos diferentes (ex.: processos de importação de mercadorias que, por questões legais do país onde são executados, apresentam diferenças em sua estrutura e modelagem). Consequentemente, os modelos de processo podem ficar incompletos e com partes definidas a partir da observação empírica do analista.

Neste contexto, a literatura de BPM apresenta três classes principais de métodos para descoberta de processos [Dumas et al. 2018]: descoberta baseada em evidência, descoberta baseada em entrevista e descoberta baseada em *workshop*.

Descoberta de Processos Baseada em Evidência: A classe descoberta baseada em evidência inclui três métodos [Dumas et al. 2018]: análise de documentos, observação e descoberta automática de processos.

- *Análise de Documentos:* a técnica de *Análise de Documentos* considera que geralmente existe na organização ou na literatura documentação relacionada ao processo que se deseja modelar. Tal documentação pode incluir descrições de processo de iniciativas de modelagem que não foram continuadas. Outros documentos incluem regras organizacionais, estruturas organizacionais, glossários, manuais, formulários, etc. Neste contexto, [Caporale 2016] apresenta uma abordagem para gerar modelos de processo a partir de sentenças em linguagem natural. As sentenças são descritas como requisitos de software. Esta abordagem considera que o texto é procedural e descrito como uma sequência de passos, fato que nem sempre ocorre em textos em linguagem natural. Já em [Friedrich et al. 2011] é proposta a extração de modelos de processo de negócio a partir de texto em linguagem natural. A principal limitação da abordagem se refere ao fato de que os textos analisados para a geração de modelos de processo devem estar gramaticalmente corretos conforme a Língua Inglesa. Ou seja, o texto é inicialmente corrigido manualmente e, após, analisado pela ferramenta. Em [Leopold 2013] é apresentada uma abordagem para extração de texto em linguagem natural a partir de modelos de processo. Entretanto, as frases geradas pela ferramenta são comparativamente curtas e simples, abstraindo a complexidade de determinados modelos de processo. Mais recentemente, em [Silva et al. 2019], [Silva et al. 2019] é proposta uma arquitetura orientada a serviços, com o objetivo de extrair modelos de processo a partir de texto em linguagem natural. O importante desta abordagem é possibilitar a marcação de determinados elementos notacionais da BPMN 2.0 no texto, assim como a análise de sentenças, visando a eliminação de ambiguidades. Tal abordagem, ainda que promissora, requer a disponibilização em uma ferramenta portátil para usuários. Outras iniciativas, referentes ao uso de Processamento de Linguagem Natural (*Natural Language Processing* - NLP) em BPM, podem ser encontradas em [Mendling et al. 2019].

- *Observação*: A técnica de *Observação* é interessante, pois ela revela como o processo é executado na prática. Este fato pode ser considerado uma vantagem em relação à técnica de análise de documentos que geralmente considera documentos mais antigos, que podem estar desatualizados. Na técnica de observação, um analista de processos irá se passar por um participante do processo em estudo ou irá observar a execução deste processo por um participante. É importante ressaltar que a técnica de observação vai resultar na análise das partes do processo em observação e não na visão geral do processo. Também se deve considerar que as pessoas podem executar suas tarefas de maneira diferente (ex.: de maneira mais rápida) por saberem que estão sendo observadas.
- *Descoberta Automática de Processos*: A técnica de *Descoberta Automática de Processos* usa *logs* de eventos, ou seja, dados de execução de processos armazenados por sistemas corporativos existentes em uma organização, para descobrir automaticamente um modelo de processo suportado por esses sistemas. Uma das limitações da técnica de descoberta automática de processos é que nem sempre os *logs* de eventos estão disponíveis ou são completos o suficiente para a descoberta do processo. Em [Kalsing et al. 2013], os autores apresentam um método para semi-automatizar a descoberta de processos de negócios implementados implicitamente no código-fonte dos sistemas legados. Em particular, os autores propõem uma ferramenta para executar a instrumentação do código fonte, identificando as regras de negócios implementadas no sistema legado e permitindo a criação de *logs* de eventos. Isso permite o uso de técnicas de mineração de *logs* para descobrir a ordem parcial de execução das regras de negócios. Já em [Kalsing et al. 2013] os autores propõem técnicas para mineração incremental de *logs* de execução. Tais técnicas possibilitam a descoberta de alterações nas instâncias do processo, mantendo o modelo de processo descoberto sincronizado com o processo que está sendo executado.

Descoberta de Processos Baseada em Entrevista

O objetivo da descoberta de processos baseada em *entrevista* é entrevistar especialistas de domínio [Dumas et al. 2018]. Como o processo pode envolver diferentes perspectivas, diferentes especialistas de domínio devem ser entrevistados. Conforme [Dumas et al. 2018], existem duas estratégias que podem ser seguidas para a realização de uma entrevista: i) iniciar a entrevista a partir das saídas do processo e, então identificar todos os passos até o início do processo ou; ii) seguir a abordagem inversa, analisando o início do processo e os passos seguintes até a sua conclusão.

Para realizar a entrevista de maneira mais estruturada é importante ter uma lista de perguntas pre-definidas para guiar a entrevista. Uma parte do tempo de entrevista pode ser usada para responder às perguntas e outra parte para discutir aspectos gerais e relevantes ao processo. Uma das limitações da descoberta baseada em entrevista é que exceções no processo podem ser omitidas por parte do entrevistado. Em vista disso, é importante que o analista faça questionamentos também em relação as possíveis exceções no processo [Thom et al. 2010].

Após coletar as informações iniciais sobre o processo, um modelo de processo preliminar pode ser modelado. Este modelo deve ser validado com o usuário e completado

com novas informações. O modelo pode ser considerado completo, quando o entrevistado aprova a versão apresentada pelo analista.

Descoberta de Processos Baseada em *Workshop*

A técnica de descoberta de processos baseada em *workshop* possibilita obter o entendimento detalhado do processo, além de resolver inconsistências de domínio. Isso é possível pois mais participantes do processo interagem durante um *workshop*. Um *workshop* pode envolver dois papéis adicionais, além dos participantes do processo [Dumas et al. 2018], que são um facilitador para coordenar o *workshop* e um analista de processos para modelar uma primeira versão do processo.

Em termos de organização, o *workshop* precisa ser comunicado aos participantes com semanas de antecedência. Isso é importante para garantir que participantes-chave do processo tenham disponibilidade para participar. Conforme [Dumas et al. 2018], quando da realização do *workshop*, os organizadores devem, inicialmente, expor os objetivos do *workshop*. Eles podem apresentar a importância do projeto de BPM para as metas da organização e o escopo em que as modelagens serão realizadas. Após, deve-se aplicar alguma técnica para iniciar a obtenção de informações sobre o processo. Neste contexto, deve-se restringir o tempo de fala de participantes mais extrovertidos e motivar a fala de participantes mais introvertidos. Em outro momento do *workshop*, poderá apresentar a BPMN 2.0 e o modelo inicial obtido através do *workshop*.

1.2.2. Elementos Notacionais Básicos da BPMN 2.0

O principal objetivo da BPMN 2.0 é prover uma notação de fácil entendimento por todos os usuários, incluindo os analistas que criam os rascunhos iniciais dos processos, os desenvolvedores técnicos que são responsáveis por implementar a tecnologia que irá executar estes processos e as pessoas que irão administrar e monitorar os processos. [Object Management Group (OMG) 2011]. Existem cinco categorias básicas de elementos na BPMN 2.0 (ver Figura 1.4: Objetos de Fluxo, Dados, Objetos de Conexão, Partições e Artefatos).

Objetos de fluxo são os principais elementos da BPMN 2.0. Eles definem o comportamento do processo, através de três tipos básicos: *eventos* representam algo que acontece instantaneamente em um processo, afetando o seu fluxo, tal como a necessidade de um solicitante de um auxílio financeiro por seu trabalho ter sido aceito por um evento; *atividades* são os elementos que representam o trabalho executado durante o processo, podendo ser atômicas (denominadas *tarefas*) ou compostas (denominadas *subprocessos*); *gateways* controlam a divergência e convergência do fluxo do processo, podendo definir que a execução de uma sequência de atividades seja feita paralelamente ou somente quando uma condição é atendida.

Os *Objetos de Conexão* fazem a ligação entre os objetos de fluxo, dados e artefatos. Existem três maneiras de conectar os elementos: através de um *fluxo de sequência*, que conectam os objetos de fluxo, definindo a ordem na qual estes são executados; de um *fluxo de mensagem*, que representa a troca de mensagens entre duas organizações; e de uma *associação*, que relaciona um objeto de fluxo a um objeto de dados.

Os objetos de fluxo e de conexão são fundamentais para a criação de modelos de

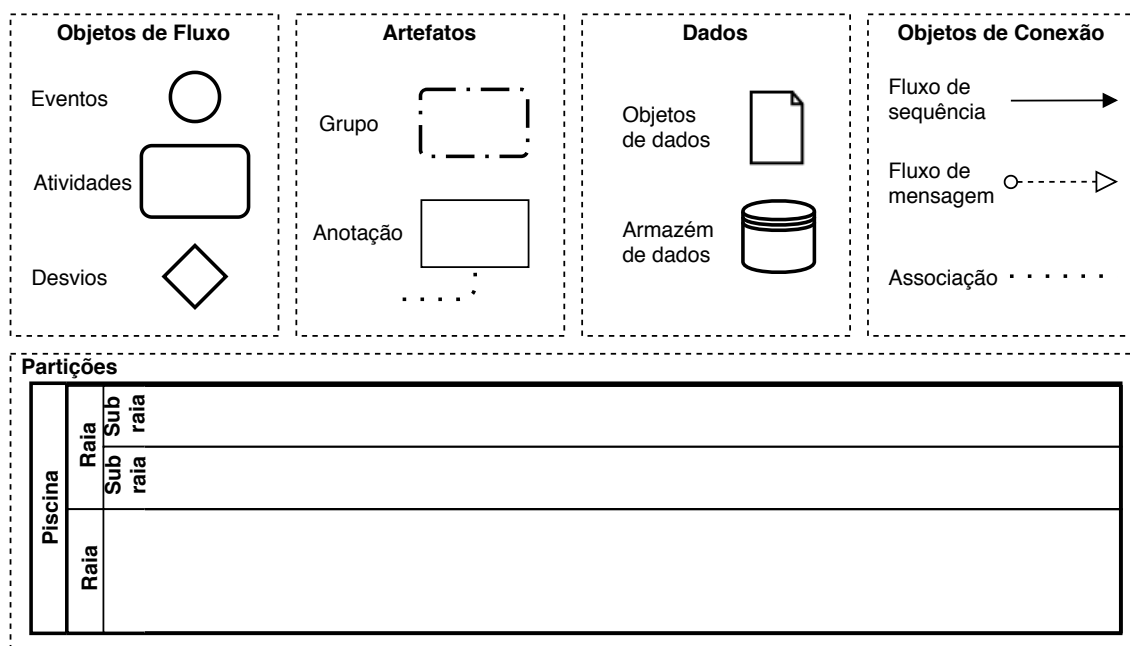


Figura 1.4. Elementos da BPMN 2.0, adaptado de [Object Management Group (OMG) 2011].

processo, pois eles definem o que o processo faz e como. Por exemplo, utilizando somente atividades, eventos e fluxos de sequência, podemos criar o modelo de processo apresentado na Figura 1.5. Este modelo contém uma sequência de atividades para o atendimento e entrega de um pedido. Uma sequência é a maneira mais elementar de relacionar objetos de fluxo na BPMN, em que a execução de um objeto A precede a execução de um objeto B. No caso da Figura 1.5, esta sequência descreve que a atividade "Confirmar quantidade de hóspedes" é executada após o evento "Reserva de quarto pedida" acontecer, e esta atividade será seguida pela execução da atividade "Definir tipo de quarto" e pelas outras atividades da sequência até o fim do processo em que acontece o evento "Reserva marcada."

Através dos *gateways*, é possível criar modelos de processo mais complexos. O comportamento de um *gateway* depende do seu tipo. Para modelar um processo onde se escolhe a execução de um fluxo de processo dentre duas ou mais alternativas, utiliza-se um *gateway de decisão exclusiva*, também chamado de *gateway XOR*. O uso deste *gateway* pode ser observado no modelo apresentado na Figura 1.6, em que existe uma decisão entre aprovar ou não um empréstimo. Existem dois *gateways* neste modelo, sendo o primeiro responsável por divergir o fluxo do processo entre às duas opções (*XOR-Split*). A escolha dentre estas opções é definida pela condição escrita nos fluxos de sequência que partem deste *gateway*. É importante que estas condições sejam mutuamente exclusivas, ou seja, que somente uma delas seja verdadeira quando o *gateway* é executado, pois esta é a semântica do *gateway XOR*. O segundo *gateway* do modelo tem a função de juntar os dois fluxos, após eles terem divergido (*XOR-Join*). Ele espera a execução e o término de somente um dos fluxos precedentes a ele para então continuar a execução do processo com a atividade "Informar beneficiário."

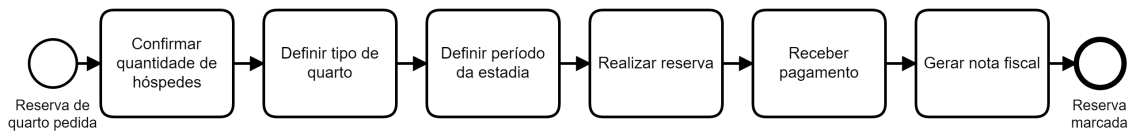


Figura 1.5. Exemplo de modelo de processo de reserva de quartos de hotel.

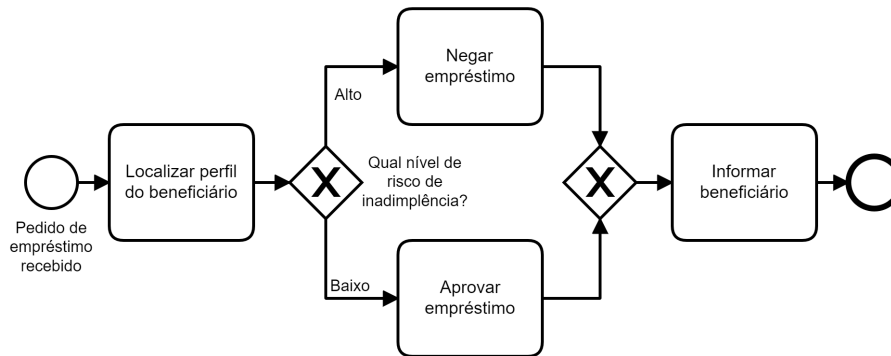


Figura 1.6. Exemplo de uso de *gateways XOR* em modelo de processo de aprovação de empréstimos.

Gateways XOR também permitem a criação de *loops* dentro de um processo, ou seja, da repetição de um fragmento de processo até que uma determinada condição seja atendida. Por exemplo, na Figura 1.7, um paciente em um hospital permanecerá hospitalizado até que os seus médicos aprovem a sua saída. Neste caso, o *gateway* que diverge o fluxo do processo está ao final do fragmento de processo em repetição, enquanto o *gateway* no início do fragmento junta o fluxo de entrada com o fluxo que determina quando acontece a repetição.

Em contraste aos *gateways XOR*, existem os *gateways paralelos*, ou *gateways AND*, que executam os fluxos do processo conectados a eles de forma paralela. Podem existir dois ou mais fluxos a serem executados e cada uma destas execuções podem acontecer independente das outras, sem ordem predefinida. A figura 1.8 apresenta um modelo de processo utilizando a execução em paralelo de duas atividades de triagem, uma de segurança e outra de bagagem. Ambas atividades podem acontecer, em qualquer ordem, após a atividade "Proceder inspeção de segurança" e somente após a execução de ambas é permitido a execução da atividade "Proceder ao embarque". Os *gateways* paralelos controlam a execução paralela (*AND-Split* dos dois fluxos deste modelo e a sincronização (*AND-Join*) dos mesmos após eles terminarem.

Existem casos durante a modelagem de um processo em que se deseja que o fluxo do processo divirja entre uma ou mais opções após um *gateway* de decisão. Esta é a função do *gateway de decisão inclusivo*, ou *gateway OR*, que combina as funcionalidades do *gateway XOR*, possibilitando a execução condicional de fluxos de processo, com a funcionalidade do *gateway AND*, que permite a execução destes fluxos em paralelo. Um exemplo destas funcionalidades é apresentado na Figura 1.9, onde durante um diagnóstico médico é possível recomendar um tratamento, prescrever um medicamento, ou ambos (*OR-Split*). Quais destas atividades serão executadas depende das condições apresentadas

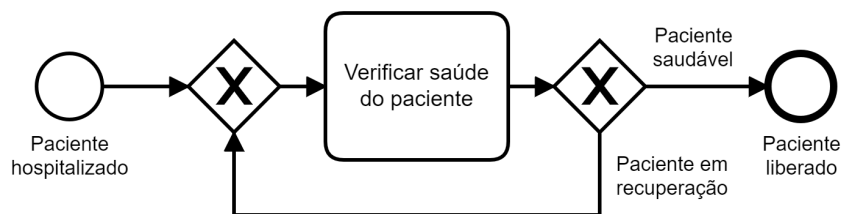


Figura 1.7. Exemplo de uso de *gateways XOR* para a criação de loops.

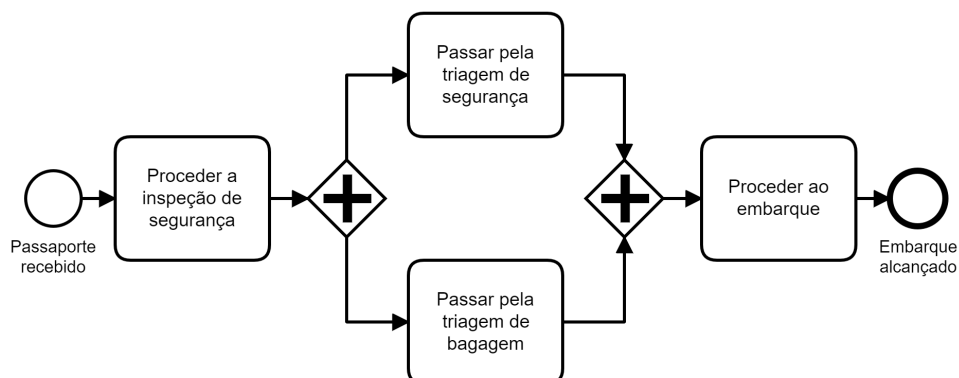


Figura 1.8. Exemplo de uso de *gateways AND* em modelo de processo de checagem de segurança ao embarque, adaptado de [Dumas et al. 2018].

nos fluxos de sequência, assim como nos *gateways XOR*, mas neste caso a execução de um dos fluxos não impede a execução dos outros. O *gateway OR* sincroniza os fluxos que chegam nele (*OR-Join*), esperando o término de todos os fluxos que estão em atividade para depois continuar o processo.

Nos exemplos apresentados, os fluxos de processo divergidos por um tipo de *gateway* são juntados por gateways do mesmo tipo. Entretanto, durante a modelagem de processos pode-se usar *gateways* de tipos diferentes. É importante que a combinação de *gateways* de tipos diferentes seja feita com cuidado, pois o uso inapropriado destes pode causar problemas durante a execução do processo. Por exemplo, o uso de um *gateway*

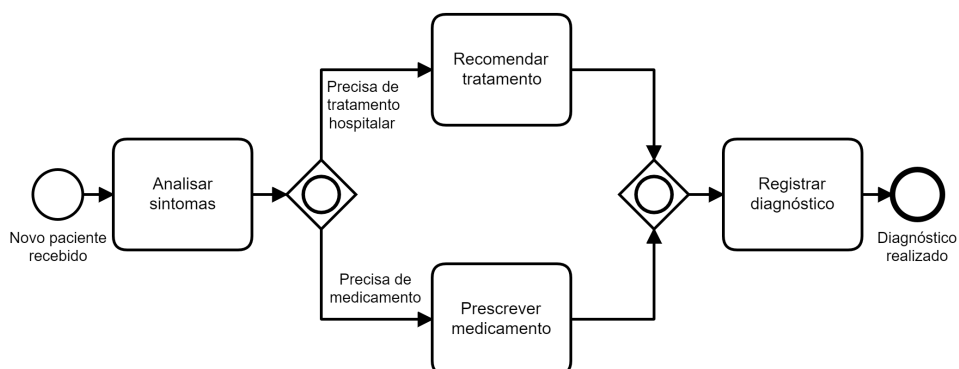


Figura 1.9. Exemplo de uso de *gateways OR* em modelo de processo de um diagnóstico médico.

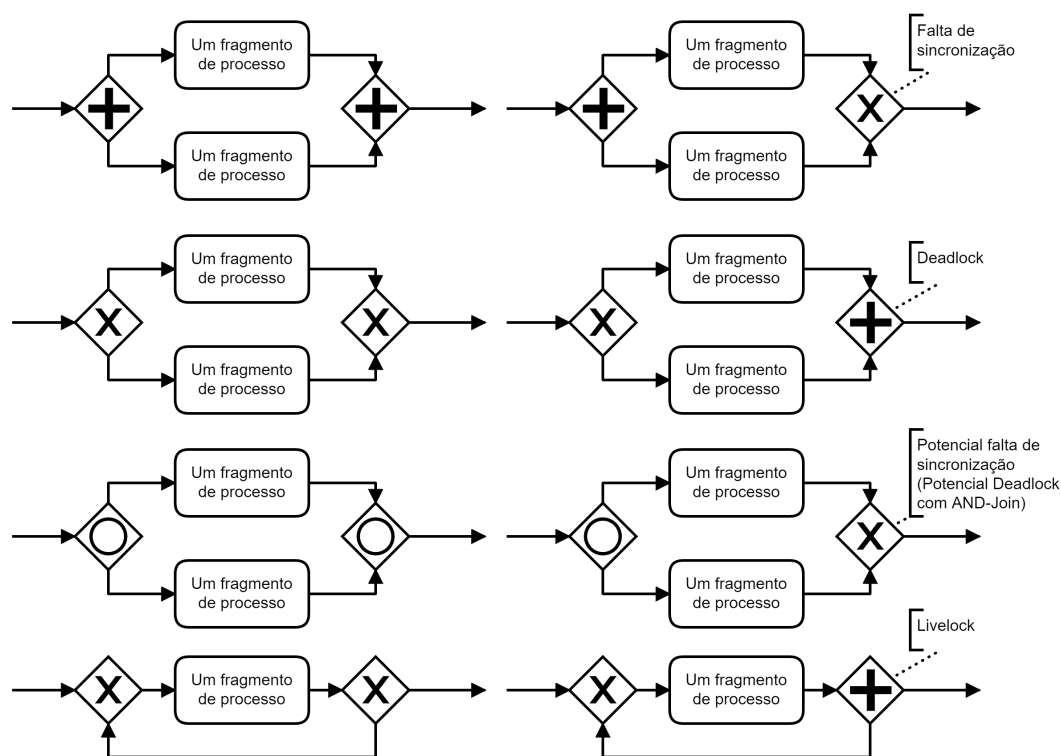


Figura 1.10. Exemplos de fragmentos de processo *sound* e *unsound* adaptado de [Dumas et al. 2018].

AND para sincronizar fluxos divergidos por um *gateway* XOR causa uma anomalia no comportamento do processo denominada *deadlock*, que é quando um processo não consegue progredir devido a um *gateway* AND que espera indefinidamente o término de um dos fluxos que ele tenta sincronizar. Modelos de processo que contém estas anomalias de comportamento são chamados de modelos sem a propriedade de *soundness*, ou modelos *unsound* [Dumas et al. 2018]. Além de *deadlock*, a falta de sincronismo e o *livelock* são outras anomalias que define modelos *unsound*. A falta de sincronismo acontece quando um *gateway* AND diverge fluxos que são unidos por um *gateway* XOR, causando a duplicação do fluxo seguinte. Um *livelock* ocorre quando o processo contém um *loop* sem término. A Figura 1.10 apresenta exemplos de fragmentos de modelos *sound* e *unsound*.

Com os objetos de fluxo e de conexão, os modelos apresentados nos exemplos anteriores mostram a perspectiva funcional, que atividades acontecem durante o processo, e a perspectiva de controle defluxo, quando as atividades devem ocorrer. Os *Objetos de Dados* são elementos da BPMN 2.0 que mostram a perspectiva de dados do processo, ou seja, quais documentos ou arquivos são necessários para executar uma atividade e quais são produzidos como resultado de uma atividade. Por exemplo, a Figura 1.11 apresenta uma versão mais detalhada do modelo de processo de reserva de quartos previamente visto na Figura 1.5. Neste modelo, as primeiras atividades executadas produzem o objeto de dados "Requisitos de quarto", que será utilizado pela atividade "Verificar disponibilidade de quartos" para buscar um quarto que atende os requisitos. Objetos de dados são conectados a atividades utilizando uma *associação*. A direção desta associação determina se o objeto de dado serve como entrada ou saída de uma atividade. Este modelo também

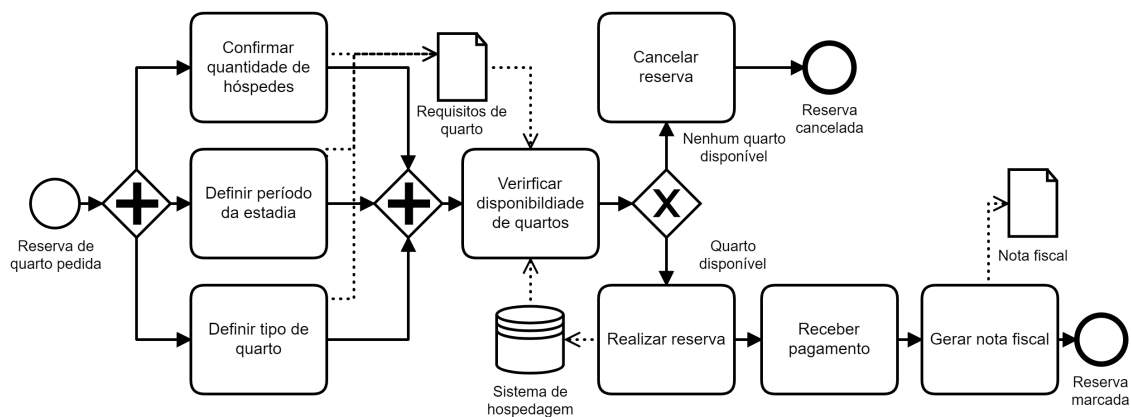


Figura 1.11. Uma versão mais detalhada do modelo de processo para reserva de quartos de hotel com objetos de dados.

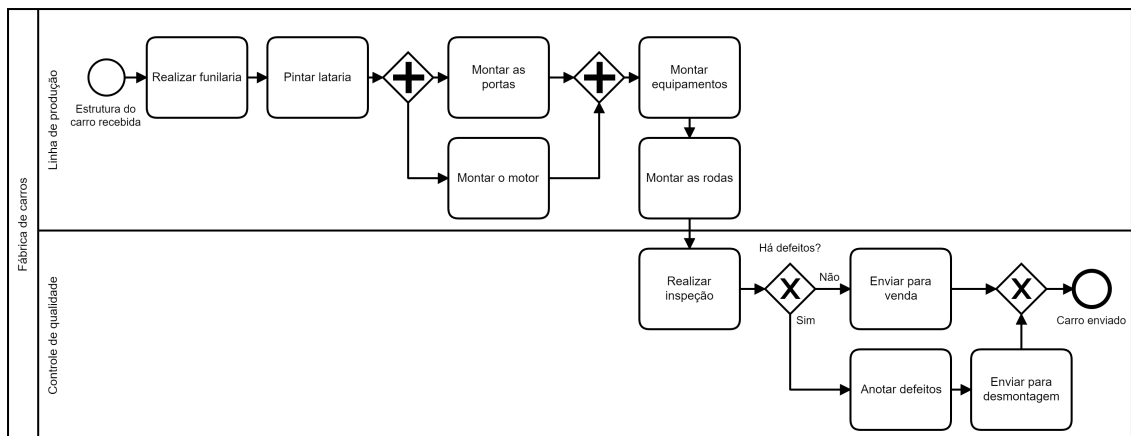


Figura 1.12. Exemplo de uso de piscina com raias em modelo de processo de manufatura de carros.

apresenta um repositório de dados, o "Sistema de hospedagem". Repositórios de dados são locais que contêm objetos de dados que continuam existindo após o término de uma execução de um processo, tais como bancos de dados, para objetos de dados digitais, ou repositórios de arquivos, para objetos de dados físicos.

As *Partições* são elementos da BPMN 2.0 que mostram a perspectiva de recursos do processo, ou seja, quem ou o que executa as atividades do processo. Partições podem ser *piscinas* e *raias*. As piscinas representam as entidades de um processo, tal como uma organização. Cada piscina contém as atividades realizadas dentro desta organização. Piscinas também podem ser subdivididas em *raias*, representando os diferentes recursos presentes dentro da organização. Raias também podem ser contidas em outras raias, tendo assim múltiplos níveis de aninhamento. O uso de piscinas e raias pode ser visto no modelo da Figura 1.12 que apresenta um processo de manufatura de carros. A organização que realiza este processo, a "Fábrica de carros" é representado pela piscina que contém todos os elementos dentro de si. As raias "Linha de produção" e "Controle de qualidade" dividem esta piscina. É importante que as atividades, eventos e *gateways* de decisão sejam posicionados dentro das piscinas e raias que são responsáveis pelas suas

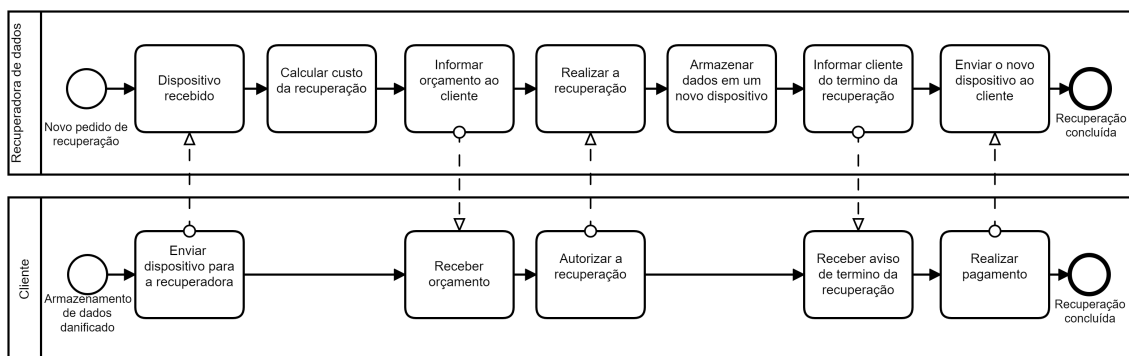


Figura 1.13. Exemplo de uso de piscinas com troca de mensagens em modelo de processo de recuperação de dados.

execuções.

Pode-se utilizar mais de uma piscina para modelar duas ou mais organizações participando de um mesmo processo. Neste caso, cada piscina contém o modelo de processo respectivo àquela organização. O uso de múltiplas piscinas permite mostrar como é feita a interação entre as diferentes organizações. Entretanto, não é possível utilizar o fluxo de sequência, pois não é permitido a este cruzar as fronteiras de uma piscina. Ao invés disto, deve-se utilizar o fluxo de mensagem, que representa a troca de informação entre duas organizações. Na Figura, 1.13 mostra-se exemplos de troca de mensagens entre um cliente e uma recuperadora de dados, tal como entre as atividades "Informar orçamento ao cliente" e "Receber orçamento." Um fluxo de mensagem pode ser de qualquer tipo de comunicação, como um e-mail, fax, ligação telefônica ou entrega manual de cartas e pacotes.

Piscinas, tais como apresentadas na Figura 1.13 mostram como funciona o processo internamente naquela organização. É possível modelar piscinas onde este processo interno não é exibido, como uma *caixa preta*. Para isto, utiliza-se uma "pool colapsada", que apresenta somente o retângulo vazio da piscina com o nome da organização e os fluxos de mensagens enviadas e recebidas conectados às suas bordas.

Finalmente, a BPMN 2.0 apresenta os *Artefatos*. Artefatos agregam informações adicionais aos modelos, como *Anotações* ou *Grupos* de elementos. Anotações podem ser conectadas a outros elementos usando uma associação. Artefatos não alteram a execução do processo.

1.2.3. Elementos Notacionais Avançados da BPMN 2.0

Elementos notacionais avançados da BPMN 2.0 incluem elementos de modelagem que possibilitam maior poder de expressão para a modelagem. Por exemplo, os eventos utilizados nos exemplos apresentados eram eventos de início e de fim, pois eles indicavam onde cada processo começava e terminava. Eventos também podem ocorrer no meio do processo, quando algo deve acontecer, antes que o processo continue. Estes eventos são do tipo intermediário e eles param a progressão do processo onde eles foram modelados até que o evento ocorra.

Eventos de início, fim e intermediários também podem ser marcados com um



Figura 1.14. Exemplo de tipos de eventos da BPMN 2.0.

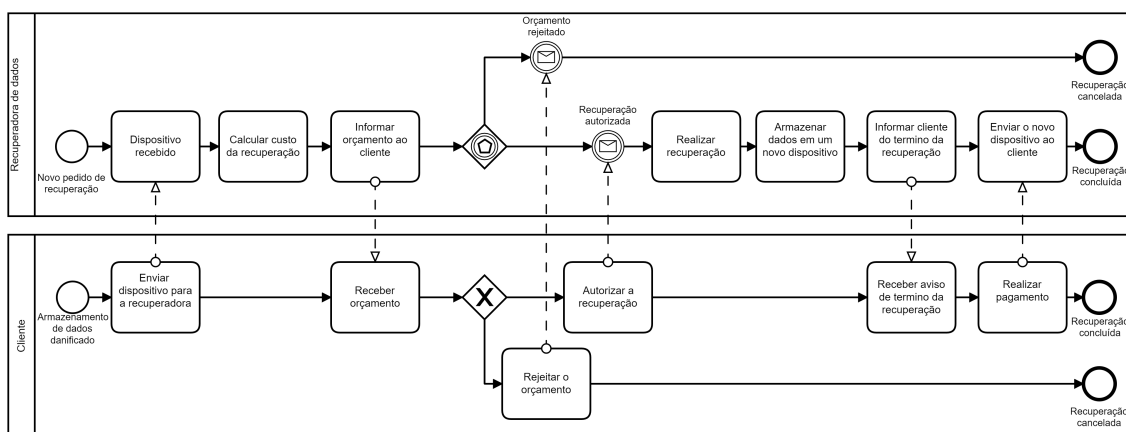


Figura 1.15. Exemplo de uso de gateway de desvio condicional baseado em eventos no modelo de processo de recuperação de dados.

símbolo representando a semântica do evento (ver Figura 1.14). Por exemplo, uma marca de um envelope em um evento inicial significa que o processo inicia com a chegada de uma mensagem. Um relógio em um evento intermediário pode representar que um certo fluxo de processo só terá continuidade após um determinado intervalo de tempo ou após a chegada de um dia ou horário específico.

Existem situações durante um processo em que o fluxo do processo é alterado dependendo de qual evento acontece primeiro de um conjunto de eventos. Por exemplo, no exemplo anterior da Figura 1.13, após informar um orçamento ao cliente, o cliente poderia enviar uma mensagem aceitando o orçamento ou outra mensagem rejeitando o mesmo. Para isto, utiliza-se um *gateway* de desvio condicional baseado em evento. Este *gateway* apresenta a mesma semântica do *gateway* XOR, porém ao invés de as condições serem apresentadas nos fluxos de sequência elas são contidas nos eventos que seguem cada fluxo. A Figura 1.15 mostra o processo de recuperação de dados utilizando este *gateway*.

Gateways podem ter outros tipos avançados. Além do desvio condicional por evento, existe a ativação exclusiva condicional e ativação incondicional em paralelo (ver

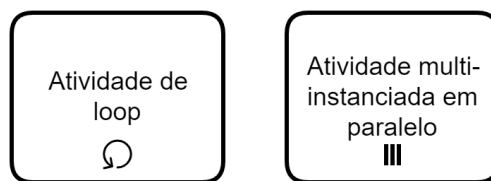


Figura 1.16. Exemplo de tipos de atividades da BPMN 2.0.

Figura 1.17). Ambos estes *gateways* podem ser usados como o início do processo e eles controlam a ativação do processo com base em qual evento ocorre primeiro. No caso da ativação exclusiva, quando um dos eventos relacionados acontece, os outros eventos são desativados. Na ativação em paralelo, os eventos permanecem ativos e a espera, após a ativação do primeiro evento relacionado. Por fim, existem os *gateways* complexos (ver Figura 1.17 "complexo"), que permitem modelar uma sincronização complexa, além das sincronizações já disponíveis pelos *gateways* XOR, AND e OR.

No caso de atividades, os elementos avançados da BPMN 2.0 contém os *subprocessos*, definido como uma atividade composta de outros elementos notacionais. Subprocessos são úteis quando um modelo de processo apresenta uma grande quantidade de elementos, o que pode dificultar o seu entendimento. Neste caso, um subprocesso pode agrupar um conjunto de atividades relacionadas em uma única atividade composta auto-contida. Um subprocesso pode ser *expandido*, quando seus detalhes internos de modelagem são visíveis ou *colapsados*, quando apenas um símbolo de "+" indica que existem mais detalhes no modelo de processo que não estão explícitos no modelo principal. A figura 1.17 apresenta exemplos de ambos os subprocessos. Subprocessos colapsados são ideais para esconder os detalhes mais complexos do processo, permitindo assim a redução do número de elementos do modelo e o aumento da sua compreensibilidade.

Atividades também pode conter marcadores avançados que expandem a semântica deste elemento (ver Figura 1.16). Por exemplo, atividade de *loop*, representada pela marca de uma seta em ciclo, define uma notação simplificada para a repetição de uma atividade (ou de um subprocesso). Desta forma, a atividade de *loop* pode substituir o uso de *gateways* XOR para a repetição de fragmentos de processo. Atividades de multi-instanciação em paralelo, representadas pela marca de 3 barras verticais, representam a repetição de uma mesma atividade executada em paralelo. Desta forma, pode-se evitar o uso de *gateways* AND que paralelizam um grande número de uma mesma atividade, o que simplifica o modelo.

1.3. Qualidade em Modelagem de Processos

A qualidade de modelos de processos é muito importante [Reijers et al. 2015], pois não é incomum que os modelos tenham problemas como erros no controle de fluxo, estruturas e layouts com designs mal projetados, e nomenclaturas incorretas [Leopold et al. 2016]. Para definir o que é a qualidade de modelos de processo, utiliza-se de abordagens que dividem a qualidade em tipos que focam em características específicas. Os tipos mais frequentemente abordados na literatura são a qualidade sintática, a qualidade semântica e a qualidade pragmática.

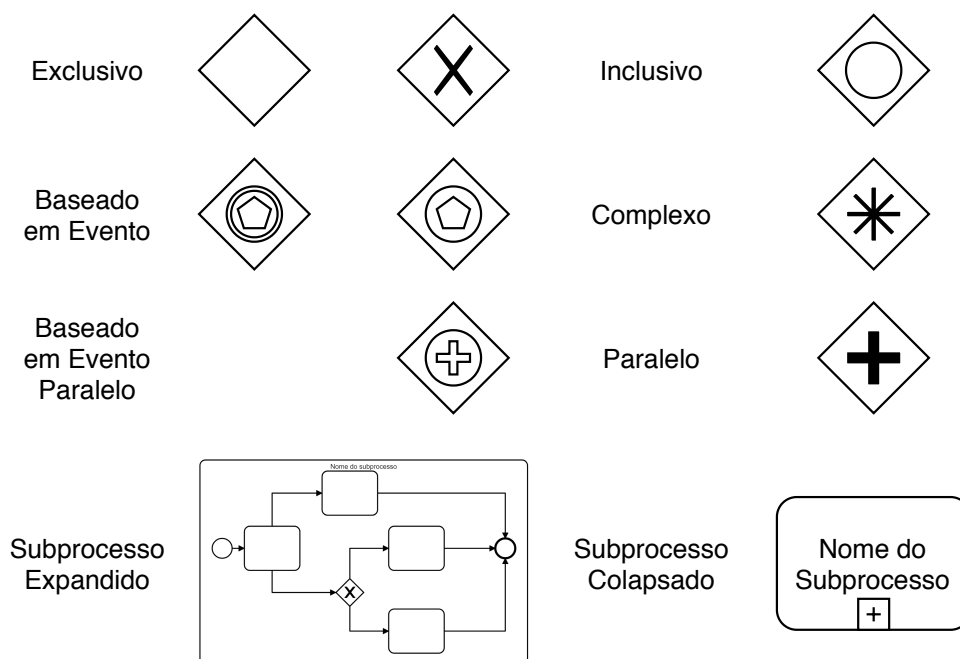


Figura 1.17. Tipos de Subprocesso e Gateways da BPMN 2.0, adaptado de [Object Management Group (OMG) 2011].

1.3.1. Qualidade sintática

A qualidade sintática está relacionada com a conformidade do modelo de processo com as regras de sintaxe da notação utilizada pelo modelo. Notações como a BPMN 2.0 possuem um conjunto diverso de regras sintáticas e instruções para criar seus modelos de processo. Para que o modelo tenha qualidade sintática, estas regras e instruções devem ser respeitadas, pois elas garantem que o modelo possa ser interpretado por outras ferramentas e outros usuários [Dumas et al. 2018].

Para verificar a conformidade de um modelo de processo, avalia-se a sua *corretude sintática*, ou seja, que todos os símbolos e construções do modelo estão de acordo com o vocabulário e a sintaxe da notação [Krogstie 2016]. Modelos que não estão sintaticamente corretos podem ter dois tipos de erros: *invalidade sintática*, em que um símbolo não presente no vocabulário da notação é utilizado; e *incompletude sintática*, na qual existe a falta de uma informação ou símbolo no modelo para que este obedeça à regras gramaticais da notação.

As duas principais maneiras de garantir a qualidade sintática de um modelo de processo são a prevenção e a detecção de erros. Ferramentas de modelagem frequentemente usam ambas estas abordagens para auxiliar seus usuários. No caso da prevenção de erros, as ferramentas de modelagem não permitem a modelagem de construções que violam as regras sintáticas da notação. Para isto, a escolha de elementos disponíveis para modelar é limitada de acordo com o vocabulário da notação e com a sintaxe esperada pelo modelo que está sendo criado. A detecção de erros, por outro lado, permite que o modelo esteja temporariamente incorreto sintaticamente para facilitar a modelagem. A corretude do modelo é então avaliada quando o usuário da ferramenta requisita está avaliação, permitindo que ele faça as correções necessárias.



Figura 1.18. Exemplo de modelo com invalidade semântica

1.3.2. Qualidade Semântica

A qualidade semântica está relacionada com o objetivo do modelo de processo de representar corretamente a execução do processo atual, no caso de modelos *as-is*, ou futura, em modelos *to-be*. As características avaliadas nesta qualidade são a validade semântica do modelo, que determina se o que é apresentado pelo modelo é relevante e que realmente acontece durante o processo, e a completude semântica do modelo, que determina que o modelo apresenta todos os elementos relevantes que fazem parte da execução do processo [Krogstie 2016].

Um exemplo de modelo de processo contendo problemas relacionados a qualidade semântica é apresentado na Figura 1.18. Nesta figura apresenta-se um processo para o atendimento médico de um paciente, porém a atividade "Beber café" não é relevante ao problema de realizar um diagnóstico, sendo portanto inválido para este processo. É possível argumentar também que este modelo está incompleto, pois a prescrição de medicamentos pode ser relevante para um atendimento médico.

A avaliação da qualidade semântica é dificultosa, pois tipicamente não é possível medir como o processo é executado e comparar esta medida ao modelo de processo. Quando este é o caso, não existem abordagens formais que permitam a fácil validação da qualidade semântica [Dumas et al. 2018]. Portanto, a forma alternativa de avaliar esta qualidade é através da comparação do modelo de processo com o conhecimento pertencente às pessoas envolvidas com a execução deste processo e com a documentação disponível para consulta. Deve-se ter em mente, entretanto, que esta avaliação não é objetiva, e portanto avalia a qualidade semântica assim como ela é percebida pelas pessoas das quais o conhecimento foi adquirido [Krogstie 2016]. É necessário que este conhecimento também seja validado para garantir que ele não seja incorreto.

Uma maneira de facilitar a avaliação da qualidade semântica pelas pessoas envolvidas é a utilização de uma descrição textual do modelo de processo junto ao próprio modelo [Silva et al. 2019]. As descrições textuais podem auxiliar a compreensão do processo por pessoas, inclusive aquelas com pouca ou nenhuma experiência com modelagem de processos [Ottensmeyer et al. 2012]. Esta descrição pode ser criada utilizando técnicas de geração automática de descrição de processos a partir dos modelos de processo [Leopold et al. 2012].

1.3.3. Qualidade Pragmática

A qualidade pragmática é definida pelo relacionamento entre um modelo de processo e de seus leitores. Ela avalia se os leitores compreendem ou não o modelo, ou seja, se o modelo é compreensível por alguém. A qualidade pragmática é um objetivo importante

de um modelo, pois nem o melhor modelo possível será útil se não for possível entendê-lo [Wesenberg 2011]. Desta forma, qualquer interpretação de um modelo de processo deve corretamente refletir o processo que foi modelado, pois assim um leitor deste modelo pode acompanhar o comportamento do processo no mundo real através da análise do modelo de processo [Krogstie 2012].

A compreensão do modelo depende primariamente de dois tipos de fatores [Dikici et al. 2018]. O primeiro tipo, fatores pessoais, contém fatores relacionados com a capacidade do leitor de ler e entender o modelo de processo, como a sua habilidade de aprendizado e sua familiaridade com modelos de processo, com suas notações e com o domínio sendo modelado. O segundo tipo, fatores do modelo de processo, contém fatores relacionados a compreensibilidade do modelo. A compreensibilidade de um modelo é definida por um conjunto de características desejáveis que fazem com que este seja inerentemente mais compreensível.

Garantir a compreensão humana de modelos de processo não é uma tarefa trivial, pois não existe um método simples e direto para garantir a qualidade pragmática de um modelo [Mendling et al. 2010]. Para melhorar a compreensão de um modelo, é necessário melhorar os fatores pessoais e/ou os fatores do modelo de processo [Dikici et al. 2018]. Entretanto, os fatores pessoais não são facilmente alterados, pois não é possível controlar quais pessoas iram ler um modelo. Desta forma, as principais abordagens para garantir a qualidade pragmáticas dos modelos de processo foca na melhoria da compreensibilidade do modelo.

Entretanto, a melhoria desta compreensibilidade depende da experiência do modelador com a modelagem de processos [Figl 2017, Nelson et al. 2012] e com o seu conhecimento sobre o domínio do processo [Dikici et al. 2018]. Isto acontece porque, durante a modelagem de um processo, o modelador precisa lidar com complexidade da notação utilizada, seus diferentes elementos e suas respectivas semânticas [Leopold et al. 2016]. Logo, não é incomum que modelos de processo tenham problemas que prejudicam sua compreensão [Wesenberg 2011], tais como erros no controle de fluxo, estruturas e *layouts* mal projetados e nomenclaturas incorretas [Mendling and Strembeck 2008, Leopold et al. 2016]. Por exemplo, as Figuras 1.19 e 1.20 mostram dois modelos de um mesmo processo, sendo que o *layout* segundo modelo é melhor desenhado, permitindo a melhor compreensão desta versão.

Para instruir como melhorar a compreensibilidade do modelo, aplicam-se as *boas práticas de modelagem*, que guiam modeladores de processo através de regras simples que devem ser seguidas para que o modelo de processo tenha maior compreensibilidade [Mendling et al. 2010]. As boas práticas também são usadas de forma retroativa, realizando a transformação de um modelo de baixa compreensibilidade para um com alta compreensibilidade e que também preserva o comportamento original do processo [Krogstie 2012]. As boas práticas restringem o uso de construções inadequadas para ajudar os modeladores a reduzir a complexidade do modelo de processo e a quantidade de erros de modelagem.

As boas práticas de modelagem tipicamente descrevem como controlar as características do modelo de processo para que este seja mais compreensível. Um conjunto de boas práticas existentes na literatura são as chamadas Sete Boas Práticas de Modelagem (*Seven Process Modeling Guidelines - 7PMG*) [Mendling et al. 2010]. Estas regras

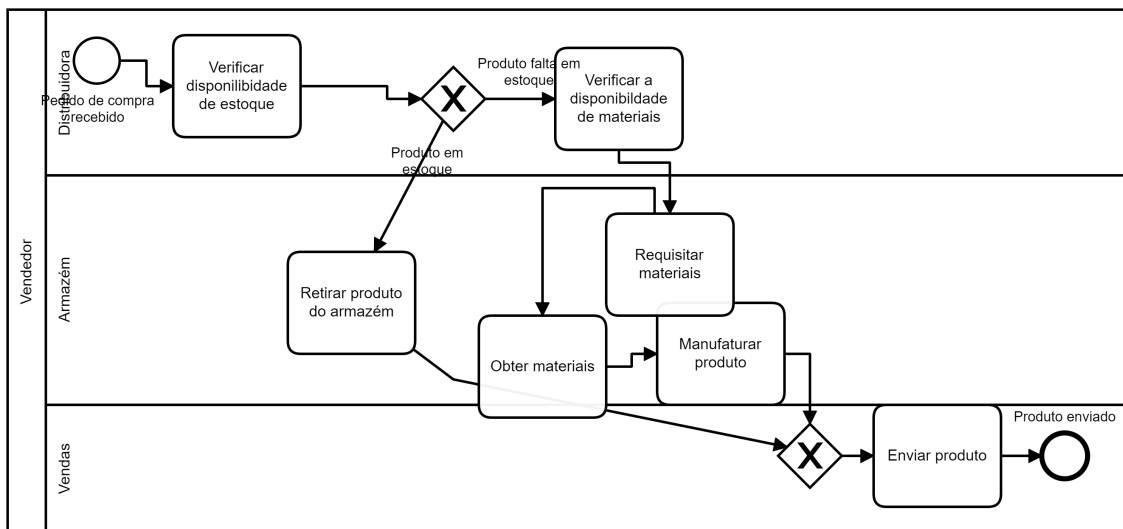


Figura 1.19. Exemplo de modelo de processo com layout ruim (adaptado de [Dumas et al. 2018]).

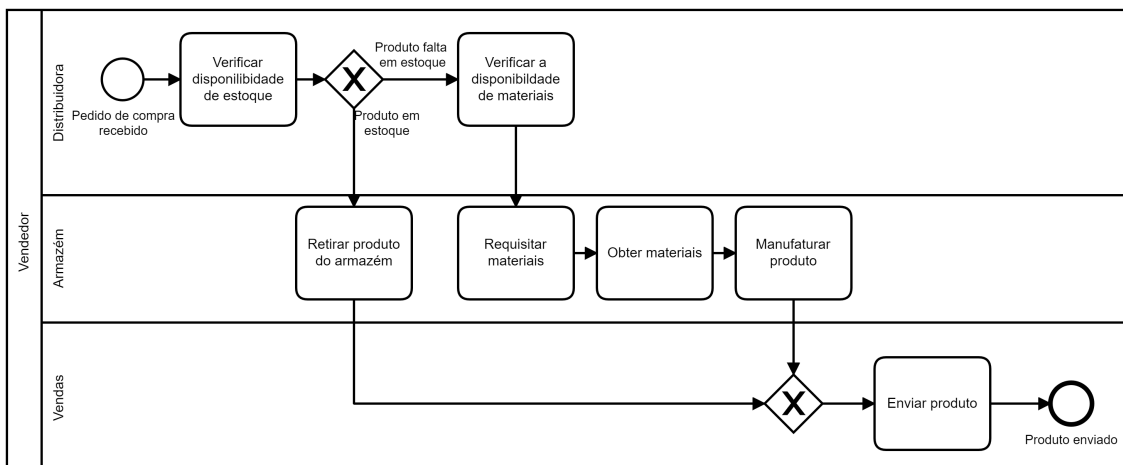


Figura 1.20. Exemplo de modelo de processo com layout bom (adaptado de [Dumas et al. 2018]).

contribuem para a redução de ambiguidade do conteúdo lógico do processo e da probabilidade de introdução de erros estruturais (ex.: *deadlock*, falta de sincronismo e *livelock*). Desta forma, o modelo é estruturalmente correto e possui alta compreensibilidade. As regras propostas neste conjunto são:

1. Usar a menor quantidade de elementos de modelagem possível.
2. Minimizar os fluxos de roteamento chegando e saindo de cada elemento.
3. Usar um evento de início e um evento de fim por processo.
4. Modelar de maneira mais estruturada possível, evitando, por exemplo, o uso de um XOR divergente, por exemplo seguido de um AND convergente, ocasionando falta de sincronismo.

5. Evitar o uso do elemento de modelagem OR.
6. Usar nomenclaturas “Verbo-Objeto” em *labels* de tarefas do processo.
7. Decompor o modelo de processo em subprocessos, caso ele tenha mais de 50 elementos de modelagem.

1.4. Automação de Modelos de Processo de Negócio em BPMS

Esta seção discute os métodos e as tecnologias existentes para a automação de modelos de processo. A execução de modelo de processo é fundamental para as fases de implementação e monitoramento e controle de processos, permitindo que os participantes destes processos realizem melhor o seu trabalho e que eles contribuam para novas métricas de performance que ajudem a melhorar o processo na próxima iteração do ciclo de vida.

Um BPMS é um sistema que suporta o projeto, a análise, a execução e o monitoramento de processo de negócio com base em seus modelos [Dumas et al. 2018]. O propósito de um BPMS é coordenar a execução de um processo automatizado para que as tarefas sejam realizadas na hora certa e pelo recurso apropriado. O uso de BPMS para a execução de processos tem vantagens, como a redução do esforço para realização do trabalho, a flexibilidade de integração de sistemas distintos, a transparência de execução e o cumprimento de regras organizacionais.

1.4.1. Arquitetura de BPMS

A figura 1.21 mostra a arquitetura de um BPMS, incluindo os cinco tipos de componentes presentes, que são o motor de execução, a ferramenta de modelagem de processo, a ferramenta de administração e monitoramento, o gerente da lista de trabalho e os serviços externos.

O *motor de execução* é o componente central de um BPMS, interagindo com os outros componentes de modo a executar os processos do sistema. Ele possui a habilidade de criar uma instância de um processo modelado, distribuir o trabalho entre os participantes do processo, resgatar e armazenar os dados necessários para a execução de um processo, comunicar-se com os serviços externos e executar as atividades do processo. Em prática, o motor de execução monitora a execução de todas as instâncias de processo ativas para coordenar quais atividades serão executadas em seguida. Ele faz isto criando itens de trabalho, que são instâncias de atividades a serem executadas das instâncias de processo ativas. O motor de execução distribui estes itens para recursos qualificados e autorizados a executá-los.

A *ferramenta de modelagem* de processos permite definir e armazenar como um processo funciona. O trabalho de automatização de um processo utiliza-se principalmente desta ferramenta, pois ela é responsável por definir o modelo do processo, os métodos de entrada e saída de dados, os participantes do processo responsáveis pela execução de cada atividade e todas as decisões e os processamentos realizados pelas atividades. O modelo de processo criado nesta ferramenta é armazenado em um *repositório de modelos de processo*, de onde ele poderá ser compartilhado e acessado. Deste repositório o modelo de processo também é acessado pelo motor de execução, o qual utiliza deste para determinar como o processo deve ser executado, gerando os itens de trabalho de acordo com a

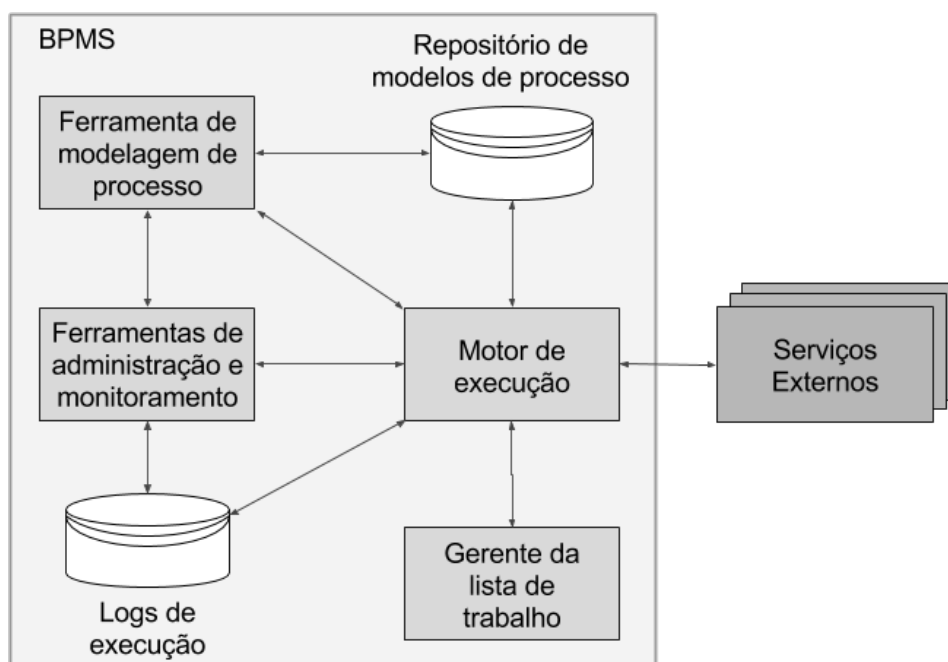


Figura 1.21. Arquitetura de um BPMS, adaptado de [Dumas et al. 2018].

ordem temporal e lógica apresentada pelo modelo, alocando estes para os recursos corretos e utilizando os serviços externos necessários. A Figura 1.22 mostra a ferramenta de modelagem *Camunda Modeler* da Camunda ¹.

O *gerente da lista de trabalho* recebe do motor de execução quais itens de trabalho estão aguardando para serem executados e os disponibiliza aos devidos usuários, normalmente através de uma “caixa de entrada”, tal como uma de um cliente de e-mails. Quando um usuário seleciona um item de trabalho desta caixa de entrada, o gerente da lista de trabalho pode apresentar um formulário eletrônico contendo os dados de entrada e saída de uma atividade. Após os usuários informarem a ferramenta que completaram o item de trabalho selecionado, o motor de execução determina os próximos itens de trabalho a serem executados para aquela instância de processo específica. Dependendo das características do BPMS sendo utilizado, os usuários podem possuir controle avançado da lista de trabalho, permitindo a priorização de determinados itens de trabalho, a suspensão temporária dos mesmos e a transferência destes itens para outros usuários. A Figura 1.23 mostra um exemplo da lista de trabalho apresentada pela Bonita BPM ².

As *ferramentas de monitoramento e administração* permitem administrar a operação do BPMS, como, por exemplo, definir os usuários presentes e suas responsabilidades. Se um usuário estiver indisponível, como, por exemplo, devido a uma doença ou férias, ele não deve receber itens de trabalho a serem executados, portanto a ferramenta deve ser informada sobre esta indisponibilidade. Elas também permitem verificar os processos em andamento e quais as suas performances. Por exemplo, em casos excepcionais, itens de trabalho obsoletos podem ser removidos pelas ferramentas. Em um contexto maior, as

¹<https://camunda.com/>

²<https://www.bonitasoft.com/>

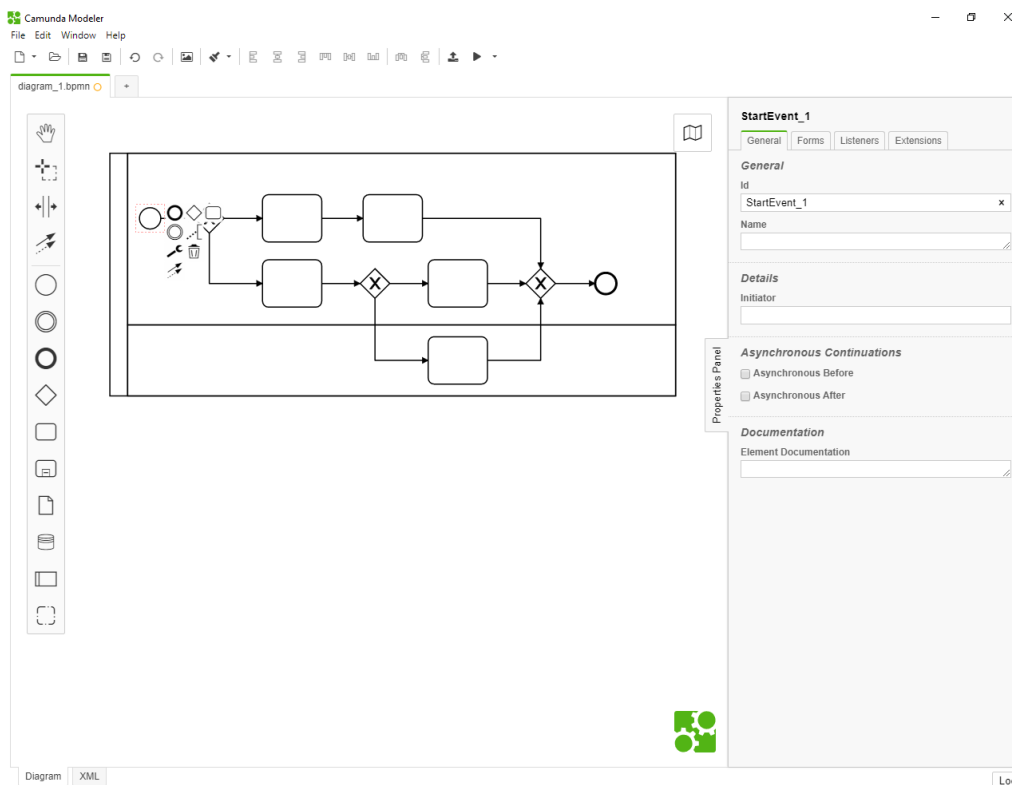


Figura 1.22. A ferramenta de modelagem *Camunda Modeler* da Camunda.

ferramentas de monitoramento e administração podem agregar dados de diferentes instâncias, como tempo de execução ou número de instâncias que terminaram com sucesso.

Os *serviços externos* são sistemas externos ao BPMS que oferecem serviços importantes, tais como bases de dados ou servidores de e-mail. Um processo de negócio tal como definido por um modelo de processo pode exigir o uso de um serviço externo para executar alguma funcionalidade já implementada pela aplicação externa. Neste caso, a aplicação externa expõe uma interface de serviço que será chamada pelo motor de execução com os dados necessários para executar determinada tarefa para uma instância de processo. Ao completar o requisitado, o serviço externo retorna o resultado ao motor de execução, que registra o item de trabalho como concluído.

1.4.2. Transformação de Modelo de Processo em Modelo Executável

Os modelos de processo construídos durante a fase de descoberta de processos representam o estado atual do trabalho sendo executado em uma organização. Porém, o nível de detalhamento deste modelos normalmente é baixo, apresentando de forma conceitual o que é feito neste processo sem esclarecer totalmente como as atividades são executadas ou quais os dados são gerenciados. Para a automação dos modelos de processo, é necessário a transformação dos modelos ao nível conceitual para modelos executáveis.

Modelos executáveis apresentam diferenças com relação aos modelos conceituais, pois eles servem para objetivos diferentes. Modelos conceituais focam na facilidade de compreensão e transmissão de ideias e não são necessariamente precisos, podendo conter

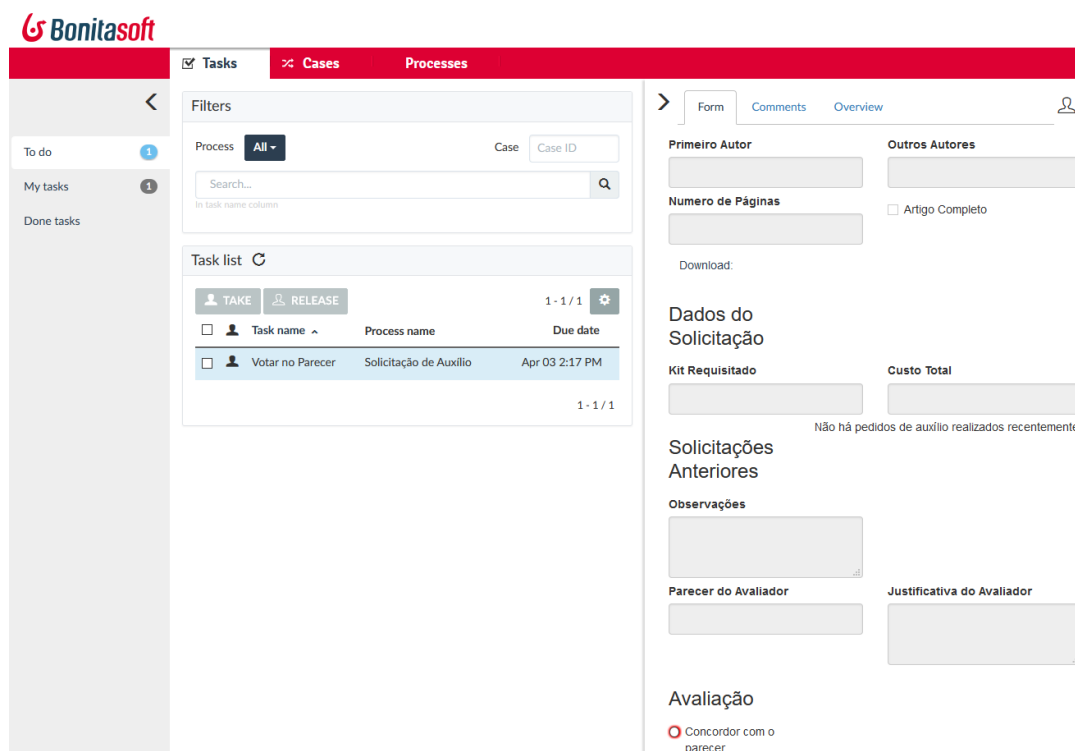


Figura 1.23. A lista de trabalho apresentada por Bonita BPM.

ambiguidades. O modelo executável, entretanto, necessita ser preciso, pois é interpretado por um BPMS [Dumas et al. 2018].

A transformação de um modelo conceitual para um executável segue cinco etapas [Dumas et al. 2018]: identificar as fronteiras de automatização, revisar as atividades manuais, completar o modelo de processo, atingir o nível de granularidade apropriado, e definir as propriedades de execução. A maneira como esta transformação é feita poderá ser acompanhada através do modelo apresentado na Figura 1.24, que representa um processo de solicitação de auxílio financeiro para a viagem a conferências.

Na etapa de *identificar as fronteiras de automatização* identifica-se quais atividades podem ser executadas pelo BPMS e quais não podem. As atividades são classificadas em tarefas *automatizadas*, *manuais* ou de *usuário*. As tarefas automatizadas são aquelas que podem ser realizadas pelo BPMS ou um serviço externo sem a intervenção humana. Em contrapartida, as tarefas manuais não podem ser realizadas com o auxílio de um software, necessitando que um participante do processo realize elas manualmente. As tarefas de usuário são um meio-termo entre as tarefas manuais e automatizadas, podendo ser realizadas no contexto do BPMS com a assistência de um participante do processo agindo como um usuário através da sua lista de trabalho no BPMS.

A distinção das tarefas é importante para identificar quais as barreiras que impedem a automatização do processo, assim como ele é representado no modelo conceitual. As tarefas manuais identificadas nesta etapa são analisadas na próxima etapa para determinar como realizar uma ligação entre a execução destas tarefas com o BPMS.

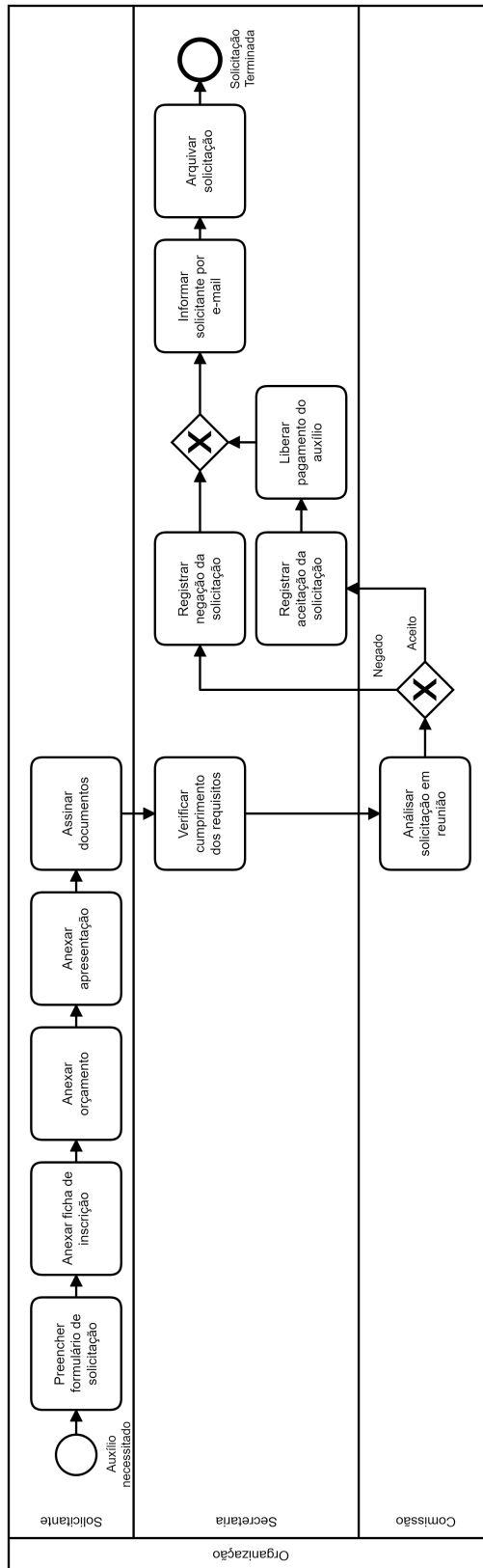


Figura 1.24. Modelo de solicitação de auxílio financeiro que queremos transformar em executável.

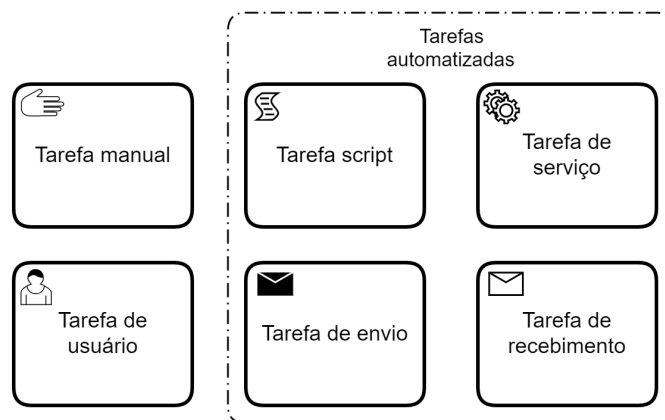


Figura 1.25. Marcadores que definem os tipos de tarefas em BPMN.

A BPMN 2.0 apresenta marcadores no canto superior-esquerdo das tarefas que permitem definir quais destas são automatizadas, manuais ou de usuário (ver Figura 1.25). O marcador de uma tarefa manual é uma mão, enquanto o marcador de tarefas de usuário é um ícone de usuário. Tarefas automatizadas são marcadas com 4 sub-tipos que representam a maneira com que a automatização é realizada. Estes sub-tipos são: tarefas *script*, nas quais o BPMS executa um código internamente; tarefas de *serviço*, são aquelas que o BPMS chama um serviço externo para realizar a atividade; tarefas de *envio*, que enviam uma mensagem para outra piscina do modelo; e tarefas de "recebimento", que recebem uma mensagem vinda de outra piscina do modelo.

Considerando as atividades apresentadas na Figura 1.24, percebe-se que várias atividades requerem a intervenção de uma pessoa. No início deste processo, o solicitante precisa enviar fisicamente um conjunto de documentos relevantes para a secretaria. A preparação destes documentos é uma série de atividades manuais, que não podem ser automatizadas completamente pelo BPMS. O mesmo pode ser dito pelas atividades *Verificar o cumprimento dos requisitos*, *Analisar solicitação em reunião* e *Arquivar solicitação*, que dependem do formato dos documentos enviados pelo solicitante. Por outro lado, o restante das atividades podem ser automatizadas como atividades *script* ou de envio de mensagem.

Na etapa de *revisar as atividades manuais* identifica-se como é possível integrar as atividades manuais no BPMS. Objetiva-se transformar as tarefas manuais em tarefas automatizadas ou de usuário. Para isso, deve-se criar alguma maneira de informar que a tarefa foi concluída. No caso de uma tarefa automatizada, pode-se considerar o uso de alguma tecnologia que esteja integrada com a execução da tarefa manual para notificar o BPMS sobre a mesma, através do recebimento de uma mensagem. Caso isto não seja possível, uma tarefa de usuário pode ser usada para indicar ao usuário que ele execute a tarefa manual e retorne ao BPMS para informar quando a tarefa seja concluída.

No caso do processo de solicitação de auxílio financeiro, a revisão das atividades manuais transforma o modelo para a versão apresentada na Figura 1.26. Percebe-se que as atividades realizadas pelo solicitante foram transformadas em atividades de usuário, para que o BPMS receba estas informações como os dados de entrada do processo.

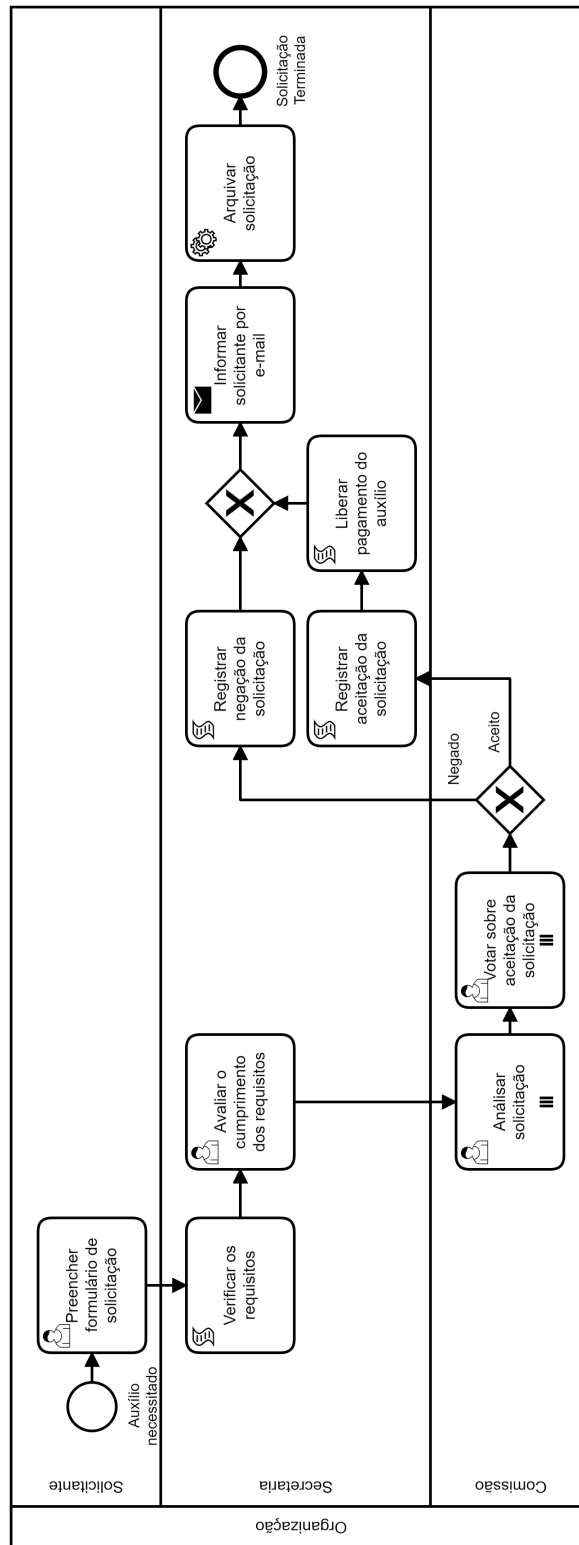


Figura 1.26. Modelo de solicitação de auxílio financeiro com as atividades manuais revisadas.

A atividade *verificar cumprimento dos requisitos*, feita pela secretaria, foi automatizada parcialmente, com uma tarefa *script* que verifica os requisitos através de um código e outra tarefa de usuário, que verifica o resultado da tarefa anterior. A análise da solicitação realizada anteriormente em uma reunião foi transformada em uma votação, dispensando a realização de uma reunião entre todos os participantes da comissão. Por fim, temos o arquivamento da solicitação, que foi transformada em uma atividade de serviço que chama o serviço adequado para armazenar os dados da solicitação que foi avaliada.

Alternativamente, pode não ser possível que uma tarefa manual seja automatizada. Por exemplo, tarefas manuais que frequentemente são realizadas sem ordem predefinida são inadequadas para a automatização no BPMS. Neste caso, estas tarefas devem ser isoladas ou removidas do modelo, para que o resto do processo seja automatizado. Entretanto, se grande parte do modelo for composto de tarefas manuais de difícil automatização, então a criação de um modelo executável pode não ser vantajoso ou ser impraticável.

Na etapa de *completar o modelo de processo* considera-se quais elementos precisam ser adicionadas ao modelo de processo conceitual para que este seja completo. É comum que este modelo não contenha todas as informações necessárias para a execução automatizada do processo, pois elas não foram consideradas importantes, assumindo que elas sejam de conhecimento comum, ou por elas não serem conhecidas pelos modeladores deste processo. Estas informações podem não afetar o entendimento do processo no nível conceitual, mas elas podem impedir a criação do modelo de processo executável.

Um caso de modelo que precisa ser completado é quando este não apresenta as situações negativas que podem acontecer durante a execução do processo. Estes modelos apresentam somente o cenário em que todas as atividades serão completadas com sucesso, negligenciando os problemas que podem acontecer e como estes problemas devem ser resolvidos para voltar à execução esperada do processo. Portanto, para completar estes modelos, deve-se considerar quais os possíveis casos de exceção que pode acontecer e modela-los, permitindo assim que estes aspectos sejam executados pelo BPMS.

Para completar o modelo também é preciso especificar todos os objetos de dados eletrônicos que servem de entrada e saída para as atividades do processo. Estes objetos podem ser utilizados pelas BPMS para tomar decisões de controle de fluxo do processo ou para passar informações entre atividades. A Figura 1.27 mostra o modelo de solicitação de auxílio financeiro completado. Anteriormente, o modelo não considerava o que aconteceria se a solicitação não cumprisse os requisitos avaliados pela secretaria. Também não era apresentado os documentos enviados pelo solicitante e quais atividades recebiam estes documentos durante suas execuções.

Na etapa *Atingir o nível de granularidade apropriado* é considerado o quão detalhado o modelo executável precisa ser durante sua execução pelo BPMS. Para isto, deve-se ter em mente que o BPMS gerencia a coordenação do processo e de suas atividades entre diferentes recursos que podem executa-las. Logo, se atividades são realizadas de forma sequencial pelo mesmo recurso, é preferível que estas atividades sejam agregadas em uma única atividade, de forma com que o BPMS não interfira no trabalho deste recurso com uma coordenação de atividades desnecessária.

Portanto, a agregação de tarefas feitas consecutivamente simplifica o modelo de

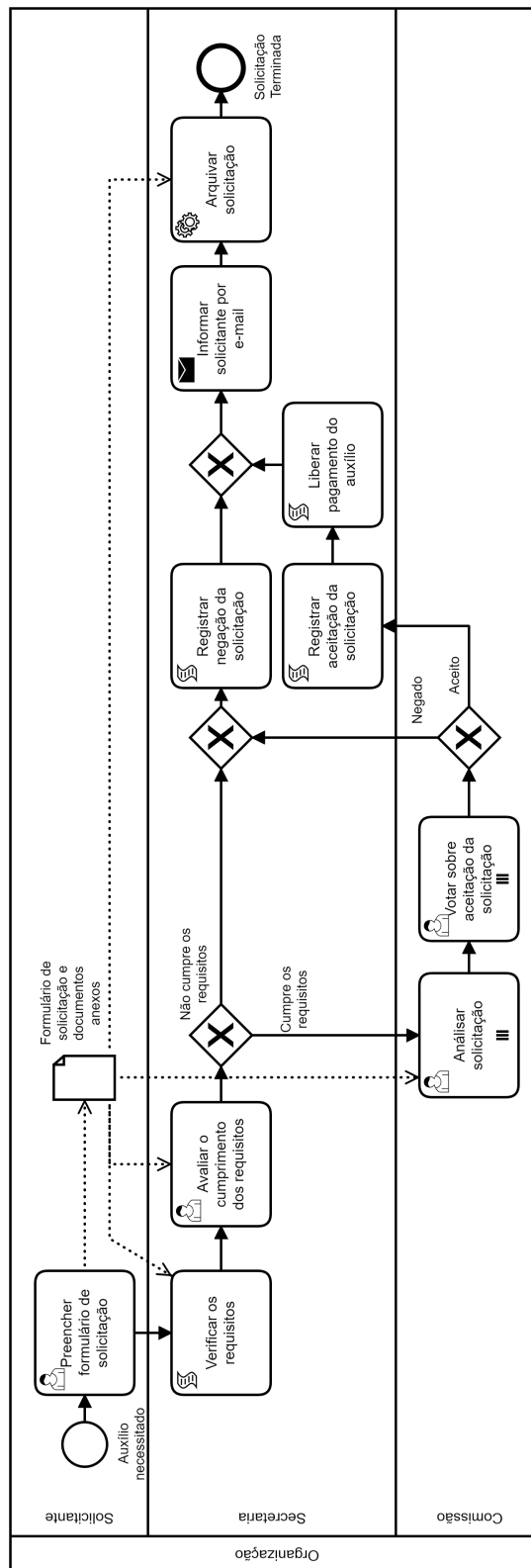


Figura 1.27. Modelo de solicitação de auxílio financeiro completo.

processo. O modelo apresentado na Figura 1.27 já apresenta a agregação das atividades realizadas pelo solicitante, pois se considera que na tarefa de usuário é possível criar um formulário que receba os documentos que anteriormente eram anexados manual e individualmente.

Por outro lado, é possível que não seja adequado agregar atividades mesmo quando elas são consecutivas e realizadas pelo mesmo recurso. Dependendo do que é realizado pelas atividades, pode ser desejável que o BPMS monitore quando cada atividade é completada, monitorando assim o progresso do processo e sendo possível gerenciar eventuais problemas de cada atividade. Também é possível que uma atividade precise ser dividida em múltiplas atividades, para permitir que cada atividade seja realizada por recursos diferentes.

Na última etapa *Definir as propriedades de execução* é definido como o processo funciona no BPMS, ou seja, a programação do processo. Não é suficiente, por exemplo, criar a atividade *Arquivar solicitação* e ligar ela com os dados de entrada. É necessário especificar as *propriedades de execução* desta atividade, como por exemplo, o serviço que é chamado pela atividade, os dados que devem ser enviados a este serviço e como receber os dados presentes na resposta desta chamada.

As propriedades de execução não possuem uma representação gráfica na BPMN 2.0. Eles são armazenados na representação textual da notação em código XML, o que facilita o uso de modelos BPMN 2.0 entre diferentes ferramentas. As regras definidas para a semântica dos elementos da BPMN 2.0 são expressas pelo código XML para que não exista ambiguidade entre as ferramentas de como deve funcionar a lógica de cada elemento da notação [Santos et al. 2019]. As possíveis propriedades de execução existentes são:

- *Variáveis do processo, mensagens, sinais, erros*, que definem os dados do processo que são gerenciados pelo BPMS e que persistem entre a execução de diferentes elementos do processo. Cada objeto de dados é uma variável do processo. O mesmo se aplica a mensagens, sinais e erros, que devem ser definidas com um tipo de dados.
- *Variáveis de atividades e eventos*, que definem os dados cujo tempo de vida é limitado ao tempo de execução de uma atividade ou evento. Um exemplo destas variáveis são as interfaces entre os dados de entrada e saída das atividades, dos eventos e dos dados presentes no processo ou em mensagens, sinais e erros.
- *Atividades de Serviços*, que necessitam estabelecer como a atividade irá se comunicar com um serviço externo para a sua execução. Para isto, defini-se uma *interface de serviço*, que determina os protocolos de comunicação com o serviço e onde o serviço pode ser localizado. Cada interface contém uma ou mais *operações de serviço* que descrevem maneiras de interagir com o serviço externo. Cada operação é associada com uma mensagem do modelo BPMN 2.0, ligando assim o tipo de dados da mensagem com a operação.
- *Atividades ou Eventos de Comunicação*, que enviam ou recebem mensagens para ou de um serviço externo. Elas são um caso especial das atividades de serviço, sendo caracterizadas pela direção única das mensagens. Por exemplo, uma atividade de envio de mensagem somente envia os seus dados de entrada ao serviço externo, mas não espera nenhuma resposta. Desta forma, o processo não precisa parar a espera

do envio ou recebimento de mensagens.

- *Atividades de Script*, que necessitam de um trecho de código para ser executado pelo BPMS. Este código é escrito por qualquer linguagem de programação que o BPMS aceite. Os dados de entrada da atividade servem como os parâmetros de entrada para o código, enquanto os dados de saída recebem os resultados do mesmo.
- *Atividades de Usuário*, nas quais precisa-se definir quais participantes do processo podem executar estas atividades, quais os dados de entradas são apresentados ao participante, quais dados de saída recebem os resultados e como os participantes irão interagir com a atividade. Para esta interação, cria-se um formulário eletrônico que organiza como os dados de entrada e saída são apresentados em uma tela. Também determina-se como atribuir o item de trabalho gerado pela atividade de usuário a um participante disponível para realizá-la e como ele é alertado sobre a disponibilidade deste item de trabalho.
- *Expressões de tarefas, eventos e fluxos de sequência*, nas quais são definidas as condições para a execução destes elementos. Por exemplo, os fluxos de sequência que partem de um *gateway* condicional possuem condições que precisam ser verificadas através de expressões que resultam em valores *booleanos*. Da mesma forma, atividades em *loop* precisam de uma expressão para determinar quando este *loop* termina. Eventos de tempo precisam de uma expressão temporal que determina uma duração, um intervalo ou um dia e hora específicos para quando o evento será ativado.
- *Propriedades específicas ao BPMS*, nas quais certas configurações que variam conforme o BPMS escolhido precisam ser definidas. Estas podem ser a maneira como as atividades de usuário são apresentadas aos seus recipientes, ou como um processo executável se conecta com o sistema da organização. BPMSs tipicamente oferecem *conectores de serviço* que implementam a conexão a sistemas comuns, como bases de dados, e-mail, agenda, entre outros.

1.4.3. Introdução a Tipos de BPMS Existentes

Para qualquer projeto de automação de processos, existem diversas plataformas de BPMS disponíveis no mercado. Os vendedores destas plataformas oferecem conjuntos de funcionalidades diferentes, podendo cobrir fases diferentes do ciclo de vida [Dumas et al. 2018]. Por exemplo, algumas plataformas oferecem livre acesso a módulos de modelagem de processo, reservando a requisição de licenças proprietárias para funcionalidades de análise, automação e monitoramento dos processos. Esta seção compara 4 plataformas BPMS diferentes (ver Tabela 1.1), duas de código-livre e duas proprietárias: A Bonita, a Signavio Process Manager, a Bizagi e a Camunda BPM.

A *Bonita* é uma plataforma de código-aberto criada em 2001. Ela é escrita em Java e possui versões disponíveis em Inglês, Português, e outras linguagens. O uso da plataforma pode ser feito através da *Community Edition*, que não possui custo, ou da *Subscription Edition*, que requer uma assinatura, mas apresenta mais funcionalidades. Ambas as edições permitem a automação de processos.

A *Signavio Process Manager* é um BPMS criado em 2009 com acesso restrito a compradores. Apesar disto, professores e estudantes de universidades podem ter acesso limitado a suas funcionalidades através de uma versão acadêmica da plataforma. A plata-

Tabela 1.1. Comparação de BPMS

	Bonita	Signavio	Bizagi	Camunda
Licença	GNUv2	Proprietária	Proprietária	Apache License
Suporte a Base de Dados	mySQL, SQL, Oracle	mySQL, SQL, Oracle	SQL, Oracle	mySQL, SQL, Oracle, Outros
Servidor Aplicação	jBoss, Tomcat, Outros	Tomcat	IIS	jBoss, Tomcat, Wildfly, Oracle, Outros
Modelagem Grátis	Sim	Não	Sim	Sim
Automação Grátis	Sim	Não	Não	Sim

forma está disponível em Inglês, Alemão e Francês.

A *Bizagi* é uma plataforma que oferece a automação de processo BPMN 2.0 através de três módulos: *Bizagi Modeler*, *Studio* e *Automation*. Os dois primeiros módulos são de livre acesso e eles oferecem as funcionalidades necessárias para a criação de modelos de processo e a automação destes em ambientes de teste. Entretanto, para utilizar estes modelos automatizados para trabalho, é necessário o módulo *Automation*, que possui custo. A Bizagi está disponível em Inglês, Português, e outras linguagens.

A *Camunda BPM* é uma plataforma de código-aberto criada em 2013. Ela é escrita em Java e está disponível em Inglês. O módulo de modelagem de processos da Camunda também está disponível a uso pela web³. Para automatizar modelos, é necessário que as propriedades de execução de processos sejam definidas na ferramenta de modelagem e executadas pelo módulo *BPMN Workflow Engine*, que contém o motor de execução do BPMS.

Um aspecto importante destas ferramentas é como elas garantem a qualidade dos modelos de processo criados e executados por elas. Diversos anti-padrões existem que podem causar problemas para a qualidade sintática, semântica e pragmática de um modelo [Rozman and Horvat 2008]. Apesar de que todas as ferramentas apresentadas nesta seção utilizam a BPMN 2.0 como notação padrão, as respostas delas em frente a anti-padrões encontrados são diferentes [de Brito Dias et al. 2019]. Por exemplo, para um mesmo anti-padrão, cada ferramenta pode apresentar uma mensagem de erro, de alerta, ou nenhuma mensagem. Similarmente, cada um destes BPMSs pode apresentar suporte a um conjunto de boas práticas de modelagem. Entretanto, quais boas práticas foram integradas nas ferramentas de modelagem e como elas são verificadas e apresentadas varia entre os BPMSs [Snoeck et al. 2015]. Logo, ao selecionar qual BPMS para a automação de processo, deve-se dar atenção ao suporte existente nestas ferramentas para a qualidade dos modelos de processo.

³<https://bpmn.io>

1.4.4. Mineração de Processos

Uma das vantagens da utilização de BPMS para a automação de processo são os *logs de execução* gerados [Dumas et al. 2018]. *Logs* de execução, também conhecidos como *logs de eventos*, contém gravações dos eventos que acontecem durante a execução de um modelo de processo, ou seja, de quais atividades, eventos e decisões aconteceram passo-a-passo [Van Der Aalst 2016]. Em cada evento é armazenado também outras informações, como os dados do processo naquele momento, quem foi responsável por executá-lo e quando ele foi executado. O BPMS utiliza estes *logs* para realizar o monitoramento e o controle do processo.

Após exportados, os *logs* permitem a análise da execução de processo através da mineração de seus dados. No contexto da BPM, chamamos esta análise de *mineração de processos*, onde se utiliza ambas as perspectivas de dados e de processos para extrair conhecimento sobre processos [Van Der Aalst 2016]. Por exemplo, uma das técnicas possíveis através da mineração de processo é a descoberta automática de processos a partir dos *logs* de eventos [Fahland and van der Aalst 2015]. Para isto, analisa-se com um algoritmo a ordem cronológica de eventos de cada instância de um processo. Com base nestas ordens, o algoritmo tenta determinar o melhor modelo de processo que consiga replicar as mesmas ordens de eventos analisadas. Os critérios que definem a qualidade deste modelo podem ser, por exemplo, a sua cobertura dos eventos presentes no log, a sua precisão, sua generalização ou sua simplicidade [Van Der Aalst 2016]. Outras técnicas possíveis pela mineração de processos é a verificação da conformidade, na qual se verifica se um modelo de processo consegue replicar os eventos gravados no *log*, e a análise de performance, onde se utiliza o conhecimento extraído dos *logs* e dos modelos de processo para avaliar a performance deste e possivelmente melhorá-lo [Van Der Aalst et al. 2012].

1.5. Resumo do Capítulo

Este capítulo apresentou a disciplina de BPM. Os principais conceitos de BPM foram discutidos, incluindo a definição e ilustração de processo de negócio. O capítulo também descreveu o ciclo de vida de BPM. Neste contexto, foram definidas as fases de identificação, descoberta, análise, redesenho, implementação e monitoramento, observando que o ciclo pode ser executado mais de uma vez em um projeto de BPM. O capítulo também apresentou os principais papéis relacionados ao ciclo de vida, incluindo o gerente de time, proprietário de processo, participante do processo, analista de processo, engenheiro de processo e grupo de BPM. Nem sempre, um projeto de BPM inclui todos estes papéis, mas é importante que a organização tenha um time de BPM em suas iniciativas de BPM.

O capítulo discutiu iniciativas que evidenciam as principais aplicabilidades de BPM, tal como na educação, meio ambiente, saúde e entretenimento. Importante observar que existem diversos outros domínios onde BPM vem sendo aplicado (ex.: bioinformática, atividades governamentais), objetivando maior documentação e padronização de processos. O capítulo descreveu a modelagem de processos de negócio, apresentando as técnicas para descoberta de processos, assim como os principais elementos básicos e avançados da BPMN 2.0, seguidos de exemplos ilustrativos. Neste contexto, foram apresentados diversas abordagens emergentes relacionados a este tema, com ênfase nos trabalhos de descoberta de processos.

Foi coberto também a importância da qualidade de modelos de processo em uma organização. A qualidade foi definida em três tipos, sintática, semântica e pragmática, e para cada tipo foi descrito quais características são avaliadas, como elas são avaliadas, quais problemas podem ocorrer em modelos sem este tipo de qualidade e como é possível garantir que modelos tenham esta qualidade. Com relação à qualidade pragmática, apresentou-se o que são boas práticas de modelagem, para que servem e um conjunto de boas práticas existentes na literatura.

Por fim, este capítulo discutiu a automação dos modelos de processo criados em BPMSs. Mostrou-se qual é a arquitetura comum de BPMSs, como ele ajuda a modelagem, execução e monitoramento de processos e como usuários podem interagir com os processos executados por ele. Após isto, foi demonstrado, através de um exemplo, como um modelo de processo pode ser transformado para uma versão que possa ser executada por um BPMS. Foi apresentado também uma comparação entre alguns BPMSs existentes no mercado e como podemos utilizar os logs de execução de processos obtidos de um BPMS para a mineração de processos.

Referências

- [Biazus et al. 2019] Biazus, M., dos Santos, C. H., Takeda, L. N., Palazzo Moreira de Oliveira, J., Fantinato, M., Mendling, J., and Thom, L. H. (2019). Software resource recommendation for process execution based on the organization's profile. In Hartmann, S., Küng, J., Chakravarthy, S., Anderst-Kotsis, G., Tjoa, A. M., and Khalil, I., editors, *Database and Expert Systems Applications - 30th International Conference, DEXA 2019, Linz, Austria, August 26-29, 2019, Proceedings, Part II*, volume 11707 of *Lecture Notes in Computer Science*, pages 118–128. Springer.
- [Caporale 2016] Caporale, T. (2016). A tool for natural language oriented business process modeling. In Hochreiner, C. and Schulte, S., editors, *Proceedings of the 8th ZEUS Workshop, Vienna, Austria, January 27-28, 2016.*, volume 1562 of *CEUR Workshop Proceedings*, pages 49–52. CEUR-WS.org.
- [de Brito Dias et al. 2019] de Brito Dias, C. L., Dani, V. S., Mendling, J., and Thom, L. H. (2019). Anti-patterns for process modeling problems: An analysis of BPMN 2.0-based tools behavior. In *Business Process Management Workshops*, pages 745–757. Springer International Publishing.
- [Dikici et al. 2018] Dikici, A., Turetken, O., and Demirors, O. (2018). Factors influencing the understandability of process models: A systematic literature review. *Information and Software Technology*, 93:112–129.
- [Drăgan et al. 2014] Drăgan, M., Ivana, D., and Arba, R. (2014). Business process modeling in higher education institutions. developing a framework for total quality management at institutional level. *Procedia Economics and Finance*, 16:95 – 103. 21st International Economic Conference of Sibiu 2014, IECS 2014 Prospects of Economic Recovery in a Volatile International Context: Major Obstacles, Initiatives and Projects.
- [Dumas et al. 2012] Dumas, M., Rosa, M. L., Mendling, J., Mäesalu, R., Reijers, H. A., and Semenenko, N. (2012). Understanding business process models: The costs and

- benefits of structuredness. In *Proc. of the 24th Int. Conf. on Advanced Information Systems Engineering (CAiSE)*, volume 7328 of *LNCS*, pages 31–46, Berlin Heidelberg. Springer.
- [Dumas et al. 2018] Dumas, M., Rosa, M. L., Mendling, J., and Reijers, H. A. (2018). *Fundamentals of Business Process Management, Second Edition*. Springer Berlin Heidelberg.
- [Enríquez et al. 2019] Enríquez, F., Troyano, J. A., and Romero-Moreno, L. M. (2019). Using a business process management system to model dynamic teaching methods. *The Journal of Strategic Information Systems*, 28(3):275 – 291.
- [Fahland and van der Aalst 2015] Fahland, D. and van der Aalst, W. M. (2015). Model repair — aligning process models to reality. *Information Systems*, 47:220–243.
- [Ferreira et al. 2017] Ferreira, R. C. B., Thom, L. H., and Fantinato, M. (2017). A semi-automatic approach to identify business process elements in natural language texts. In Hammoudi, S., Smialek, M., Camp, O., and Filipe, J., editors, *ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems, Volume 3, Porto, Portugal, April 26-29, 2017*, pages 250–261. SciTePress.
- [Figl 2017] Figl, K. (2017). Comprehension of procedural visual business process models: A literature review. *Business and Information Systems Engineering*, 59(1):41–67.
- [Friedrich et al. 2011] Friedrich, F., Mendling, J., and Puhlmann, F. (2011). Process model generation from natural language text. In Mouratidis, H. and Rolland, C., editors, *Advanced Information Systems Engineering - 23rd International Conference, CAiSE 2011, London, UK, June 20-24, 2011. Proceedings*, volume 6741 of *Lecture Notes in Computer Science*, pages 482–496. Springer.
- [Gassen et al. 2014] Gassen, J. B., Mendling, J., Thom, L. H., and Oliveira, J. P. M. d. (2014). Business process modeling: Vocabulary problem and requirements specification. In *Proc. of the 32nd ACM Int. Conf. on The Design of Communication (SIGDOC)*, pages 1–10, New York, USA. ACM.
- [Kalsing et al. 2013] Kalsing, A. C., Iochpe, C., Thom, L. H., and do Nascimento, G. S. (2013). Evolutionary learning of business process models from legacy systems using incremental process mining. In Hammoudi, S., Maciaszek, L. A., Cordeiro, J., and Dietz, J. L. G., editors, *ICEIS 2013 - Proceedings of the 15th International Conference on Enterprise Information Systems, Volume 2, Angers, France, 4-7 July, 2013*, pages 58–69. SciTePress.
- [Krogstie 2012] Krogstie, J. (2012). *Model-Based Development and Evolution of Information Systems*. Springer London, London.
- [Krogstie 2016] Krogstie, J. (2016). *Quality in Business Process Modeling*. Springer International Publishing.

- [Lenz et al. 2012] Lenz, R., Peleg, M., and Reichert, M. (2012). Healthcare process support: Achievements, challenges, current research. *International Journal of Knowledge-Based Organizations (IJKBO)*, 2(4).
- [Leopold 2013] Leopold, H. (2013). *Natural Language in Business Process Models - Theoretical Foundations, Techniques, and Applications*, volume 168 of *Lecture Notes in Business Information Processing*. Springer.
- [Leopold and Mendling 2012] Leopold, H. and Mendling, J. (2012). Automatic derivation of service candidates from business process model repositories. In Abramowicz, W., Kriksciuniene, D., and Sakalauskas, V., editors, *BIS*, volume 117 of *Lecture Notes in Business Information Processing*, pages 84–95. Springer.
- [Leopold et al. 2016] Leopold, H., Mendling, J., and Gunther, O. (2016). Learning from quality issues of BPMN models from industry. *IEEE Software*, 33(4):26–33.
- [Leopold et al. 2012] Leopold, H., Mendling, J., and Polyvyanyy, A. (2012). Generating natural language texts from business process models. In *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 64–79. Springer International Publishing.
- [Mendling et al. 2019] Mendling, J., Leopold, H., Thom, L. H., and van der Aa, H. (2019). Natural language processing with process models (NLP4RE report paper). In *Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, Live Studies Track, and Poster Track co-located with the 25th International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2019), Essen, Germany, March 18th, 2019*, volume 2376 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- [Mendling et al. 2010] Mendling, J., Reijers, H. A., and van der Aalst, W. M. P. (2010). Seven process modeling guidelines (7pmg). *Inf. Softw. Technol.*, 52(2):127–136.
- [Mendling and Strembeck 2008] Mendling, J. and Strembeck, M. (2008). Influence factors of understanding business process models. *Lecture Notes in Business Information Processing*, 7 LNBIP:142–153.
- [Nelson et al. 2012] Nelson, H. J., Poels, G., Genero, M., and Piattini, M. (2012). A conceptual modeling quality framework. *Software Quality Journal*, 20(1):201–228.
- [Object Management Group (OMG) 2011] Object Management Group (OMG) (2011). Business Process Model and Notation (BPMN) Version 2.0. *Business*, 50(January):170.
- [Ottenssooser et al. 2012] Ottenssooser, A., Fekete, A., Reijers, H. A., Mendling, J., and Menictas, C. (2012). Making sense of business process descriptions: An experimental comparison of graphical and textual notations. *Journal of Systems and Software*, 85(3):596–606.
- [Priego-Roche et al. 2012] Priego-Roche, L.-M., Thom, L. H., Front, A., Rieu, D., and Mendling, J. (2012). Business process design from virtual organization intentional models. In Ralyté, J., Franch, X., Brinkkemper, S., and Wrycza, S., editors, *CAiSE*, volume 7328 of *Lecture Notes in Computer Science*, pages 549–564. Springer.

- [Reichert and Pryss 2017] Reichert, M. and Pryss, R. (2017). Flexible support of health-care processes. In Combi, C., Pozzi, G., and Veltri, P., editors, *Process Modeling and Management for Healthcare*, pages 35–66. Taylor & Francis Group.
- [Reijers et al. 2015] Reijers, H. A., Mendling, J., and Recker, J. (2015). Business process quality management. In *Handbook on Business Process Management 1*, pages 167–185. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Rozman and Horvat 2008] Rozman, T. and Horvat, R. (2008). Analysis of most common process modelling mistakes in bpmn process models. In *2008 BPM and Workflow Handbook*. University of Maribor Slovenia.
- [Santos et al. 2019] Santos, C. H. D., Thom, L. H., Cota, É., and Fantinato, M. (2019). Supporting BPMN tool developers through meta-algorithms. *International Journal of Business Information Systems*, 32(4):460.
- [Silva et al. 2019] Silva, T. S., Avila, D. T., Flesch, J. A., Peres, S. M., Mendling, J., and Thom, L. H. (2019). A service-oriented architecture for generating sound process descriptions. In *23rd IEEE International Enterprise Distributed Object Computing Conference, EDOC 2019, Paris, France, October 28-31, 2019*, pages 1–10. IEEE.
- [Snoeck et al. 2015] Snoeck, M., de Oca, I. M.-M., Haegemans, T., Scheldeman, B., and Hoste, T. (2015). Testing a selection of BPMN tools for their support of modelling guidelines. In *Lecture Notes in Business Information Processing*, pages 111–125. Springer International Publishing.
- [Sánchez-Ferreres et al. 2020] Sánchez-Ferreres, J., Delicado, L., Andaloussi, A. A., Burattin, A., Calderón-Ruiz, G., Weber, B., Carmona, J., and Padró, L. (2020). Supporting the process of learning and teaching process models. *IEEE Transactions on Learning Technologies*, pages 1–1.
- [Thom et al. 2010] Thom, L., Reichert, M., Iochpe, C., and de Oliveira, J. P. M. (2010). Why rigid process management technology hampers computerized support of health-care processes. In *X Workshop on Medical Informatics (WIM'2010), Workshop held in conjunction with SBC'10 Congress*, pages 1522–1531.
- [Thom et al. 2009] Thom, L. H., Reichert, M., and Iochpe, C. (2009). On the support of workflow activity patterns in process modeling tools: Purpose and requirements. In *Proc. of the 3rd Workshop on Business Process Management (WBPM)*, pages 1–6, Porto Alegre, Brazil. Brazilian Computing Society.
- [Van Der Aalst 2016] Van Der Aalst, W. (2016). *Process Mining*. Springer Berlin Heidelberg.
- [Van Der Aalst et al. 2012] Van Der Aalst, W., Adriansyah, A., de Medeiros, A. K. A., Arcieri, F., Baier, T., Blicke, T., Bose, J. C., van den Brand, P., Brandtjen, R., Buijs, J., Burattin, A., Carmona, J., Castellanos, M., Claes, J., Cook, J., Costantini, N., Curbura, F., Damiani, E., de Leoni, M., Delias, P., van Dongen, B. F., Dumas, M., Dustdar, S., Fahland, D., Ferreira, D. R., Gaaloul, W., van Geffen, F., Goel, S., Günther, C.,

- Guzzo, A., Harmon, P., ter Hofstede, A., Hoogland, J., Ingvaldsen, J. E., Kato, K., Kuhn, R., Kumar, A., Rosa, M. L., Maggi, F., Malerba, D., Mans, R. S., Manuel, A., McCreesh, M., Mello, P., Mendling, J., Montali, M., Motahari-Nezhad, H. R., zur Muehlen, M., Munoz-Gama, J., Pontieri, L., Ribeiro, J., Rozinat, A., Pérez, H. S., Pérez, R. S., Sepúlveda, M., Sinur, J., Soffer, P., Song, M., Sperduti, A., Stilo, G., Stoel, C., Swenson, K., Talamo, M., Tan, W., Turner, C., Vanthienen, J., Varvaressos, G., Verbeek, E., Verdonk, M., Vigo, R., Wang, J., Weber, B., Weidlich, M., Weijters, T., Wen, L., Westergaard, M., and Wynn, M. (2012). Process mining manifesto. In *Business Process Management Workshops*, pages 169–194, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Van Der Aalst 2013] Van Der Aalst, W. M. P. (2013). Business process management: A comprehensive survey. *ISRN Software Engineering*, 2013:1–37.
- [vom Brocke et al. 2012] vom Brocke, J., Seidel, S., and Recker, J., editors (2012). *Green Business Process Management*. Springer Berlin Heidelberg.
- [Vuksic and Bach 2012] Vuksic, V. B. and Bach, M. P. (2012). Simulation Games in Business Process Management Education. *International Journal of Industrial and Systems Engineering*, 6(9):6.
- [Weber et al. 2009] Weber, B., Sadiq, S. W., and Reichert, M. (2009). Beyond rigidity - dynamic process lifecycle support. *Comput. Sci. Res. Dev.*, 23(2):47–65.
- [Wesenberg 2011] Wesenberg, H. (2011). Enterprise modeling in an agile world. In *Lecture Notes in Business Information Processing*, pages 126–130. Springer Berlin Heidelberg.
- [Weske 2012] Weske, M. (2012). *Business Process Management*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Weske 2019] Weske, M. (2019). *Business Process Management - Concepts, Languages, Architectures, Third Edition*. Springer.