

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MILLER BIAZUS

**Modelagem e Análise de Conformidade do
Processo Presente em Estratégias
Computacionais de Atracamento Molecular**

Monografia apresentada como requisito parcial para
a obtenção do grau de Bacharel em Ciência da
Computação

Orientador: Profa. Dra. Lucinéia Heloisa Thom
Co-orientador: Prof. Dr. Márcio Dorn

Porto Alegre
2015

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb

Coordenador do Curso de Ciência de Computação: Prof. Carlos Arthur Lang Lisbôa

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Em primeiro lugar, gostaria de expressar minha gratidão à orientadora, Profa. Dra. Lucinéia Heloisa Thom, pelo apoio expressivo durante a escrita da monografia, por me acolher em seu grupo de pesquisa e por sua motivação e paciência ao longo do ano. Agradeço, também, ao Prof. Dr. Márcio Dorn, coorientador, por dissipar seu conhecimento no decorrer da monografia, além de sua disponibilidade para dúvidas, sugestões e reuniões. As opiniões e críticas de ambos os professores auxiliaram muito em meu desenvolvimento acadêmico.

Gostaria de agradecer à Profa. Dra. Renata de Matos Galante e à Profa. Dra. Mariana Recamonde Mendoza, pela disponibilidade e flexibilidade ao aceitar o convite e comporem a banca de avaliação.

Gostaria de agradecer meus amigos da universidade, os quais acompanharam minha trajetória acadêmica e compartilharam alegrias e dificuldades, além da companhia para rodadas de cerveja. Amigos que levo para a vida e que serão parte do meu futuro.

Palavras não podem expressar o quanto sou grato à minha família, a qual me apoiou durante toda a minha vida, em quaisquer momentos que precisei. Agradeço à vocês por não medirem esforços para que eu chegasse até esta etapa de minha vida.

Agradeço à minha namorada Caroline, que me inspira cada dia mais, me suportando, apoiando, encorajando e fazendo de mim uma pessoa melhor. Agradeço por me compreender e por trazer paz na correria de cada semestre. O agradecimento se estende à sua família, a qual me acolheu como um filho e que sempre me deseja o melhor.

Agradeço, enfim, à todas as pessoas que contribuíram de uma forma ou de outra para minha formação acadêmica, incluindo meus amigos de Canela, de Porto Alegre, da Alemanha, e o corpo docente do Instituto de Informática.

RESUMO

A modelagem de processos com características dinâmicas e vocabulário especialista, tal como processos de atracamento molecular é complexa. Tal processo envolve macro atividades, dentre elas preparação de receptor, preparação de ligante, adição de solvente, identificação do sítio de ligação, etc, podendo ser simulado por uma imensa quantidade de softwares existentes (por exemplo Gold, Dockthor e Autodock Vina). Porém, cada um desses softwares existentes pode apresentar comportamento diverso para realizar as macro atividades, resultando em processos de atracamento molecular divergentes entre si e em comparação com o processo proposto pela literatura de biologia molecular. O presente trabalho objetiva modelar o processo de atracamento molecular a partir da literatura, gerando um modelo de referência. O modelo de referência do processo de atracamento molecular é validado com especialistas do domínio e verificado usando-se Redes de Petri. Após, realiza-se um estudo comparativo entre softwares para atracamento molecular, identificando convergências e divergências entre os processos de cada software e com o processo identificado a partir da literatura. As principais contribuições deste trabalho são a documentação validada e verificada do processo de atracamento molecular em uma notação para modelagem de processos e um estudo comparativo entre os processos de dois softwares existentes a fim de analisar a conformidade dos mesmos com o modelo de referência. Os resultados deste trabalho são um modelo de referência de atracamento molecular verificado e validado, além de uma análise de dois softwares de atracamento molecular.

Palavras-chave: Atracamento molecular. Gerenciamento de processos de negócio. Notação de modelagem de processos de negócio (BPMN). Modelo de referência. Análise de conformidade.

ABSTRACT

The designing of particular processes with specialized vocabulary and dynamic characteristics including those encompassed by existent computational strategies for molecular docking are very complex. Molecular docking process includes macro-activities (e.g. receptor and ligand preparation, solvent adding and binding site identification) which can be simulated by several and different softwares (e.g. Gold, DockThor, Autodock Vina). However, the softwares may have different behavior for performing the macro activities which implies different molecular docking processes in particular when compared with the process proposed by the molecular biology literature. This work proposes a literature-based design of the process of molecular docking, generating a reference model. The reference model of the molecular docking process will be validated with specialists from the docking domain and will be verified using Petri Nets. Thereafter, a comparative study of softwares for molecular docking is conducted in order to identify convergences and divergences between the processes of each software and the process identified from the literature. The main contributions of this work are a validated and verified documentation of the molecular docking process in a notation for modeling processes and a comparative study of compliance between the processes of the selected softwares and the reference model. The results of this work are a validated and verified reference model, complemented with an analysis of two molecular docking softwares.

Keywords: molecular docking, Business process management, Business process model and notation (BPMN), Reference model, Compliance analysis.

LISTA DE FIGURAS

Figura 2.1 Elementos receptor e ligante no atracamento molecular (HERNÁNDEZ-SANTOYO et al., 2013)	14
Figura 2.2 Modelos implícitos de solvatação (KLEINJUNG; FRATERNALI, 2014)	18
Figura 3.1 Ciclo de vida de processos (DUMAS et al., 2013).....	21
Figura 3.2 Representação do elemento Tarefa	22
Figura 3.3 Representação do elemento Fluxo de sequência	22
Figura 3.4 Representação do elemento Desvio exclusivo divergente	23
Figura 3.5 Representação do elemento Desvio exclusivo convergente.....	23
Figura 3.6 Representação do elemento Desvio paralelo divergente	23
Figura 3.7 Representação do elemento Desvio paralelo convergente.....	24
Figura 3.8 Representação do elemento Evento de início	24
Figura 3.9 Representação do elemento Evento de fim.....	24
Figura 3.10 Representação do elemento Subprocesso	24
Figura 3.11 Representação do elemento Associação	24
Figura 3.12 Representação do elemento Anotação de texto	25
Figura 5.1 Modelo de referência - Processo principal de atracamento molecular	30
Figura 5.2 Modelo de referência - Subprocesso Selecionar Moléculas Interagentes	30
Figura 5.3 Modelo de referência - Subprocesso Determinar método de busca para o sistema	31
Figura 5.4 Modelo de referência - Subprocesso Determinar função de avaliação para o sistema.....	31
Figura 5.5 Modelo de referência - Subprocesso Preparar Ligante.....	32
Figura 5.6 Modelo de referência - Subprocesso Preparar Receptor.....	33
Figura 5.7 Modelo de referência - Subprocesso adicionar cofatores	33
Figura 5.8 Modelo de referência - Subprocesso Determinar sitio de ligação	34
Figura 6.1 Compreensibilidade dos modelos	37
Figura 7.1 Exemplo de Rede de Petri.....	39
Figura 7.2 Exemplo de disparo de transição	40
Figura 7.3 Mapeamento dos elementos tarefa, eventos e desvios de BPMN para Redes de Petri (DIJKMAN; DUMAS; OUYANG, 2008)	42
Figura 7.4 Mapeamento de um subprocesso para Redes de Petri (DIJKMAN; DUMAS; OUYANG, 2008)	42
Figura 7.5 Chamada de subprocesso através de uma atividade SI (DIJKMAN; DUMAS; OUYANG, 2008)	42
Figura 7.6 Processo principal, em Redes de Petri	44
Figura 7.7 Subprocesso Selecionar Moléculas Interagentes, em Redes de Petri	44
Figura 7.8 Subprocesso Preparar o receptor, em Redes de Petri.....	44
Figura 7.9 Subprocesso Preparar o ligante, em Redes de Petri.....	45
Figura 7.10 Subprocesso Adicionar cofatores, em Redes de Petri	45
Figura 7.11 Subprocesso Determinar função de avaliação para o sistema, em Redes de Petri	45
Figura 7.12 Subprocesso Determinar método de busca para o sistema, em Redes de Petri	45
Figura 7.13 Subprocesso Determinar sítio de ligação, em Redes de Petri.....	45
Figura 8.1 Processo minerado do software Dokthor	56
Figura 8.2 Modelo de referência	57

Figura 8.3 Processo minerado do software Autodock Vina..... 57

LISTA DE TABELAS

Tabela 6.1	Validação da modelagem.....	36
Tabela 7.1	Verificação da Rede de Petri.....	47
Tabela 8.1	Log de eventos do processo do software DockThor - Case 1.....	49
Tabela 8.2	Log de eventos do processo do software DockThor - Case 2.....	50
Tabela 8.3	Log de eventos do processo do software DockThor - Case 3.....	50
Tabela 8.4	Log de eventos do processo do software DockThor - Case 4.....	51
Tabela 8.5	Log de eventos do processo do software DockThor - Case 5.....	51
Tabela 8.6	Log de eventos do processo do software Autodock Vina - Case 1.....	52
Tabela 8.7	Log de eventos do processo do software Autodock Vina - Case 2.....	52
Tabela 8.8	Log de eventos do processo do software Autodock Vina - Case 3.....	53
Tabela 8.9	Log de eventos do processo do software Autodock Vina - Case 4.....	53
Tabela 8.10	Log de eventos do processo do software Autodock Vina - Case 5.....	54

LISTA DE ABREVIATURAS E SIGLAS

BPM	Business Process Management
BPMN	Business Process Modeling and Notation
PN	Rede de Petri
SI	Invocação de subprocesso

SUMÁRIO

1 INTRODUÇÃO	11
2 PROCESSO DE ATRACAMENTO MOLECULAR	14
3 GERENCIAMENTO DE PROCESSOS DE NEGÓCIO E NOTAÇÃO DE MO- DELAGEM DE PROCESSOS DE NEGÓCIO	20
3.1 Gerenciamento de Processos de Negócio	20
3.2 Notação de Modelagem de Processos de Negócio.....	22
4 METODOLOGIA DO TRABALHO	26
5 MODELAGEM DO PROCESSO DE ATRACAMENTO MOLECULAR EM BPMN	29
5.1 Modelos de Referência.....	29
5.2 Modelo de Referência do Processo de Atracamento Molecular	29
6 VALIDAÇÃO DA MODELAGEM DO PROCESSO DE ATRACAMENTO MO- LECULAR	35
7 VERIFICAÇÃO DA MODELAGEM	39
7.1 Redes de Petri	39
7.2 Mapeando BPMN para Redes de Petri.....	40
7.2.1 Mapeamento de tarefas, eventos e desvios	41
7.2.2 Mapeamento de subprocessos.....	43
7.2.3 Marcação inicial.....	43
7.3 Modelagem do processo usando Redes de Petri.....	43
7.4 Verificação das Redes de Petri	46
8 ANÁLISE COMPARATIVA E DE CONFORMIDADE DO PROCESSO DE RE- FERÊNCIA COM OS PROCESSOS DE SOFTWARES DE ATRACAMENTO MOLECULAR	48
8.1 Análise de Conformidade	55
8.1.1 Conformidade entre modelo de referência e modelo de processo do software DockThor	58
8.1.2 Conformidade entre modelo de referência e modelo de processo do software Auto- dock Vina	58
9 CONCLUSÃO	59
REFERÊNCIAS	61
APÊNDICES	65

1 INTRODUÇÃO

Bioinformática estrutural visa o entendimento de processos biológicos e a aplicação de técnicas computacionais para aprender e organizar informação estrutural associada a macromoléculas (ALTMAN; DUGAN, 2005). Um dos principais problemas em bioinformática estrutural é relacionado com o desenho de fármacos *in silico* (LENGAUER; RAREY, 1996). Nos primeiros estágios da descoberta de fármacos o foco é no entendimento da interação entre um fármaco candidato (ligante) e uma proteína de destino (receptor), e isso leva ao objetivo de testar um receptor com um conjunto de fármacos candidatos (KITCHEN et al., 1996). Métodos de atracamento molecular são utilizados regularmente para predição das interações proteína-ligante e para auxiliar na seleção de potenciais moléculas como parte do *virtual screening*¹ de grandes bancos de dados (ELOKELY; DOERKSEN, 2013) (GHOSH et al., 2006). Atracamento molecular pode ser definido como uma metodologia que lida com o problema de prever a ligação não-covalente de um receptor (macromolécula, geralmente uma enzima ou proteína) e um ligante (molécula menor) em um nível atômico. A qualidade dessa ligação depende de uma função de avaliação e dos métodos computacionais aplicados. Uma simulação de atracamento molecular começa com estruturas de um receptor e um ligante desvinculadas (MENG et al., 2011). Durante a simulação o algoritmo de atracamento testa uma grande variedade de conformações e orientações do ligante a fim de achar a melhor afinidade de ligação entre receptor e ligante de acordo com uma função de avaliação.

A modelagem de processos particulares com vocabulário especialista e características dinâmicas como as existentes em estratégias computacionais para atracamento molecular é muito complexa, não somente pela variedade de estratégias computacionais, mas também porque requer o conhecimento de termos de um domínio específico que pode levar a problemas de interpretação, ambiguidades e falta de conformidade entre os *stakeholders*. O processo de atracamento molecular inclui macroatividades como preparação do receptor e do ligante, adição de solvente e identificação do sítio de ligação, as quais podem ser simuladas por diversos softwares (por exemplo DockThor (MAGALHÃES et al., 2014) e Autodock Vina (TROTT; OLSON, 2010)). Contudo, cada software possui suas próprias técnicas para realizar as macroatividades, o que implica processos de atracamento molecular divergentes, em particular quando comparados com o processo proposto pela literatura de biologia molecular.

Considerando a modelagem de processos de negócio, especificamente, a Notação de Modelagem de Processos de Negócio (BPMN) é reconhecida como padrão pela *Object Mana-*

¹Técnica computacional utilizada para busca de moléculas com estruturas passíveis de ligação com uma droga de interesse. Tal molécula é, em geral, uma proteína ou uma enzima

gement Group (OMG) (RECKER, 2010). BPMN vem sendo usada para uma grande variedade de propósitos de modelagem de processos, incluindo áreas como análise de negócios, análise de requerimentos, descrição, documentação, simulação e execução de processos (MATEJA et al., 2015). Em métodos tradicionais para realizar a modelagem de processos o analista depende de informações obtidas e organizadas a partir de especialistas no domínio (DUMAS et al., 2013).

O trabalho tem por hipóteses fundamentais: a) é possível representar um processo complexo, tal como o processo de atracamento molecular, em BPMN, utilizando como ponto de partida a literatura de biologia molecular e b) tal representação pode servir como base para a análise de conformidade com os processos dos softwares de atracamento molecular.

O principal objetivo deste trabalho é propor um modelo de referência, baseado na literatura, do processo de atracamento molecular utilizando a notação gráfica padrão BPMN. É feita uma validação do modelo de referência de atracamento molecular com especialistas do domínio de atracamento molecular, além de uma verificação da modelagem através de Redes de Petri.

As principais contribuições esperadas do presente trabalho incluem, mas não são restritas a:

- Documentação do processo de atracamento molecular em uma notação para modelagem de processos, propondo uma visão padronizada e sistemática do processo que pode servir de referência para o aprendizado e para auxílio no desenvolvimento de softwares para atracamento molecular;
- Sustentar a expansão de referências para criação de ferramentas e serviços de atracamento molecular;
- Instigar o uso de BPMN para representação de processos complexos incorporados na literatura;
- Metodologia utilizada neste trabalho pode servir de inspiração para outros trabalhos em BPM, desde modelar um processo a partir da literatura até validá-lo com questionários e verificá-lo através de Redes de Petri;
- Apresentar um estudo comparativo e de conformidade entre os processos de dois softwares existentes de atracamento molecular.

Como resultado do trabalho, é apresentado o modelo de referência do processo de atracamento molecular validado e verificado, além de uma análise entre dois softwares existentes de atracamento molecular, comparando-os com o modelo de referência.

O restante deste trabalho é estruturado da seguinte maneira: o Capítulo 2 apresenta uma visão geral do processo de atracamento molecular. O Capítulo 3 apresenta fundamentos de

modelagem de processos de negócio, bem como elementos da notação BPMN. O Capítulo 4 apresenta a metodologia utilizada para elaboração deste trabalho. O Capítulo 5 apresenta uma modelagem do processo de atracamento molecular em BPMN realizada a partir da literatura, a qual foi validada e complementada através de entrevistas com especialistas. O resultado desta modelagem é um modelo de referência. O modelo de referência é um modelo genérico baseado na literatura, não sendo específico a uma técnica. No Capítulo 6 é feita a validação do modelo de referência utilizando-se questionários, e no capítulo 7 é feita a verificação da modelagem através de Redes de Petri. No Capítulo 8 é feita uma análise de conformidade do processo de atracamento molecular de referência com os processos utilizados em dois softwares de atracamento molecular. O Capítulo 9 apresenta conclusões e trabalhos futuros do trabalho realizado.

2 PROCESSO DE ATRACAMENTO MOLECULAR

O processo para a descoberta de um novo fármaco é demasiadamente demorado e caro (ADAMS; BRANTNER, 2010); em virtude disso, técnicas computacionais vêm sendo amplamente utilizadas para minimizar custos. Uma das atividades chave no processo de descoberta de um novo fármaco é o atracamento molecular, uma abordagem *in silico* que consiste na predição de modos de ligação e afinidade de ligação entre i) um receptor (macromolécula) com estrutura tridimensional conhecida, o qual possui um sítio de ligação de interesse, e ii) um ligante (molécula), o qual atuará no sítio de ligação do receptor, alterando (seja inibindo, seja ativando) determinada função do mesmo (figura 2.1).

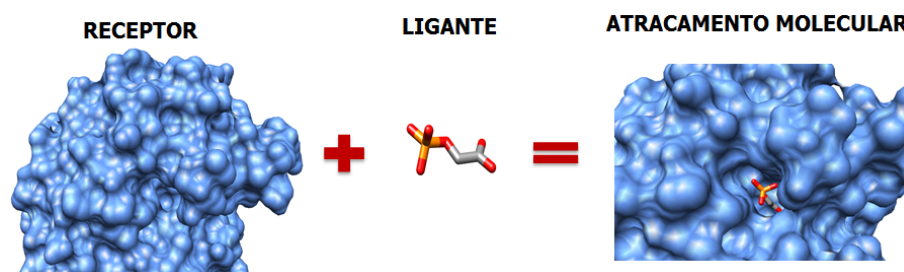


Figura 2.1 – Elementos receptor e ligante no atracamento molecular (HERNÁNDEZ-SANTOYO et al., 2013)

As moléculas utilizadas para o atracamento molecular podem possuir um certo grau de flexibilidade, o que possibilita diferenciar o tipo de atracamento molecular em três abordagens distintas (GUEDES; MAGALHÃES; DARDENNE, 2014):

- *Rigid-body docking*. Neste caso, ambas as moléculas são rígidas, e apenas os graus de liberdade translacionais e rotacionais podem ser explorados na molécula ligante (6 graus ao todo). É comum em atracamento molecular proteína-proteína; porém, atualmente, quase nenhum método de atracamento molecular utiliza exclusivamente esta abordagem;
- *Flexible-ligand docking*. Tratar ambos os corpos como rígidos pode ser inconveniente por não ser possível simular mudanças conformacionais em quaisquer uma das moléculas. Para contornar essa situação, *flexible-ligand docking* permite que todos os graus de liberdade (translacional, rotacional e conformacional) do ligante possam ser explorados. Já que computar todas as possíveis conformações é demasiado caro para ser considerado, um conjunto de possíveis ligações rotacionáveis que permitirão variações conformacionais deve ser selecionado. Uma possibilidade é gerar certas conformações de um ligante específico e “atracar” cada uma dessas conformações no receptor separadamente (SCHNEIDER, 2006).

- *Flexible-receptor docking* (receptor flexível e ligante flexível). Na maioria dos casos, por simplicidade, é assumido que o receptor é um corpo rígido. Contudo, a maioria das proteínas de interesse farmacêutico são flexíveis a fim de posicionar o ligante em seu sítio de ligação (DHANIK; KAVRAKI, 2001). *Flexible-receptor docking* considera esse caso no qual o receptor é totalmente ou parcialmente flexível. Quanto ao ligante, assim como na abordagem *flexible-ligand docking*, todos os graus de liberdade podem ser explorados. Por ser um problema difícil de tratar devido ao alto número de graus de liberdade a serem considerados, na maioria das vezes permite-se apenas flexibilidade parcial do receptor (DHANIK; KAVRAKI, 2001).

Flexible-ligand docking é a abordagem de atracamento molecular utilizada neste trabalho por ser o tipo de atracamento molecular mais comumente utilizado (DHANIK; KAVRAKI, 2001). O processo de atracamento molecular se inicia com a seleção de duas moléculas interagentes (um ligante e um receptor). A escolha de cada molécula varia de acordo com o objeto de estudo e a aplicação, porém uma alternativa comum é utilizarem-se métodos de *Virtual Screening* (VYAS et al., 2008).

Após a seleção das duas moléculas, é necessário obter as estruturas tridimensionais das mesmas. A busca por arquivos de estruturas tridimensionais de uma molécula receptora pode ser feita a partir de bancos de dados como *Protein Data Bank* (BERMAN et al., 2000) e *ChemBank* (SEILER et al., 2008). A busca por arquivos de estruturas unidimensionais e bidimensionais de uma molécula ligante pode ser feita a partir de bancos de dados específicos para tal, e a transformação de uma e duas dimensões para três dimensões pode ser feita através de aplicações para tal (SCHNEIDER, 2006) (por exemplo OMEGA (HAWKINS et al., 2010)) ; de qualquer maneira, estruturas tridimensionais podem ser encontradas diretamente em bibliotecas como ZINC (IRWIN; SHOICHET, 2005) e *Protein Data Bank*.

Uma molécula pode ser preparada assim que sua estrutura tridimensional é obtida. A preparação das moléculas interagentes é feita como segue (GUEDES; MAGALHÃES; DARDENNE, 2014):

- Preparação do receptor. São feitos ajustes e verificações fundamentais para a correteza da representação da estrutura molecular receptora, tais como:
 1. Verificação de ausência e excesso de átomos;
 2. Verificação de mal posicionamento de cadeias laterais;
 3. Verificação de configurações alternativas para a mesma proteína;
 4. Verificação de conformações não nativas;

5. Verificação da necessidade de estabelecer estado de protonação. Protonação é, resumidamente, a adição de um próton a uma molécula, alterando sua carga. Uma mudança na carga de uma molécula pode implicar em uma diferença significativa no resultado final do atracamento molecular.
- Preparação do ligante. Nessa fase são feitos ajustes fundamentais para a corretude da representação da estrutura da molécula ligante, os quais exercem grande influência no processo de atracamento molecular (BRINK; EXNER, 2009):

1. Verificação de ausência de átomos;
2. Determinação de estado tautomérico. Um composto exhibe tautomerismo se pode ser representado por duas estruturas que são relacionadas por um movimento intramolecular de hidrogênio de um átomo para outro (MARTIN, 2009);
3. Determinação da forma enantiomérica. Enantiômeros são moléculas quirais que são espelhos uma da outra (MCCONATHY; OWENS, 2003), sendo suas composições e funções químicas e físicas idênticas, porém a estereoquímica é diferente. Enantiômeros possuem interações diferentes com outras moléculas quirais;
4. Verificação da necessidade de estabelecer estado de protonação.

É fácil perceber que o subprocesso de preparar uma molécula ligante se diferencia do subprocesso de preparar uma molécula receptora. Além disso, ambos os subprocessos podem ser feitos em paralelo.

A representação correta do receptor e do ligante, bem como dos demais resíduos que porventura façam parte de um atracamento molecular, é um ponto crucial para obter sucesso no resultado do atracamento molecular, e tanto metodologia de busca como função de avaliação dificilmente são capazes de superar ou corrigir eventuais más caracterizações do estado de protonação das moléculas interagentes. (GUEDES; MAGALHÃES; DARDENNE, 2014).

Assim que o ligante estiver preparado, ocorre a identificação de ligações químicas flexíveis, ou seja, é necessário especificar as ligações rotacionáveis na molécula ligante onde podem ocorrer conformações. Cada software de atracamento molecular possui sua própria forma de possibilitar tal ação; em geral, é exibida a molécula ligante ao usuário, que deve selecionar as ligações que podem sofrer torção.

O atracamento molecular entre ligante e receptor muitas vezes necessita da presença de outras moléculas (por exemplo, solvente) para uma representação correta do sistema. Tais moléculas são chamadas de cofatores. Moléculas de água, por exemplo, não somente mediam a ligação entre ligante-receptor, como também auxiliam na função de inserir o ligante mais perto do centro do sítio ativo (GRAAF et al., 2006). Em suma, a adição de cofatores pode alterar significativamente a acurácia do atracamento molecular (ROBERTS; MANCERA, 2008). Caso seja necessária a adição de um solvente, a incorporação do mesmo no sistema é feita a partir de um modelo de solvatação definido:

- Modelo explícito: as moléculas do solvente são definidas explicitamente.
- Modelo implícito: a modelagem explícita de um grande número de moléculas de um solvente associa-se a um grande custo computacional e, além disso, não passam de aproximações.

mações; por estas e outras dificuldades, tornou-se desejável o desenvolvimento de novos métodos que incorporassem implicitamente a presença de um solvente (ROUX; SIMONSON, 1999). Métodos de solvatação implícita podem ser vistos na figura 2.2.

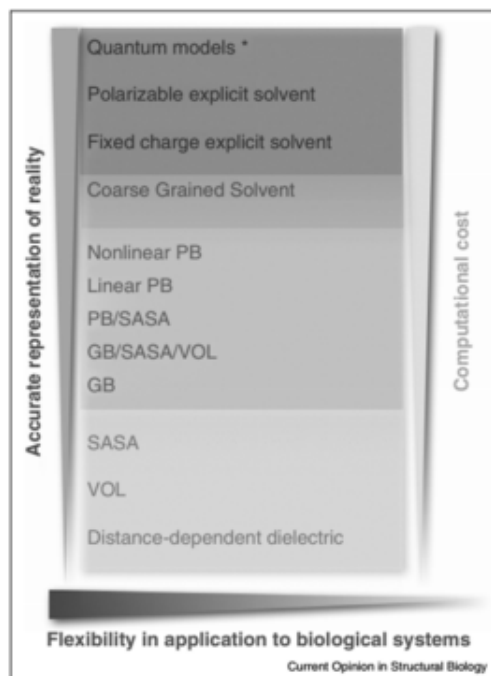


Figura 2.2 – Modelos implícitos de solvatação (KLEINJUNG; FRATERNALI, 2014)

A forma com que os dados são representados no sistema varia de acordo com o software de atracamento molecular. Xu, Xu e Liang (2007) mencionam que uma tentativa precoce de representar os dados buscava projetar as propriedades moleculares em um *grid* 2D (ZIELENKIEWICZ; RABCZENKO, 1985); posteriormente, Jiang e Kim (1991) utilizariam de *grids* em cubo para representar a superfície. Katchalski-Katzir e colaboradores (1992) mostram um notável avanço com o uso de Transformadas Rápidas de Fourier para realizar correlações em busca translacional baseada em *grids*. Além disso, muitas ferramentas possuem suas próprias formas de representar os dados e dessa forma não entram no escopo deste trabalho.

Métodos de busca utilizam de cálculos, torções e variações de conformações a fim de procurar as melhores conformações para um determinado par receptor-ligante (GUEDES; MAGALHÃES; DARDENNE, 2014). Podem ser classificados em:

- Método de busca sistemática: objetiva explorar todos os graus de liberdade do ligante, gerando todas as possíveis conformações para ligação;
- Método de busca determinística: o objetivo é fazer buscas locais baseadas em minimização de energia. Como as buscas são locais, é possível que esse método trave em um mínimo local;

- Método de busca estocástica: movimentos aleatórios são feitos associados a um grau de liberdade. Isso implica que a convergência não é garantida. Algoritmos genéticos são exemplos de busca estocástica.

Após a escolha do método de busca, uma função de avaliação deve ser definida. Esta função será responsável por estabelecer valores às diversas conformações geradas pelo método de busca para, assim, ranqueá-las de acordo com a afinidade de ligação entre o par receptor-ligante. As três principais classes de função de avaliação utilizadas em atracamento molecular receptor-ligante são: baseadas em campos de força, empíricas e baseadas em conhecimento (HUANG; GRINTER; ZOU, 2010):

- Empíricas: estima a afinidade de ligação de acordo com um conjunto de termos de energia ponderado;
- Baseadas em campos de força: são desenvolvidas com base em interações físicas atômicas, incluindo interações de *van der Waals* (VDW), interações eletrostáticas e forças de alongamento/flexão/torção da ligação.
- Baseadas em conhecimento: empregam potenciais de energia que são derivados a partir da informação estrutural incorporado em estruturas atômicas determinados experimentalmente;

É necessário, após todas as tarefas anteriores prontas, definir o sítio de ligação de interesse. A determinação do sítio de ligação pode ser tanto automática (por algum software específico, por exemplo 3DLigandSite (WASS; KELLEY; STERNBERG, 2010)) ou manual.

Com as moléculas preparadas, o algoritmo de busca e função de avaliação definidos, cofatores estabelecidos e sítio de ligação indicado, a execução do algoritmo pode ser realizada. O objetivo é buscar o modo de união ótima dentre todas as conformações e orientações possíveis do par de moléculas. Após a execução, a análise dos resultados é feita, onde é possível verificar a qualidade do atracamento molecular.

3 GERENCIAMENTO DE PROCESSOS DE NEGÓCIO E NOTAÇÃO DE MODELAGEM DE PROCESSOS DE NEGÓCIO

Neste capítulo são apresentados conceitos relacionados ao gerenciamento de processos de negócio (BPM), bem como uma descrição de elementos da notação BPMN, os quais são utilizados ao longo deste trabalho.

3.1 Gerenciamento de Processos de Negócio

BPM contribui para a redução de custos, de tempo e taxas de erro em execuções de processos (DUMAS et al., 2013). Através de desenho de processos e automação, organizações podem documentar, otimizar e controlar melhor seus processos de negócio. BPM tem sido explorado e utilizado em inúmeros domínios de aplicação, incluindo o domínio de negócios onde processos são, em geral, bem conhecidos, estáticos e raramente desviam do fluxo normal, além de domínios como bioinformática onde processos científicos apresentam características dinâmicas e requerem aspectos de flexibilidade e adaptabilidade quando implementados. O ciclo de vida de BPM tipicamente inclui as seguintes fases: (DUMAS et al., 2013):

- *Identificação do processo*: Um problema de negócio é definido, e os processos relativo ao problema de negócio definido são identificados, delimitados e relacionados, gerando uma arquitetura que fornece uma visão geral sobre os processos;
- *Descoberta do processo* (também chamada de modelagem do processo *as-is*): O estado atual dos processos é documentado em sua forma *as-is* a fim de facilitar o entendimento do processo. Entre as principais técnicas para elicitar requisitos a fim de modelar um processo constam: *brainstorming*, análise de documentos, entrevistas etc. A principal notação para modelagem de processos é chamada BPMN, a qual é adotada por diversos softwares de modelagem, tais como Bizagi e Bonita (DUMAS et al., 2013).
- *Análise do processo*: nessa fase, problemas relacionados ao processo *as-is* são identificadas, documentadas e possivelmente quantificadas. Tais problemas resultantes dessa fase são priorizados em termos de impacto e esforço para resolvê-los;
- *Redesenho do processo* (também chamada de melhoria do processo): O objetivo dessa fase é identificar mudanças no processo que auxilia a resolver os problemas identificados na fase anterior. O resultado dessa fase é o processo *to-be*.
- *Implementação do processo*: Nessa fase as mudanças identificadas na fase de redesenho

são preparadas e executadas. O processo então é executado.

- *Controle e monitoramento do processo*: Uma vez que o processo esteja sendo executado, dados relevantes são coletados e analisados a fim de determinar o desempenho do processo em relação às suas métricas e objetivos de performance.

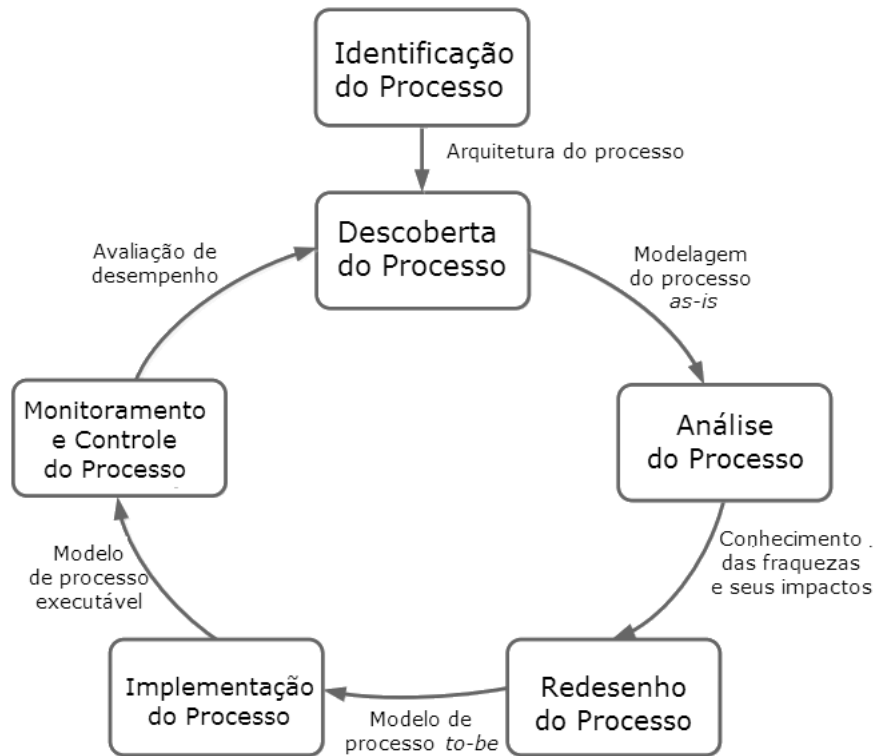


Figura 3.1 – Ciclo de vida de processos (DUMAS et al., 2013)

A modelagem (desenho) de processos de negócio é considerada crucial para o desenvolvimento de projetos (principalmente de software), dado que uma modelagem de processo fraca ou inexistente é uma das maiores causas de fracassos de projetos (BARJIS, 2008). Modelar um processo existente *as-is* pode ser uma tarefa complexa. Além disso, eventuais erros (por exemplo, equívoco por parte do analista de processos) ou problemas estruturais (por exemplo, falta de sincronismo entre elementos de controle de fluxo¹ da modelagem) não tratados nas fases iniciais de seu ciclo de vida podem se propagar para as fases subsequentes. Em métodos tradicionais para realizar a modelagem o analista depende de informações obtidas e organizadas a partir de especialistas no domínio (DUMAS et al., 2013). Uma possível alternativa para minimizar o esforço necessário no desenho de processos é utilizar literatura existente relacionada ao domínio de processo.

¹Na seção 3.2 são descritos elementos de controle de fluxo utilizados no trabalho

3.2 Notação de Modelagem de Processos de Negócio

A notação gráfica de BPMN possibilita a representação dos mais diferentes aspectos de um processo. Seu foco é ser um ponto intermediário de conexão entre o desenho do processo de negócio e a implementação. Em vista disso, é utilizada para uma grande variedade de propósitos de modelagem de processos (MATEJA et al., 2015), incluindo, principalmente, análise de negócios e documentação de processos. Existem diversas ferramentas comerciais que podem ser utilizadas para modelagem de processos (por exemplo *Bizagi*, *Signavio* e *Bonita*).

Os elementos da BPMN utilizados para a modelagem do processo de atracamento molecular são descritos a seguir.

O elemento *tarefa* é utilizado para representar uma ação atômica dentro de um processo. Sua representação é dada pela figura 3.2. Um exemplo de tarefa é “selecionar ligante”.



Figura 3.2 – Representação do elemento Tarefa

Para conexão de elementos dentro de um processo, definindo uma ordem de execução, utiliza-se um *fluxo de sequência*, exibido na figura 3.3. A conexão entre o evento de início de um processo e sua primeira atividade, como “selecionar moléculas interagentes”, é um exemplo de uso do fluxo de sequência.



Figura 3.3 – Representação do elemento Fluxo de sequência

Um *desvio exclusivo divergente* (figura 3.4) é um elemento de controle de fluxo utilizado para criar caminhos alternativos dentro de um processo. A partir desse elemento, apenas um fluxo de execução deve ser seguido. Um *desvio exclusivo convergente* (figura 3.5) é um elemento de controle de fluxo utilizado para sincronizar caminhos alternativos (o primeiro fluxo que chegar no elemento *desvio exclusivo convergente* fará com que o fluxo de execução continue). Um exemplo de desvio exclusivo divergente é a representação da decisão de adicionar ou não adicionar átomos ao ligante. Independentemente da decisão, um desvio exclusivo convergente deve ser utilizado para sincronizar os fluxos de sequência de saída de cada escolha.

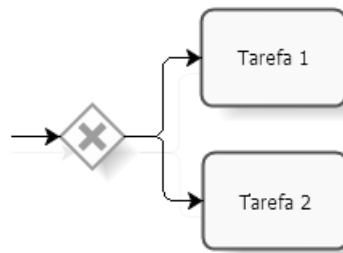


Figura 3.4 – Representação do elemento Desvio exclusivo divergente

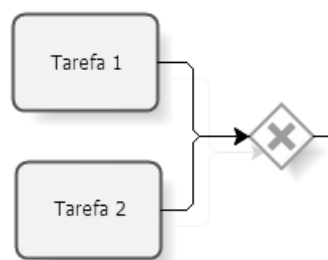


Figura 3.5 – Representação do elemento Desvio exclusivo convergente

Um *desvio paralelo divergente* (figura 3.6) é um elemento de controle de fluxo utilizado para criar vários caminhos de execução paralelos. Um *desvio paralelo convergente* (figura 3.7) é um elemento de controle de fluxo utilizado para sincronizar dois ou mais fluxos paralelos disparados pelo *desvio paralelo divergente*, ou seja, o *desvio paralelo convergente* espera por *fluxos de sequência* de chegada antes de continuar o fluxo através de seus *fluxos de sequência* de saída. Um exemplo de desvio paralelo divergente é a representação das seguintes tarefas que podem ser executadas em paralelo: “selecionar receptor” e “selecionar ligante”. Um desvio paralelo convergente serve como ponto de sincronismo entre os fluxos das duas tarefas.

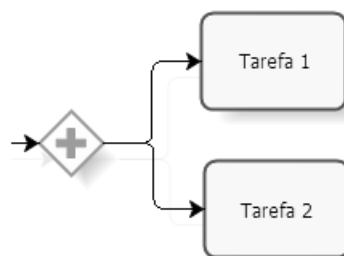


Figura 3.6 – Representação do elemento Desvio paralelo divergente

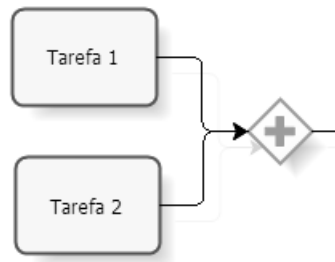


Figura 3.7 – Representação do elemento Desvio paralelo convergente

O *evento de início* (figura 3.8) indica onde um processo em particular começa, contrastando com o *evento de fim* (figura 3.9), que indica onde um processo termina.



Figura 3.8 – Representação do elemento Evento de início



Figura 3.9 – Representação do elemento Evento de fim

Um *subprocesso* (figura 3.10) é uma atividade que pode ser decomposta em um grau de granularidade maior através de um conjunto de subatividades. “Preparar o receptor” é um exemplo de subprocesso, pois inclui subatividades como “verificar necessidade de protonação” e “verificar conformações não nativas”.



Figura 3.10 – Representação do elemento Subprocesso

A *associação* (figura 3.11) é utilizada para ligar informações e artefatos à elementos gráficos da modelagem BPMN.



Figura 3.11 – Representação do elemento Associação

Para exibir informações adicionais ao leitor da modelagem, utiliza-se a *anotação de texto* (figura 3.12). Uma *anotação de texto* é sempre acompanhada de uma associação. Um

exemplo é conectar (utilizando uma associação) uma anotação de texto à tarefa “determinar função de avaliação para o sistema” a fim de informar que a mesma será utilizada para ranquear as conformações geradas na execução do algoritmo.

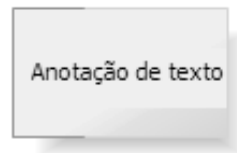


Figura 3.12 – Representação do elemento Anotação de texto

Neste trabalho são aplicadas as duas primeiras fases do ciclo de vida de BPM a fim de obter como resultado o processo *as-is* de atracamento molecular. BPMN é particularmente utilizada para modelar o processo, já que possui elementos de modelagem úteis para a representação de processos e suas particularidades.

4 METODOLOGIA DO TRABALHO

Neste capítulo são apresentadas as técnicas adotadas para o desenvolvimento do trabalho.

Existem vários trabalhos que abordam técnicas para descoberta e mineração de processos. Em Gonçalves, Santoro e Baiao (2009) é explorada a técnica narrativa associada a mineração de texto e interpretação de linguagem natural para geração automática de modelos de processo. No trabalho consta que os participantes do processo descrevem a maneira de agir através de histórias de usuário.

Em Santoro, Borges e Pino (2008) os autores exploram um método para capturar histórias de um determinado grupo (como alternativa para as entrevistas individuais tradicionalmente utilizadas para elicitare processos). Segundo os autores, quando em coletivo, os participantes são capazes de verificar de forma cooperativa os processos descritos pelos outros integrantes por meio de suas perspectivas e adicionar novas histórias com descrições adicionais ou alternativas.

No contexto de extração de modelos a partir da literatura, existem algumas contribuições como o trabalho apresentado por Vakulenko (2015) que fornece uma abordagem inicial ao problema de extração de modelos de processo a partir de linguagem natural. Ainda, em Friedrich, mendling, Puhlmann (2011) os autores apresentam um método automático para gerar modelos em BPMN a partir de textos em linguagem natural.

Os métodos mais utilizados em trabalhos de elicitação de processos de negócio são a entrevista, em geral baseada nas informações dos *stakeholders*, e a análise da literatura existente. Segundo Silva e colaboradores (2012), as informações obtidas a partir de diferentes técnicas de elicitação (por exemplo literatura, *brainstorms* e entrevistas) devem ser agrupadas e organizadas, e informações conflitantes devem ser resolvidas. Essas ações irão, então, refletir no modelo de processo desejado, que representa o conhecimento obtido na elicitação.

O trabalho apresenta a modelagem do processo de atracamento molecular. O modelo gerado pode ser denominado *modelo de referência*. É importante ressaltar que as fases básicas do ciclo de vida de BPM utilizadas no presente trabalho são Identificação e Descoberta do processo, resultando no processo *as-is*. As demais fases do ciclo de vida do processo de atracamento molecular ficam fora do escopo deste trabalho e são sugeridas como trabalho futuro.

A fim de identificar e descobrir o processo de atracamento molecular, os seguintes passos foram seguidos neste trabalho:

1. Identificação das principais funções de negócio do processo de atracamento molecular. Para isso, foram considerados os conceitos de arquitetura de processo propostos por Eid-

Sabbagh, Dijkman e Weske (2012), além da literatura de atracamento molecular. Foram analisados ao todo 37 referências. As principais atividades de atracamento molecular foram identificadas a partir da convergência dos dados entre as referências, ou seja, atividades que eram definidas ou pressupostas como existentes em todas as referências. Atividades específicas de técnicas particulares de atracamento molecular não foram incluídas.

2. Modelagem do processo de atracamento molecular utilizando a notação BPMN e as sete diretrizes de modelagem de processo, incluindo utilização do padrão verbo-objeto para descrever tarefas, eventos de início e fim únicos, evitar o elemento OR, entre outras (MENDLING; REIJERS; AALST, 2010). Tais diretrizes são relevantes para minimizar a probabilidade de erros estruturais no modelo. O primeiro esboço da modelagem foi feito tendo como base a literatura de atracamento molecular. Isso foi feito para reduzir o viés de interpretação que é frequentemente gerado a partir de técnicas de elicitação baseadas em entrevistas com diferentes participantes do processo. Além disso, pessoas entrevistadas nem sempre se sentem confortáveis em falar o que é realmente necessário, ou não o querem.
3. Refinamento do modelo de processo de atracamento molecular. Com o primeiro esboço do modelo do processo de atracamento molecular em mãos, foram conduzidos debates com três especialistas do domínio de bioinformática voltado à atracamento molecular. O objetivo foi refinar e complementar o modelo de processo de atracamento molecular obtido na literatura.

Em seguida foi realizada a validação do modelo de referência do processo de atracamento molecular através de questionários. A validação externa refere-se à adequação e precisão de um modelo correspondente a dados do mundo real (CARLEY, 1996). Cada processo e subprocesso do modelo de referência é examinado por pesquisadores da área de atracamento molecular para detecção de eventuais erros na modelagem.

Após a validação, o modelo é verificado. Existem, na literatura, diversos trabalhos que visam a verificação semântica de modelos em BPMN, incluindo verificações com o uso de YAWL (YE et al., 2008) e Redes de Petri (DIJKMAN; DUMAS; OUYANG, 2008).

O modelo de referência em BPMN é mapeado para uma Rede de Petri, um formalismo matemático forte que possui elementos visualmente similares à BPMN, a qual é verificada a fim de comprovar que o modelo não possui erros semânticos e que possui propriedades como ausência de construções suspeitas, ausência de transições mortas, ausência de transições não vivas e *soundness*. A verificação é feita no software Woflan (AALST, 1999).

Por último, dois softwares de atracamento molecular (DockThor e Autodock Vina) e

seus processos são analisados de maneira comparativa com o modelo de referência gerado. O software Autodesk VINA foi escolhido baseando-se no histórico de uso dos entrevistados no questionário (mais da metade dos entrevistados já utilizaram Autodesk VINA), enquanto o software DockThor foi selecionado por ser amplamente utilizado por pesquisadores brasileiros do domínio de atracamento molecular. *Logs* de eventos de ambos os softwares são gerados manualmente e servem de entrada para o software de mineração de processos ProM (DONGEN et al., 2005). ProM é o mais conhecido e utilizado *framework* de mineração de processo com código aberto (KALENKOVA; LEONI; AALST, 2014). ProM oferece uma ampla variedade de algoritmos de mineração que possibilitam minerar um determinado processo a partir de *logs* de eventos no formato XES (padrão baseado em XML para *logs* de eventos). Nesse contexto, *Heuristic Miner* (AYUTAYA; PALUNGSUNTIKUL; PREMCHAIWADI, 2012) é um algoritmo de mineração de processos utilizado para expressar o comportamento principal gravado num *log* de eventos, possibilitando ainda a geração de um modelo de processo baseado no *log*. O algoritmo *Heuristic Miner* é encontrado no *framework* ProM na forma de um *plugin* que possibilita gerar tal modelo em BPMN. O *plugin* é chamado de *BPMN analysis using heuristic miner*.

A metodologia utilizada neste trabalho é encorajada para servir de referência a outros trabalhos similares na área de BPM (desde a modelagem de processo até validação externa, verificação e análise de conformidade).

5 MODELAGEM DO PROCESSO DE ATRACAMENTO MOLECULAR EM BPMN

Esse capítulo apresenta uma representação sistemática do processo de atracamento molecular. O processo é modelado usando elementos da BPMN 2.0 (ALLWEYER, 2010) e o software de modelagem Bizagi versão 2.9 (escolhido por preferência do autor). As figuras subsequentes, 5.1 à 5.8 mostram os subprocessos inerentes ao processo geral de atracamento molecular e devem ser lidas conforme notação já introduzida na seção 3.2.

5.1 Modelos de Referência

Segundo O’Leary e Richardson (2012), modelos de referência de processos proporcionam uma solução genérica, servindo como um molde para definir um modelo em um ambiente específico. Além disso, utilizam de vocabulário do domínio, bem como suas restrições. Modelos de referência, ao capturarem o conhecimento de um determinado domínio, proporcionam reutilização de conhecimento, o que também influencia na redução de custos de um projeto (WINTER; SCHELP, 2006).

5.2 Modelo de Referência do Processo de Atracamento Molecular

Os modelos exibidos nas figuras são os modelos resultantes das fases de identificação e descoberta do processo de atracamento molecular, incluindo refinamento pelos três especialistas do domínio. O refinamento foi feito a partir de *brainstorms*. Esses modelos são utilizados no questionário subsequente para que sejam validados por um grupo maior de pesquisadores.

O processo de atracamento molecular (figura 5.1) começa com o subprocesso a) *selecionar moléculas interagentes*. Após, três tarefas são disparadas em paralelo: b) *obter estrutura tridimensional do receptor*, c) *obter estrutura tridimensional do ligante*, e d) *determinar forma de representação dos dados*. Quando a tarefa b termina a execução, o subprocesso e) *preparar o receptor* executa. Quando a tarefa c completa a execução, o subprocesso f) *preparar o ligante* está apto a executar. Tendo os dois subprocessos executados (e e f) a tarefa g) *decidir entre adicionar ou não cofatores* pode ser executada. Note que a tarefa f é seguida de um desvio exclusivo relacionado à decisão de adição de cofatores. Se cofatores são necessários, o subprocesso h) *adicionar cofatores* é disparado. Caso contrário, o fluxo continua e espera-se que os subprocessos i) *determinar método de busca para o sistema* e j) *determinar função de avaliação*

para o sistema terminem. Após seus términos, o subprocesso k) *determinar sítio de ligação* está apto a executar. Em seguida, o algoritmo é executado disparando a tarefa l) *executar algoritmo*. O processo termina com a avaliação dos resultados na tarefa m) *avaliar resultados da execução*.

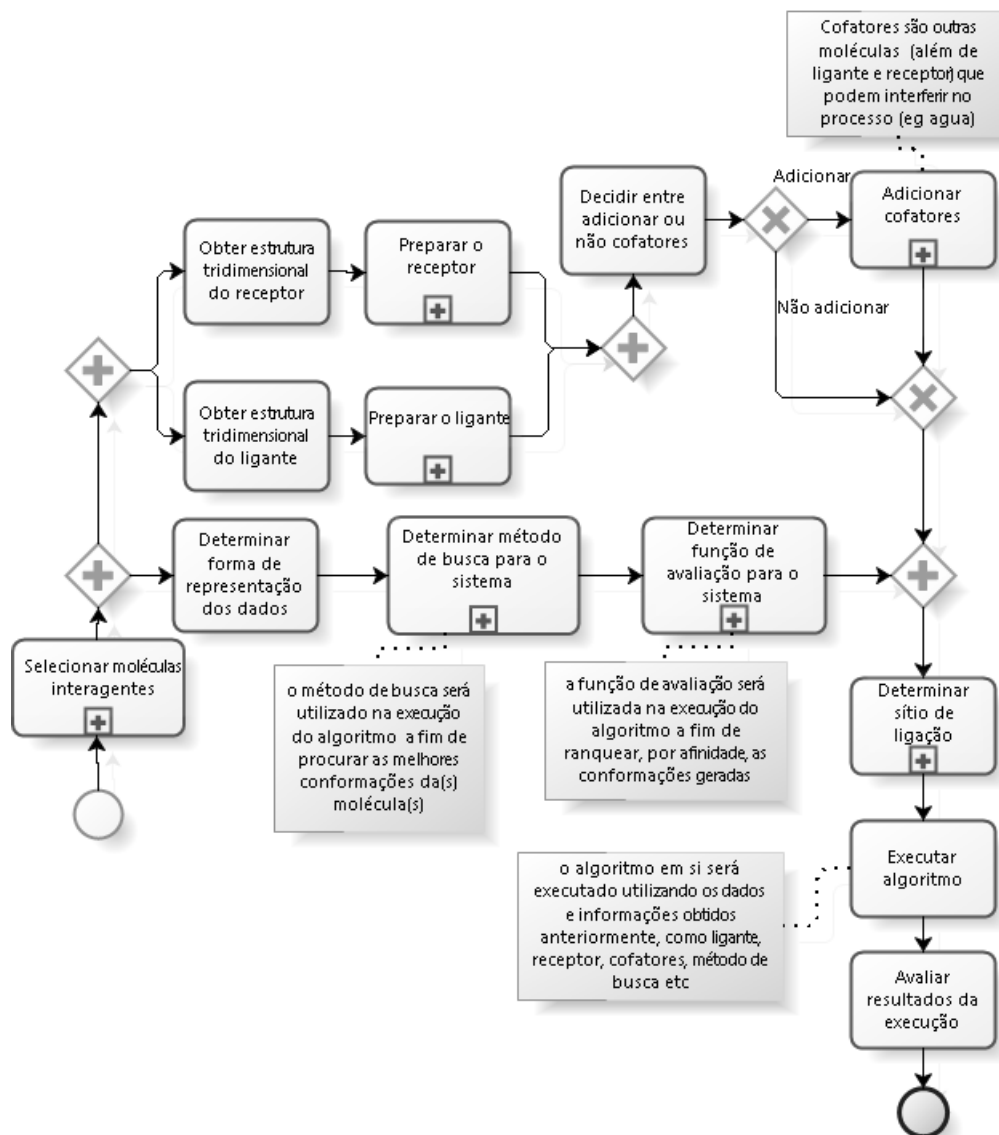


Figura 5.1 – Modelo de referência - Processo principal de atracamento molecular

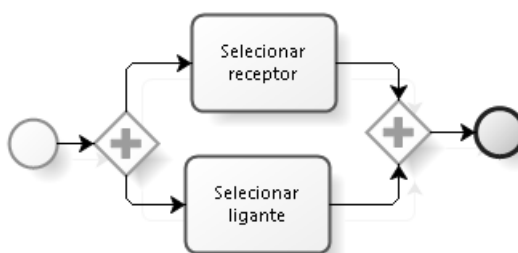


Figura 5.2 – Modelo de referência - Subprocesso Selecionar Moléculas Interagentes

O subprocesso selecionar moléculas interagentes (figura 5.2) é um subprocesso simples que dispara duas tarefas paralelas: *selecionar ligante* e *selecionar receptor*. O subprocesso termina com um desvio exclusivo convergente.

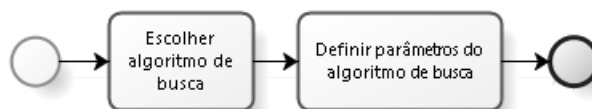


Figura 5.3 – Modelo de referência - Subprocesso Determinar método de busca para o sistema

O subprocesso determinar método de busca para o sistema (figura 5.3) começa com a tarefa *escolher algoritmo de busca*, seguida da tarefa *definir parâmetros do algoritmo de busca*. Logo após, o subprocesso é finalizado.

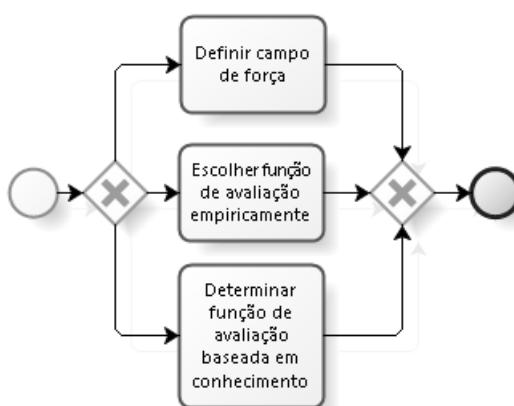


Figura 5.4 – Modelo de referência - Subprocesso Determinar função de avaliação para o sistema

O subprocesso *determinar função de avaliação para o sistema* (figura 5.4) é executado após a escolha de um método de busca. Tal função será responsável por atribuir valores às variadas conformações geradas pelo método de busca, para que assim tais conformações possam ser ranqueadas de acordo com a afinidade de ligação entre receptor e ligante. O cenário é representado com uma decisão onde apenas uma das seguinte tarefas será executada: *determinar campo de força*, *escolher função de avaliação empiricamente* e *determinar função de avaliação baseada em conhecimento*.

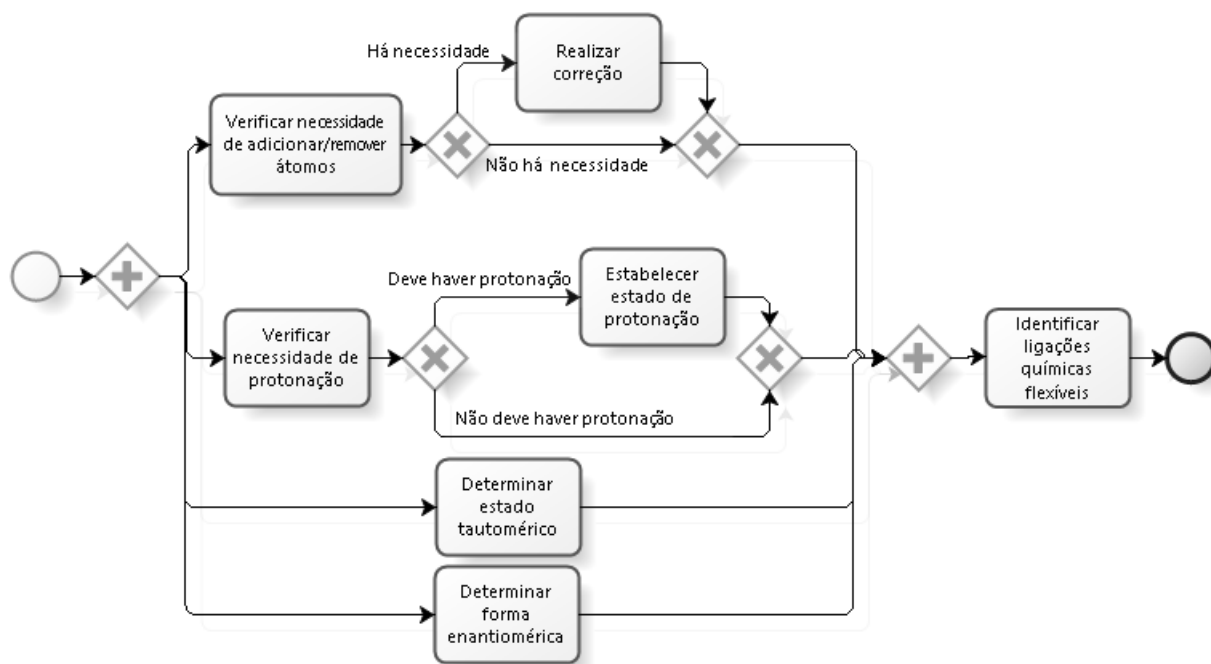


Figura 5.5 – Modelo de referência - Subprocesso Preparar Ligante

O subprocesso de preparação do ligante (figura 5.5) começa com a execução paralela das tarefas a) *verificar necessidade de adicionar/remover átomos*, b) *verificar necessidade de protonação*, c) *determinar estado tautomérico* e d) *definir forma enantiomérica*. Após a execução da tarefa a), o processo pode tanto seguir para a tarefa de *realizar correção* quanto continuar, onde será sincronizado no desvio paralelo convergente com o restante dos fluxos das tarefas paralelas. Do mesmo modo, no caso da tarefa b), se for necessária protonação, o fluxo continua a fim de executar a tarefa *estabelecer estado de protonação*. Ao término da execução dos fluxos paralelos, o processo finaliza.

O subprocesso *preparar o receptor* (figura 5.6) dispara uma série de tarefas que podem ser executadas em paralelo. Foi utilizado um desvio paralelo divergente (*AND-Split*) para representar a execução paralela das tarefas a) *verificar necessidade de adicionar/remover átomos/resíduos*, b) *verificar mal posicionamento de cadeias laterais*, c) *verificar presença de configurações alternativas para a mesma proteína*, d) *verificar conformações não-nativas* e e) *verificar necessidade de protonação*. Se a tarefa a) resulta em átomos/resíduos necessários, então são realizadas correções. Caso contrário, o processo finaliza. O mesmo pode ser aplicado no caso da tarefa b) resultar em cadeias laterais mal posicionadas. Se a tarefa c) resultar em uma configuração alternativa para a mesma proteína, então a tarefa *escolher configuração a ser usada* deve ser executada. Caso contrário, nada é feito. Se na tarefa d) for identificada a presença de conformações não-nativas, a tarefa *Realizar correção* é disparada. Finalmente, a

tarefa *verificar necessidade de protonação* é executada. A tarefa pode resultar em necessidade de protonar o receptor; neste caso, a tarefa *estabelecer estado de protonação* é executada.

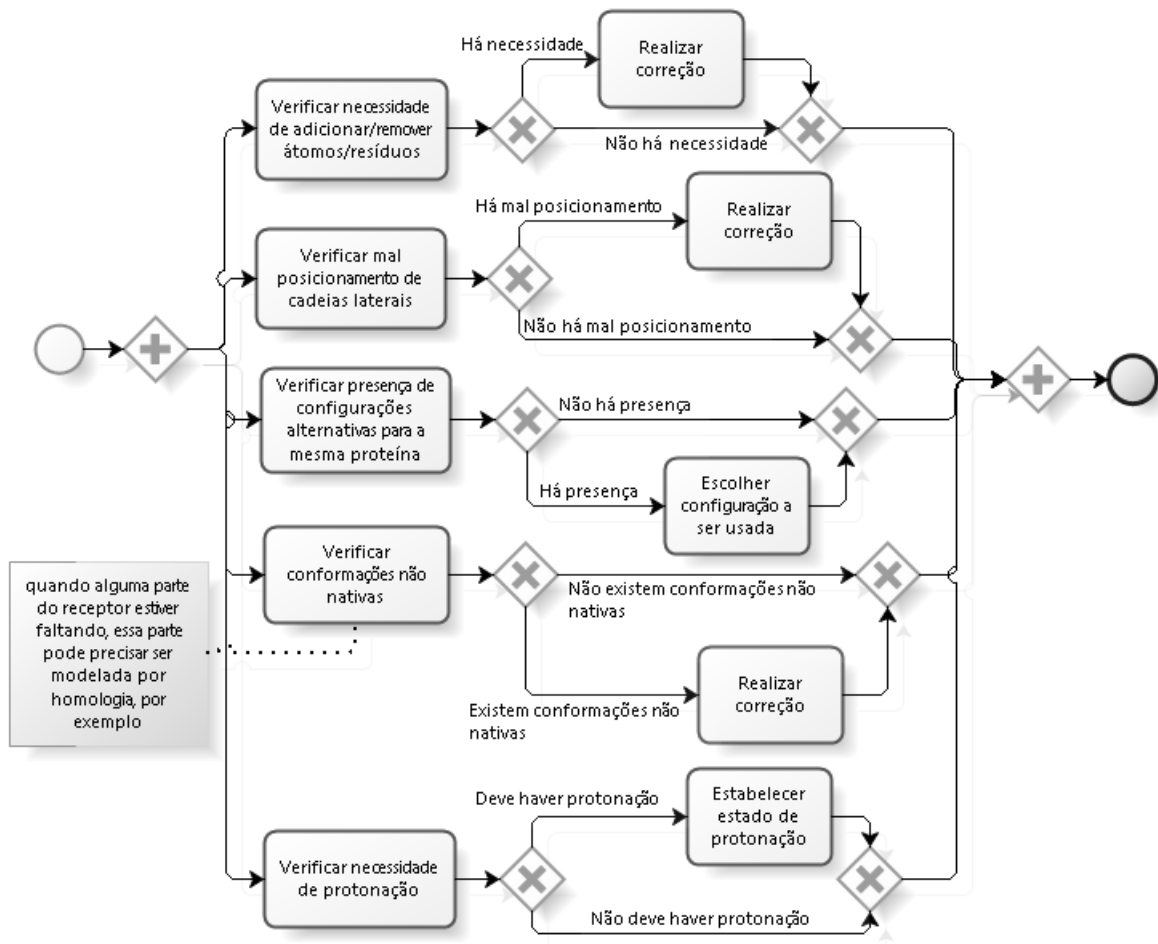


Figura 5.6 – Modelo de referência - Subprocesso Preparar Receptor

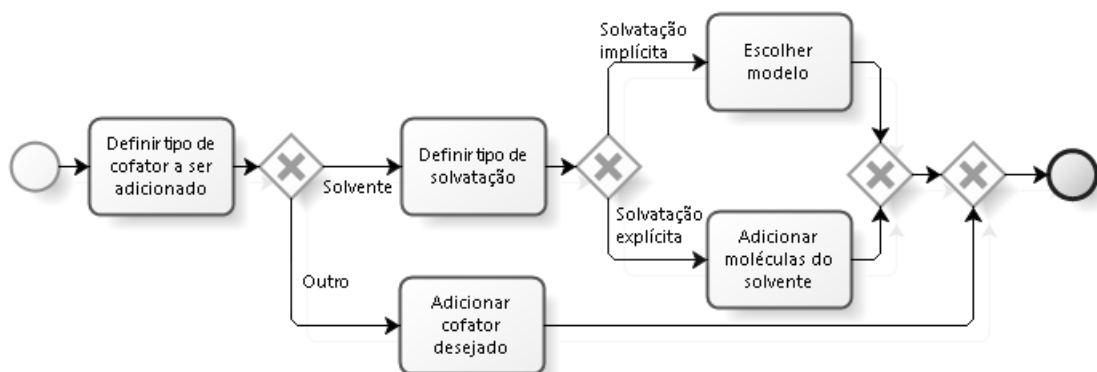


Figura 5.7 – Modelo de referência - Subprocesso adicionar cofatores

O subprocesso *adicionar cofatores* (figura 5.7) começa com a tarefa *definir tipo de cofator a ser adicionado*. No caso que o cofator será um solvente, é necessário decidir, através da tarefa *definir tipo de solvatação*, se o solvente é implícito ou explícito. No caso de solvatação implícita, a tarefa *escolher modelo* é executada e o subprocesso finaliza. No caso de solvatação explícita, a tarefa *adicionar moléculas do solvente* é executada e o subprocesso finaliza. Caso o cofator não seja um solvente, a tarefa *adicionar cofator desejado* é disparada e o processo finaliza.

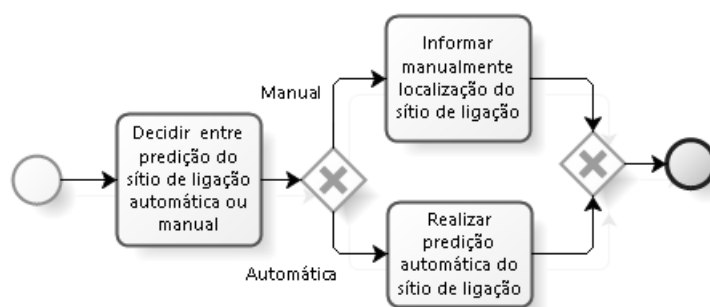


Figura 5.8 – Modelo de referência - Subprocesso Determinar sítio de ligação

O subprocesso *determinar sítio de ligação* (figura 5.8) começa com a tarefa *decidir entre predição do sítio de ligação automática ou manual*. O resultado dessa tarefa será uma decisão entre manual ou automática. No caso de manual, a tarefa *informar manualmente localização do sítio de ligação*. Caso contrário (predição automática) a tarefa *realizar predição automática do sítio de ligação* é eleita para executar.

6 VALIDAÇÃO DA MODELAGEM DO PROCESSO DE ATRACAMENTO MOLECULAR

Para validar a correção do processo de atracamento molecular, foram conduzidos questionários com pesquisadores relacionados à área de atracamento molecular. A validação é utilizada como um meio para comparar o processo apresentado no modelo de referência de atracamento molecular com o processo existente no mundo real.

Os questionários foram utilizados como uma forma experimental de validação da modelagem por pesquisadores da área de atracamento molecular, de modo a minimizar possíveis vieses de interpretação do autor resultante da coleta de informações da literatura, assim como para obter parâmetros relativos à importância da modelagem para a área. Os dois questionários foram feitos via Google forms e respondidos por 9 pesquisadores da área de biologia molecular (especificamente com envolvimento em atracamento molecular), sendo três pesquisadores com conhecimento avançado em atracamento molecular, três pesquisadores com conhecimento intermediário e três pesquisadores com conhecimento básico (sem grande experiência prática, porém breve conhecimento teórico).

Primeiramente, o Questionário 1 (apêndice D) foi liberado para os participantes. Após o término do Questionário 1 pelos participantes, o acesso ao material de tutorial sobre BPMN (apêndice C) foi disponibilizado. Em seguida (após a leitura do tutorial pelos pesquisados), ocorreu a liberação do Questionário 2 (vide apêndice E). O Questionário 1 foi respondido sem consulta a nenhum tipo de fonte, enquanto o questionário 2 foi respondido com auxílio do tutorial de BPMN. Não foram permitidas outras fontes de consultas além das disponibilizadas.

O tempo para resolver os questionários não foi definido pois não era o foco da avaliação, porém recomendou-se a reserva de um tempo de aproximadamente 30 minutos para resolver cada questionário e cerca de 10 minutos para o tutorial. Além disso, os questionários foram anônimos, ou seja, as informações pessoais solicitadas foram usadas apenas para dados estatísticos neste trabalho.

Dentre os pesquisados, 5 são alunos com superior incompleto ou completo, 2 possuem mestrado e 2 possuem doutorado. Além disso, 7 dos 9 pesquisados utilizam ou já utilizaram softwares Autodock (Autodock 4, Autodock Vina e Autodock Tools). Dentre os pesquisados, 2 não relataram o uso de quaisquer softwares (conhecimento teórico apenas) e o software DockThor é utilizado por 1 dos pesquisados. O uso de um software próprio de atracamento foi reportado por 1 pesquisado; o nome desse software não foi divulgado no questionário.

Tabela 6.1 – Validação da modelagem

	1	2	3	4	5	6	7	8	9
Processo principal	Não	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Selecionar moléculas interagentes	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Preparar o ligante	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Preparar o receptor	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Adicionar cofatores	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Determinar mét. de busca para o sistema	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Determinar fun. de aval. para o sistema	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim
Determinar sitio de ligação	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim

Para cada uma das modelagens, o pesquisado deveria responder se a modelagem estava correta e, em caso negativo, relatar o porquê. A tabela 6.1 exhibe cada um dos processos e sub-processos, bem como o parecer de cada um dos pesquisados, onde consta *Sim* para modelagem correta e *Não* para modelagem incorreta, sendo que cada coluna representa um pesquisado.

A resposta dos pesquisados foi bastante positiva: 6 dentre as 8 modelagens do processo de atracamento molecular foram avaliadas por todos os pesquisadores como corretas, ou seja, todos os pesquisadores concordaram com o modelo. Nas 2 modelagens restantes, 8 pesquisadores avaliaram como corretas, enquanto 1 avaliou como incorreta. Portanto, 70 das 72 avaliações foram positivas (modelagem correta) e apenas 2 negativas (modelagem incorreta), sendo que os comentários feitos pelos pesquisadores para modelagem incorreta serão comentados a seguir.

Um dos pesquisados (nr. 1) reportou a modelagem do processo principal como incorreta. Segundo ele, “*Determinar forma de representação dos dados*” deve ser feita antes de preparar o receptor e o ligante e não em paralelo, pois dependendo da forma dos dados isso pode influenciar diretamente em como preparar o ambiente. “*Determinar função de avaliação do sistema*” poderia ser feito após definir todo o ambiente, permitindo assim adicionar novos pesos baseados nos “cofatores” e no ambiente como um todo. O mesmo vale para “*Determinar um método de busca para o sistema*”.

Explicação: Primeiramente, a preparação de receptor e ligante pode ser feita em paralelo com o subprocesso *Determinar forma de representação dos dados*, pois a preparação das moléculas não requer que os dados sejam representados (seja em forma de *grid*, seja em outra forma). Em segundo lugar, a determinação da função de avaliação pode levar em conta cofatores mesmo que não se conheçam os cofatores que serão adicionados, e isso se aplica também ao método de busca, ou seja, os subprocessos *Determinar função de avaliação para o sistema*

e *Determinar método de busca para o sistema* podem ser executados sem que os componentes do sistema sejam conhecidos. Por exemplo, ao adicionarmos uma variável na função de busca que valora um dado cofator dependendo do método de solvatação, é normal que não se saiba *a priori* qual método de solvatação será usado. O método de solvatação utilizado será descoberto quando o algoritmo entrar em execução e a função for avaliada. O mesmo acontece com o método de busca: o algoritmo de busca e seus parâmetros devem ser genéricos o suficiente para levar em conta quaisquer moléculas que sejam adicionadas no sistema.

Outro pesquisado (nr. 8) reportou a modelagem do subprocesso de “Determinar função de avaliação do sistema” como incorreta. Segundo o próprio pesquisado, *fiquei em dúvida se todas as etapas mostradas são exclusivas*.

Explicação: As tarefas são exclusivas. Existem diversas evidências na literatura que confirmam as diferentes classes das funções de avaliação, incluindo o estudo de Cornell (2006). Cada uma das classes são derivadas de determinados elementos: funções baseadas em campo de força são baseadas em campos de força de mecânica molecular padrões; funções empíricas incluem termos ponderados para reproduzir conjuntos de energia de ligação experimentais; funções baseadas em conhecimento são derivadas a partir de dados estatísticos de interações atômicas. Contudo, apesar da tarefa ser exclusiva, nada impede que a função de avaliação seja híbrida. Como relatado nos trabalhos futuros, cada tarefa (por exemplo “Escolher função de avaliação empiricamente”) de um dado subprocesso (por exemplo Determinar função de avaliação) pode ser granularizada em BPMN a ponto de ser um outro subprocesso onde constariam várias outras tarefas (por exemplo “Extender a função de avaliação”) a fim de gerar resultados diferentes como, no caso, funções de avaliação híbridas.

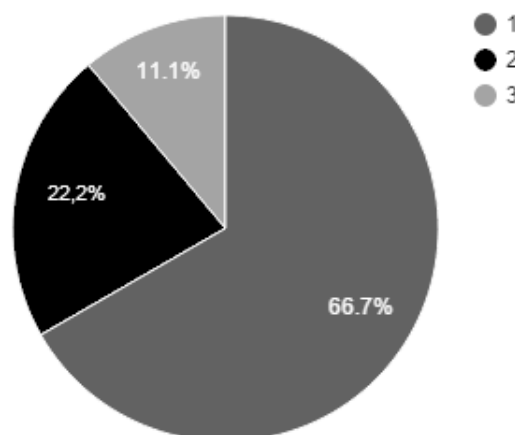


Figura 6.1 – Compreensibilidade dos modelos

Também foi questionada a compreensão da modelagem através do seguinte questionamento: Qual sua impressão geral sobre todas as modelagens em relação à compreensibilidade? Os votos foram classificados em uma escala Likert (1 a 5), sendo que o valor 1 indica fácil compreensão e o valor 5 difícil compreensão (figura 6.1). Percebe-se que os votos ficaram concentrados entre 1 e 3, sendo em sua maioria 1, o que comprova a compreensibilidade do modelo de referência segundo opinião dos pesquisados.

7 VERIFICAÇÃO DA MODELAGEM

A notação BPMN, apesar de suas características favoráveis, como fácil entendimento e capacidade de representação de processos complexos, é ambígua em suas construções e deficiente em semântica formal, o que pode levar os modelos a exibirem estados indesejáveis como *deadlocks* e *livelocks*¹, além de inconsistência, ambiguidade e incompletude (NUFFEL; MULDER; KERVEL, 2009). Para suprimir esses potenciais problemas, é feita uma verificação na modelagem seguindo uma abordagem relatada por Dijkman, Dumas e Ouyang (2008), onde utiliza-se um mapeamento de BPMN para Redes de Petri. Com tal mapeamento é gerado um modelo na forma de Rede de Petri, o qual é verificado com um software específico a fim de analisar determinadas propriedades semânticas como, por exemplo, a ausência de transições mortas.

7.1 Redes de Petri

Redes de Petri (PNs) são uma técnica de especificação formal gráfica e matemática, utilizadas para modelagem de sistemas com componentes que interagem entre si e que exibem características de um sistema orientado a eventos (concorrência, operações assíncronas, *deadlocks*, etc.) (SEATZU; SILVA; SCHUPPEN, 2013).

Uma PN é um grafo direcionado composto por dois tipos de nodos: círculos (chamados de lugares) e barras (chamadas de transições), os quais são interconectados por arestas direcionadas, de lugares à transições ou vice-versa (PETERSON, 1977). Ainda em Peterson (1997), se uma aresta é direcionada do nodo I ao nodo J, por exemplo, então diz-se que I é uma entrada de J, e J é uma saída de I. Um exemplo de Rede de Petri é exibido na figura 7.1.

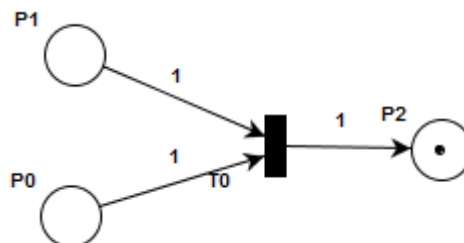


Figura 7.1 – Exemplo de Rede de Petri

¹Deadlock e Livelock são situações de bloqueio no fluxo do processo, relacionados à má utilização dos elementos de desvio. Em caso de ocorrência de tais situações, o processo pode não finalizar ou conduzir a estados incorretos

O número inteiro de lugares e o número inteiro de transições em um grafo PN são ambos finitos e diferentes de zero. Dados i e j números inteiros, sendo i o número de transições e j o número de lugares em uma PN, o grupo de transições de uma PN é dado por $T = \{t_0, t_1, t_2, \dots, t_{i-1}\}$, e o grupo de lugares, por $P = \{p_0, p_1, p_2, \dots, p_{j-1}\}$.

Cada lugar contém um número inteiro e finito de marcas. A marcação de uma PN é representada graficamente por um ponto preto dentro de um círculo (lugar). A marcação, assim, define o estado de uma PN. A marcação de uma PN é dada pelo vetor $M = \{m_0, m_1, \dots, m_{j-1}\}$, onde as posições dentro do vetor representam o número de marcas em cada um dos lugares.

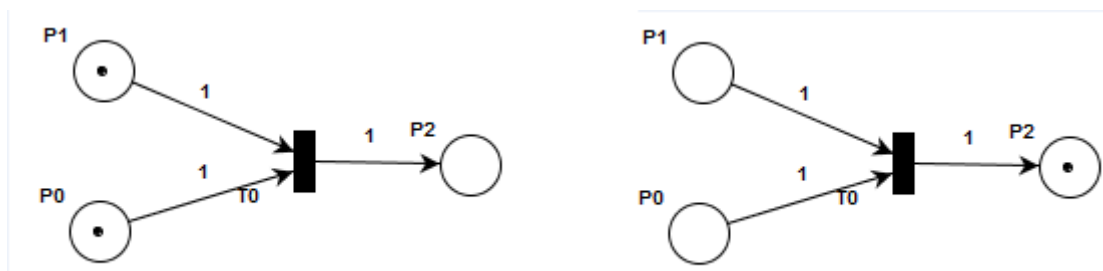


Figura 7.2 – Exemplo de disparo de transição

O estado de uma PN pode ser alterado devido a disparos de transições. Uma transição pode ser disparada se em todo lugar de entrada da transição existe ao menos uma marca. Portanto, disparar uma transição t_0 consiste em consumir uma marca em cada um dos lugares de entrada de t_0 e colocar uma marca em cada um dos lugares de saída de t_0 (DAVID; ALLA, 2010). Um exemplo é visto na figura 7.2, onde à esquerda é mostrada a PN antes do disparo da transição t_0 e, à direita, PN resultante do disparo de t_0 .

Transições cujo disparo não pode ser observado são chamadas transições silenciosas. Uma transição silenciosa é rotulada com um caracter de vazio.

7.2 Mapeando BPMN para Redes de Petri

Como mencionado no início deste capítulo, a abordagem seguida neste trabalho é a descrita e utilizada em Dijkman, Dumas e Ouyang (2008). Primeiro são exibidos os mapeamentos utilizados; logo após, o modelo de referência BPMN é mapeado para Redes de Petri e, por fim, o modelo de referência em Redes de Petri é verificado.

Segundo Dijkman, Dumas e Ouyang (2008) um processo de BPMN é bem formado se satisfaz os seguintes requisitos:

1. Um evento de início ou um evento de exceção possui apenas um fluxo de sequência de saída e não possui nenhum fluxo de sequência de chegada;
2. Um evento de fim possui apenas um fluxo de sequência de chegada e não possui nenhum fluxo de sequência de saída;
3. Atividades e eventos intermediários possuem apenas um fluxo de sequência de chegada e apenas um fluxo de sequência de saída;
4. Desvios AND-Split e XOR-Split possuem apenas um fluxo de sequência de chegada e mais de um fluxo de saída;
5. Desvios AND-Join e XOR-Join possuem vários fluxos de sequência de chegada, porém apenas um fluxo de sequência de saída.

O processo de atracamento molecular modelado - modelo de referência - assim como cada um de seus subprocessos satisfaz todos os requisitos acima.

7.2.1 Mapeamento de tarefas, eventos e desvios

O mapeamento de tarefas, eventos e desvios de BPMN para módulos de Redes de Petri usado em (DIJKMAN; DUMAS; OUYANG, 2008) é exibida na figura 7.3.

- Tarefa: mapeada para uma transição contendo um lugar de entrada e um de saída.
- Eventos de início e fim: mapeados para módulos similares, porém uma transição silenciosa é utilizada para denotar o início ou fim do processo.
- Desvios: mapeados para módulos com transições silenciosas, capturando o comportamento de roteamento.

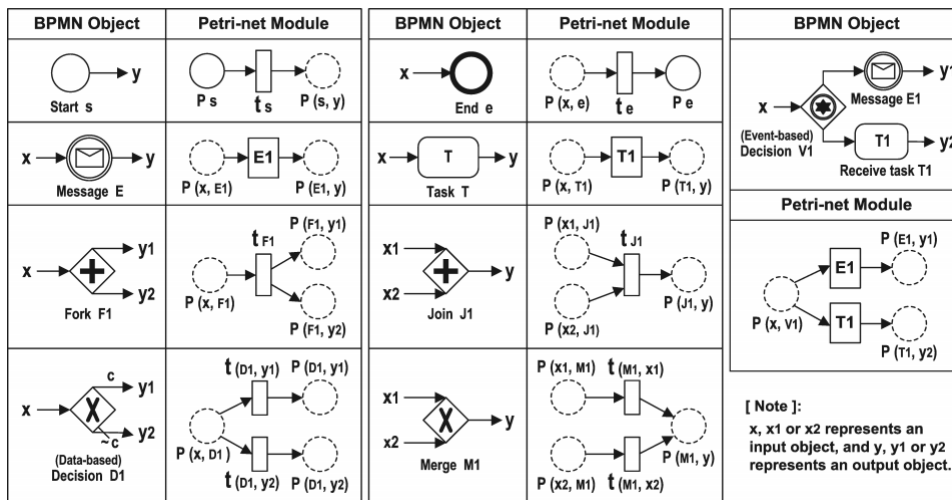


Figura 7.3 – Mapeamento dos elementos tarefa, eventos e desvios de BPMN para Redes de Petri (DIJKMAN; DUMAS; OUYANG, 2008)

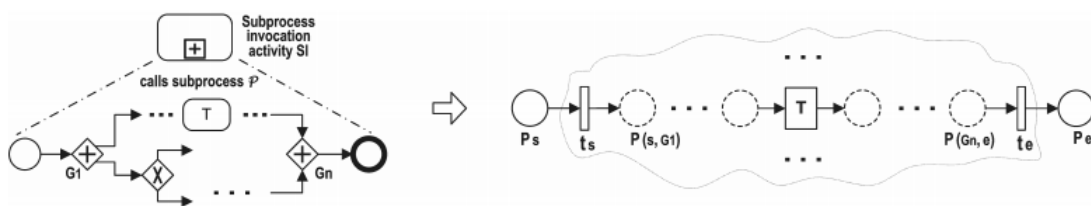


Figura 7.4 – Mapeamento de um subprocesso para Redes de Petri (DIJKMAN; DUMAS; OUYANG, 2008)



Figura 7.5 – Chamada de subprocesso através de uma atividade SI (DIJKMAN; DUMAS; OUYANG, 2008)

7.2.2 Mapeamento de subprocessos

Subprocessos são tratados como processos individuais e, portanto, seguem o mesma abordagem de mapeamento. O mapeamento de um subprocesso é exibido na figura 7.4.

A chamada de um subprocesso também é mapeada por uma tarefa de invocação de subprocesso (SI). Dois lugares (neste caso, desenhados como um círculo de bordas pontilhadas) representam os fluxos de entrada e saída de uma atividade SI. Duas transições novas: uma representando a invocação do subprocesso e outra representando o fluxo de retorno do subprocesso (após seu término) para o processo principal. A figura 7.5 exhibe a chamada de um subprocesso através de uma atividade SI.

A modelagem do processo principal de atracamento molecular em Redes de Petri, bem como de seus subprocessos, foram representadas separadamente a fim de facilitar a visualização. Um modelo único pode ser gerado unindo os subprocessos ao processo principal através de invocação de subprocessos, ou seja, uma dada transição que representa um subprocesso (por exemplo, de acordo com o mapeamento, a transição T41 na figura 7.6 representa o subprocesso “selecionar moléculas interagentes”) pode ser substituída por duas transições t_s e t_e (como exibido na figura 7.4), onde a primeira transição é utilizada para invocação de subprocesso e, a segunda, para o fluxo de retorno do subprocesso. Assim, o subprocesso pode ser inserido entre as duas transições, sendo que os eventos de início e fim do subprocesso devem ser ligado às transições de invocação de subprocesso e de fluxo de retorno do subprocesso, respectivamente.

O mapeamento dos demais elementos, como desvio OR-Split, eventos intermediários e fluxo de troca de mensagens, não serão descritos neste trabalho pois não fazem parte da modelagem do processo principal e dos respectivos subprocessos.

7.2.3 Marcação inicial

Em relação à marcação, o estado de início da modelagem em BPMN será representado por uma marca inicial no modelo de Rede de Petri correspondente.

7.3 Modelagem do processo usando Redes de Petri

Nesta seção são exibidas as modelagens em Redes de Petri relativas ao modelo de referência de atracamento molecular apresentado no capítulo 5, incluindo cada um de seus subpro-

cessos.

Aplicando-se o mapeamento descrito ao longo da seção 7.2, obtêm-se os modelos (figuras 7.6 à 7.13) no formato de Redes de Petri (desenhados no software PIPE (DINGLE; KNOTTENBELT; SUTO, 2009), o qual é gratuito e oferece uma interface amigável).

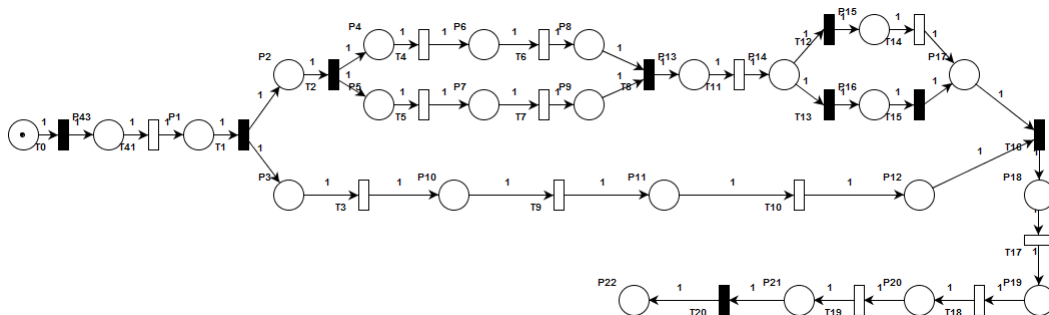


Figura 7.6 – Processo principal, em Redes de Petri

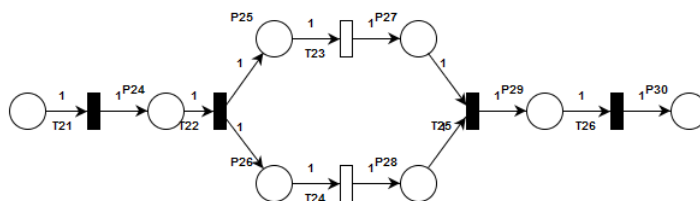


Figura 7.7 – Subprocesso Selecionar Moléculas Interagentes, em Redes de Petri

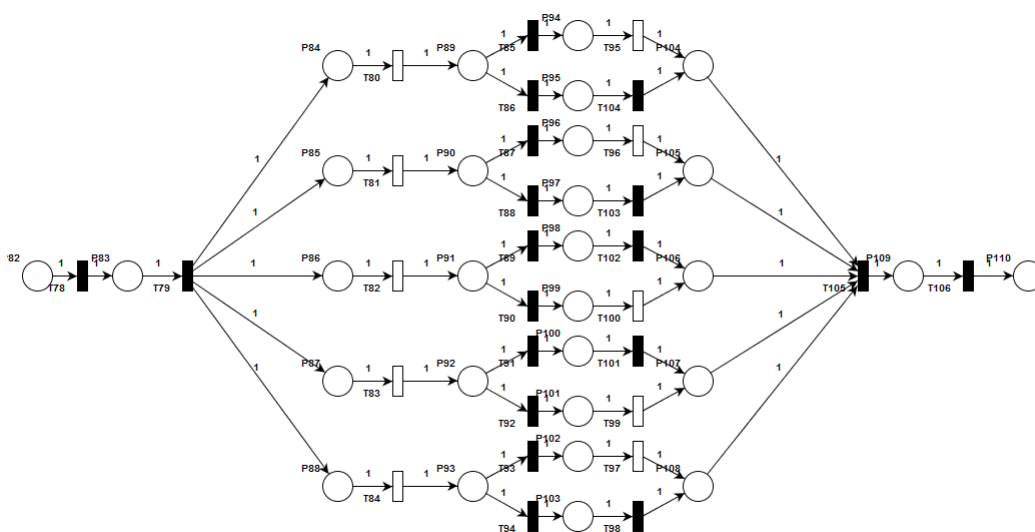


Figura 7.8 – Subprocesso Preparar o receptor, em Redes de Petri

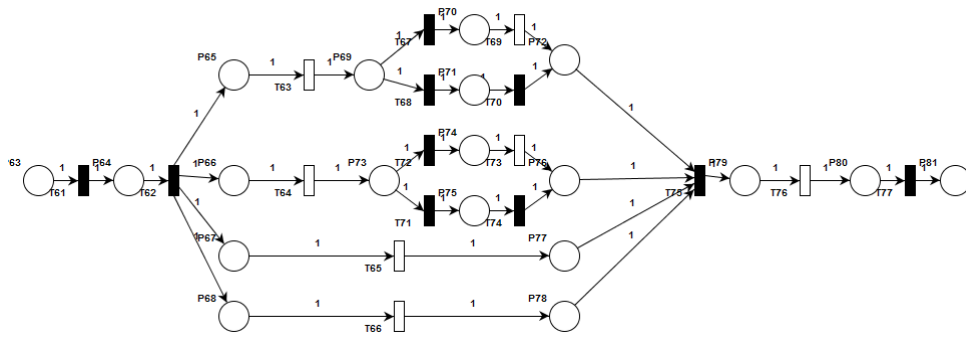


Figura 7.9 – Subprocesso Preparar o ligante, em Redes de Petri

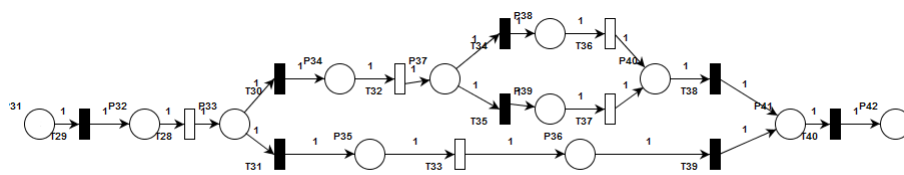


Figura 7.10 – Subprocesso Adicionar cofatores, em Redes de Petri

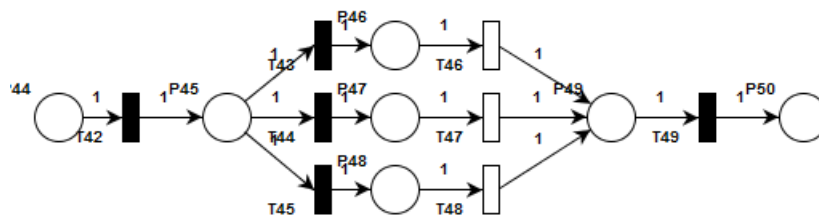


Figura 7.11 – Subprocesso Determinar função de avaliação para o sistema, em Redes de Petri

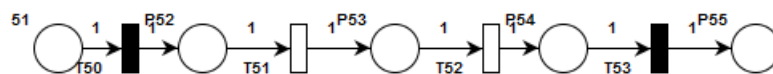


Figura 7.12 – Subprocesso Determinar método de busca para o sistema, em Redes de Petri

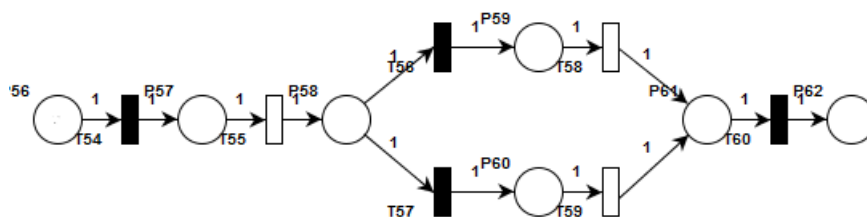


Figura 7.13 – Subprocesso Determinar sítio de ligação, em Redes de Petri

É possível perceber que a modelagem do processo de atracamento molecular em Redes de Petri é bastante similar, estruturalmente, à modelagem em BPMN apresentada no capítulo

5. Além disso, o mapeamento de BPMN para Redes de Petri é intuitivo e sem ambiguidades. Os modelos gerados em Redes de Petri permitem uma verificação semântica do modelo de referência do processo de atracamento molecular, a qual será descrita no próximo capítulo.

7.4 Verificação das Redes de Petri

A representação gráfica de Redes de Petri e suas técnicas existentes de análise são particularmente úteis para geração de informação de diagnósticos informativos. A fim de verificar a corretude do processo de negócio modelado, utiliza-se neste trabalho o software Woflan, baseado em técnicas de análise de Redes de Petri. O software foi escolhido por possuir uma vasta documentação e prover diagnósticos de análise de alta qualidade, além de ser gratuito. Este software permite a análise de certas propriedades (citadas a seguir) do processo de atracamento molecular a partir do modelo em Redes de Petri e, conseqüentemente, em BPMN. A descrição do software Woflan, bem como de seus métodos de análise das propriedades de Redes de Petri é aprofundado por Verbeek e colaboradores (2001). O software realiza a análise de propriedades das Redes de Petri, incluindo:

1. Verificação sintática (tarefas sem condição de entrada ou saída);
2. Detecção de construções suspeitas, por exemplo, violação da propriedade de escolha livre ou XOR-Splits não seguidos por XOR-Joins (mal estruturado);
3. Detecção de transições mortas e transições não-vivas;
4. *Soundness*. Aalst (1997) define *soundness* como: “Para qualquer caso, o processo irá eventualmente terminar, e no momento que o processo terminar existirá uma marca em um lugar x e todos os outros lugares estarão vazios”. É importante ressaltar que a propriedade de *soundness* implica na ausência de *livelocks* e *deadlocks* (AALST, 1999).

Para realizar a verificação necessária, o software Woflan espera, como entrada, um arquivo no formato TPN que representa uma Rede de Petri. Como saída, o software Woflan exibe a corretude de tais propriedades.

A análise das propriedades anteriormente listadas são exibidas na tabela 7.1.

A verificação sintática é feita no nível estrutural de uma Rede de Petri, ou seja, um exemplo de erro sintático seria uma Rede de Petri contendo um arco entre dois lugares. A análise não apontou nenhum erro sintático na Rede de Petri relativa ao modelo de referência.

Para cada uma das transições de uma Rede de Petri, deve existir uma instância de execução do processo onde a transição é disparada. Se não existe tal instância, a propriedade de

ausência de transições mortas não é satisfeita. Além disso, uma transição é viva se não existir nenhuma marca alcançável para a qual a transição é morta. Caso exista, a transição é não viva. O diagnóstico do software Woflan não revelou transições mortas ou não vivas.

Construções suspeitas são relativas a fragmentos do modelo que podem ocasionar algum estado indesejável (por exemplo *deadlock*). Um exemplo de construção suspeita é um desvio paralelo divergente complementado por um desvio exclusivo convergente. Observe que a construção suspeita do exemplo pode ser evitada utilizando-se as sete diretrizes de modelagem de processo, as quais foram utilizadas no trabalho como mencionado no capítulo 4. Nenhuma construção suspeita foi detectada pelo software Woflan.

Logo, todas as propriedades estão corretas, e isso indica que a Rede de Petri relacionada ao processo de referência não apresenta inconsistências semânticas e está verificada formalmente e, portanto, tanto o modelo de referência em Redes de Petri quanto o modelo de referência em BPMN não apresentam problemas semânticos e estruturais em sua modelagem.

Tabela 7.1 – Verificação da Rede de Petri

<i>Tipo de verificação</i>	<i>Correto?</i>
Verificação sintática	Sim
Ausência de construções suspeitas	Sim
Ausência de transições mortas	Sim
Ausência de transições não vivas	Sim
Soundness	Sim

A existência de erros na fase de modelagem de um processo é essencialmente custosa, pois tais erros são propagados ao longo do ciclo de vida do processo, e isso implica em uma maior dificuldade e maior custo para resolvê-los. Caso algum erro fosse encontrado em algum elemento na modelagem em Redes de Petri, o software Woflan disponibilizaria informações sobre como reparar o erro na Rede de Petri. Logo, em caso de erros, para ajustar o modelo de referência modelado em BPMN, as Redes de Petri deveriam ser traduzidas novamente para BPMN a fim de descobrir o foco do erro.

8 ANÁLISE COMPARATIVA E DE CONFORMIDADE DO PROCESSO DE REFERÊNCIA COM OS PROCESSOS DE SOFTWARES DE ATRACAMENTO MOLECULAR

A mineração de processos representa um papel importante na área de BPM. O principal objetivo é a descoberta de modelos de processo baseada nos dados disponíveis de *logs* de eventos (GEHRKE; WERNER, 2013).

Porém, como podemos obter informações confiáveis sobre a execução real de um processo de negócio?

Gehrke (2013) relata que a abordagem utilizada na mineração processo para a obtenção de informações confiáveis sobre a execução real de um processo de negócio baseia-se na exploração de dados armazenados em sistemas, os quais são gerados durante o processamento de transações de negócios. Um sistema armazena informações de dados em arquivos de *log* ou tabelas de banco de dados quando há o processamento de transações. A partir disso, extraem-se os dados disponibilizando-os em *logs* de eventos, os quais constituem a base de dados para o processo de algoritmos de mineração. Um *log* de eventos é basicamente uma tabela contendo todos os eventos registrados que se relacionam com atividades de negócios executados.

Os processos de atracamento molecular dos softwares Dockthor e Autodock Vina não geram quaisquer tipo de dados relacionados ao processo, sejam *logs*, sejam entradas em banco de dados. Não foram encontradas evidências na literatura de passos sistemáticos para realizar a extração de *logs* de processos que não geram *logs* ou de processos nos quais não é possível incorporar algoritmos de geração de *logs*. A saída encontrada para contornar tal situação foi gerar *logs* de eventos manuais, baseados em execuções sucessivas do processo do Dockthor e do Autodock Vina.

Foram realizadas, ao todo, 12 execuções manuais para cada um dos dois softwares. Tais execuções foram realizadas pelo autor a fim de identificar diferentes fluxos de execução que possibilitam realizar o processo completo de atracamento molecular. Em particular, a média de tempo para as execuções manuais de atracamento molecular no Autodock Vina foi de 10min15s, enquanto para o DockThor, 7min39seg.

Com base nas execuções manuais foi possível gerar *log* de eventos para o DockThor (foram identificados 5 logs de eventos únicos dentre as 12 execuções manuais do DockThor), representados nas tabelas 8.1 à 8.5, e *log* de eventos para o Autodock Vina¹ (foram identificados 5 logs de eventos únicos dentre as 12 execuções manuais do Autodock Vina), representado nas

¹O software Autodock Tools, ferramenta auxiliar desenvolvida para o Autodock, faz parte do processo do Autodock Vina por possibilitar tarefas como a preparação das proteínas, porém usa-se neste trabalho o nome do software Autodock Vina, representando o processo dos dois softwares em conjunto

tabelas 8.6 à 8.10. A coluna à esquerda indica a precedência de cada atividade. A precedência é definida a partir da ordem das atividades executadas pelo autor nos softwares de atracamento molecular. Por exemplo, no *case* da tabela 8.1 a atividade “selecionar moléculas interagentes” (precedência 1) foi executada anteriormente à atividade “determinar forma de representação dos dados” (precedência 2), já que seu valor de precedência é menor. Cada *case* representado nas tabelas indica um *log* de eventos único. Os *cases* não refletem em todas as possibilidades de execução do software: atividades não essenciais (por exemplo, zoom) ou particulares (por exemplo, Exportar arquivo) de cada um dos softwares não foram levadas em conta na criação dos *cases*.

O software de mineração de processos ProM consiste de uma série de *plugins* utilizáveis, em sua maioria, para mineração de dados. Entre eles destacam-se *plugins* com funções similares ao algoritmo Alpha² (DUMAS et al., 2013), possibilitando a descoberta de um processo a partir de *logs* de evento. O *plugin* “BPMN analysis using heuristic miner” foi utilizado a fim de gerar um modelo em BPMN para DockThor e AutodockVina a partir dos *logs* de eventos gerados. Os *logs* para execução do *plugin* mencionado deve estar no formato XES (GÜNTHER; VERBEEK, 2014). Apêndices A e B apresentam os arquivos XES baseados nos *logs* de eventos.

Tabela 8.1 – Log de eventos do processo do software DockThor - Case 1

Precedência	Atividade
1	Selecionar moléculas interagentes
2	Determinar forma de representação dos dados
3	Determinar método de busca para o sistema
4	Obter estrutura tridimensional do receptor
5	Preparar o receptor
6	Obter estrutura tridimensional do ligante
7	Preparar o ligante
8	Decidir entre adicionar ou não cofatores
9	Adicionar cofatores
10	Determinar função de avaliação para o sistema
11	Determinar sítio de ligação
12	Executar algoritmo
13	Avaliar resultados da execução

²O algoritmo Alpha é utilizado para descoberta de modelos de processo a partir de logs de evento

Tabela 8.2 – Log de eventos do processo do software DockThor - Case 2

Precedência	Atividade
1	Selecionar moléculas interagentes
2	Determinar forma de representação dos dados
3	Determinar método de busca para o sistema
4	Obter estrutura tridimensional do receptor
5	Preparar o receptor
6	Obter estrutura tridimensional do ligante
7	Preparar o ligante
8	Decidir entre adicionar ou não cofatores
10	Determinar função de avaliação para o sistema
11	Determinar sítio de ligação
12	Executar algoritmo
13	Avaliar resultados da execução

Tabela 8.3 – Log de eventos do processo do software DockThor - Case 3

Precedência	Atividade
1	Selecionar moléculas interagentes
2	Determinar forma de representação dos dados
3	Determinar método de busca para o sistema
4	Obter estrutura tridimensional do receptor
5	Obter estrutura tridimensional do ligante
6	Preparar o receptor
7	Preparar o ligante
8	Decidir entre adicionar ou não cofatores
9	Adicionar cofatores
10	Determinar função de avaliação para o sistema
11	Determinar sítio de ligação
12	Executar algoritmo
13	Avaliar resultados da execução

Tabela 8.4 – Log de eventos do processo do software DockThor - Case 4

Precedência	Atividade
1	Selecionar moléculas interagentes
2	Determinar forma de representação dos dados
3	Determinar método de busca para o sistema
4	Obter estrutura tridimensional do ligante
5	Obter estrutura tridimensional do receptor
6	Preparar o receptor
7	Preparar o ligante
8	Decidir entre adicionar ou não cofatores
9	Adicionar cofatores
10	Determinar função de avaliação para o sistema
11	Determinar sítio de ligação
12	Executar algoritmo
13	Avaliar resultados da execução

Tabela 8.5 – Log de eventos do processo do software DockThor - Case 5

Precedência	Atividade
1	Selecionar moléculas interagentes
2	Determinar forma de representação dos dados
3	Determinar método de busca para o sistema
4	Obter estrutura tridimensional do ligante
5	Obter estrutura tridimensional do receptor
6	Preparar o receptor
7	Preparar o ligante
8	Decidir entre adicionar ou não cofatores
9	Determinar função de avaliação para o sistema
10	Determinar sítio de ligação
11	Executar algoritmo
12	Avaliar resultados da execução

Tabela 8.6 – Log de eventos do processo do software Autodock Vina - Case 1

Precedência	Atividade
1	Selecionar moléculas interagentes
2	Determinar forma de representação dos dados
3	Obter estrutura tridimensional do receptor
4	Obter estrutura tridimensional do ligante
5	Preparar o receptor
6	Preparar o ligante
7	Decidir entre adicionar ou não cofatores
8	Adicionar cofatores
9	Determinar sítio de ligação
10	Determinar método de busca para o sistema
11	Determinar função de avaliação para o sistema
12	Executar algoritmo
13	Avaliar resultados da execução

Tabela 8.7 – Log de eventos do processo do software Autodock Vina - Case 2

Precedência	Atividade
1	Selecionar moléculas interagentes
2	Determinar forma de representação dos dados
3	Obter estrutura tridimensional do receptor
4	Preparar o receptor
5	Obter estrutura tridimensional do ligante
6	Preparar o ligante
7	Decidir entre adicionar ou não cofatores
8	Determinar sítio de ligação
9	Determinar método de busca para o sistema
10	Determinar função de avaliação para o sistema
11	Executar algoritmo
12	Avaliar resultados da execução

Tabela 8.8 – Log de eventos do processo do software Autodock Vina - Case 3

Precedência	Atividade
1	Selecionar moléculas interagentes
2	Determinar forma de representação dos dados
3	Obter estrutura tridimensional do ligante
4	Preparar o ligante
5	Obter estrutura tridimensional do receptor
6	Preparar o receptor
7	Decidir entre adicionar ou não cofatores
8	Adicionar cofatores
9	Determinar sítio de ligação
10	Determinar método de busca para o sistema
11	Determinar função de avaliação para o sistema
12	Executar algoritmo
13	Avaliar resultados da execução

Tabela 8.9 – Log de eventos do processo do software Autodock Vina - Case 4

Precedência	Atividade
1	Selecionar moléculas interagentes
2	Determinar forma de representação dos dados
3	Obter estrutura tridimensional do receptor
4	Preparar o receptor
5	Obter estrutura tridimensional do ligante
6	Preparar o ligante
7	Decidir entre adicionar ou não cofatores
8	Adicionar cofatores
9	Determinar sítio de ligação
10	Determinar método de busca para o sistema
11	Determinar função de avaliação para o sistema
12	Executar algoritmo
13	Avaliar resultados da execução

Tabela 8.10 – Log de eventos do processo do software Autodock Vina - Case 5

Precedência	Atividade
1	Selecionar moléculas interagentes
2	Determinar forma de representação dos dados
3	Obter estrutura tridimensional do ligante
4	Obter estrutura tridimensional do receptor
5	Preparar o ligante
6	Preparar o receptor
7	Decidir entre adicionar ou não cofatores
8	Determinar sítio de ligação
9	Determinar método de busca para o sistema
10	Determinar função de avaliação para o sistema
11	Executar algoritmo
12	Avaliar resultados da execução

DockThor (MAGALHÃES et al., 2014) é um software de atracamento molecular disponível em portal de acesso aberto ao público. No portal é possível submeter dados para o atracamento molecular desejado e, em seguida, um e-mail é enviado ao usuário com o resultado do processamento. Apesar de ser intuitivo e gratuito, o processo de atracamento molecular no software DockThor não é tão flexível. A título de exemplo, a tarefa de preparação da proteína deve ocorrer essencialmente antes da tarefa de preparação do ligante. Como visto no modelo de referência, as duas tarefas poderiam ser feitas em paralelo, o que pode otimizar a execução do processo.

Autodock Vina é um software gratuito de atracamento molecular. É utilizado em conjunto com o software Autodock Tools, o qual possibilita ajustes e preparações nas moléculas a serem ligadas no processo de atracamento molecular, além de definição do sítio de ligação entre as moléculas.

Tanto DockThor quanto Autodock Vina não possibilitam que o usuário participe de certas tarefas como determinar ou selecionar entre parâmetros desejados na função de avaliação nem especificar qual método de busca deseja utilizar durante a execução. Autodock Vina oferece como função de avaliação uma função híbrida (parte empírica, parte baseada em conhecimento), extraindo informação empírica das preferências conformacionais do complexo receptor-ligante e as medições de afinidade experimentais (TROTT; OLSON, 2010). Já o software DockThor utiliza função de avaliação baseada no campo de força MMFF94S (HALGREN, 1996).

Quanto ao método de busca, DockThor utiliza método de busca estocástica. Autodock Vina também utiliza método de busca estocástica, baseado em uma variedade de algoritmos (genéticos, *simulated annealing*) combinado com pequenos ajustes para otimização. O método de busca em ambos os softwares é único, ou seja, não existe a opção de escolha do usuário entre diferentes métodos.

DockThor e Autodock Vina utilizam *grids* para representação do sistema de atracamento molecular.

A determinação do sítio de ligação pode feita pelo usuário tanto no processo do DockThor quanto no processo do Autodock Vina. No DockThor a determinação é feita na tarefa anterior à execução do algoritmo (exatamente como mostrado no modelo de referência); porém, no Autodock Vina, o sítio de ligação é determinado logo após a preparação de ligante e receptor.

O sítio de ligação é determinado, no software DockThor, depois que todo o restante do sistema está preparado. No caso do Autodock Vina, o sítio de ligação é preparado antes da determinação do método de busca, utilizando o software Autodock Tools.

Tanto os processos Autodock Vina quanto DockThor possibilitam a adição de cofatores, sendo uma tarefa opcional dependente da decisão do usuário.

Por último, a execução do algoritmo leva em conta todas as tarefas anteriores do processo. O software DockThor exibe os resultados da execução no próprio portal, possibilitando que os resultados da afinidade de ligação das moléculas sejam analisados pelo usuário. Autodock Vina exibe os resultados da execução no terminal, permitindo também análise dos resultados pelo usuário.

Entre as principais observações sobre os processos dos softwares DockThor e Autodock Vina ressalta-se a pouca ou, em alguns casos, nenhuma liberdade para executar tarefas que poderiam ser feitas em paralelo, além de não disponibilizarem ao usuário meios para customizar a busca conformacional, por exemplo, por estabelecer funções de avaliação e métodos de busca fixos. A não conformidade dos softwares analisados com o modelo de referência do processo de atracamento molecular é percebida, porém as tarefas executadas são as mesmas, sendo cada um dos dois softwares analisados possui seu próprio fluxo e ordem de execução das tarefas.

8.1 Análise de Conformidade

A análise de conformidade é responsável por detectar inconsistências entre um modelo de processo e correspondentes *logs* de execução do mesmo processo, quantificando-as na forma de métricas (ROZINAT,).

Alguns estudos exploram a análise de conformidade entre um modelo de processo e *logs* de eventos relacionados. Rozinat e Aalst (2006) propõem um método incremental para avaliar a conformidade de um modelo de processo com os *logs* de evento, analisando propriedades como *aptidão* (O processo observado cumpre o fluxo de controle especificado pelo modelo de processo?) e *adequação* (O modelo descreve o processo observado de um modo adequado?). Como resultado, um plugin verificador de conformidade é apresentado para o framework *ProM*.

Aalst e colaboradores (2008) abordam o problema de verificar e quantificar o quanto o comportamento real de um serviço, como registrado em *logs* de eventos, está em conformidade com o comportamento esperado como especificado em um modelo de processo, utilizando para isso o verificador de conformidade do framework ProM mencionado no último parágrafo.

Ainda, Aalst e colaboradores (2006) exploram análise de conformidade de uma coreografia utilizando a linguagem BPEL e mensagens SOAP, traduzindo BPEL para Redes de Petri e mensagens SOAP para MXML.

Murphy e colaboradores (2001) relatam três fatores para análise de conformidade: *convergência*, *divergência* e *ausência*. *Convergência* indica relações esperadas pelo usuário; *divergência* indica relações não esperadas pelo usuário; *ausência* indica relações esperadas, porém não existentes. Esses três fatores são utilizados neste trabalho para indicar inconsistências entre o modelo de referência e os modelos de processo do DockThor e do Autodock Vina.

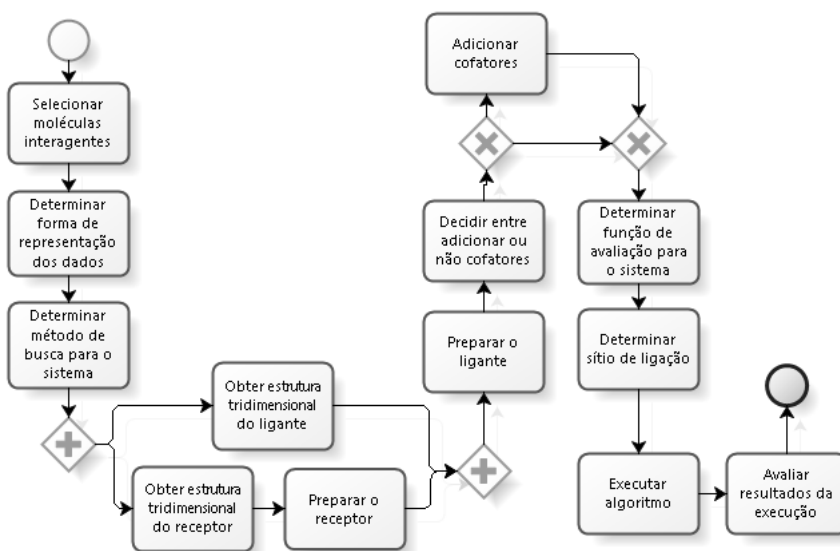


Figura 8.1 – Processo minerado do software Dokthor

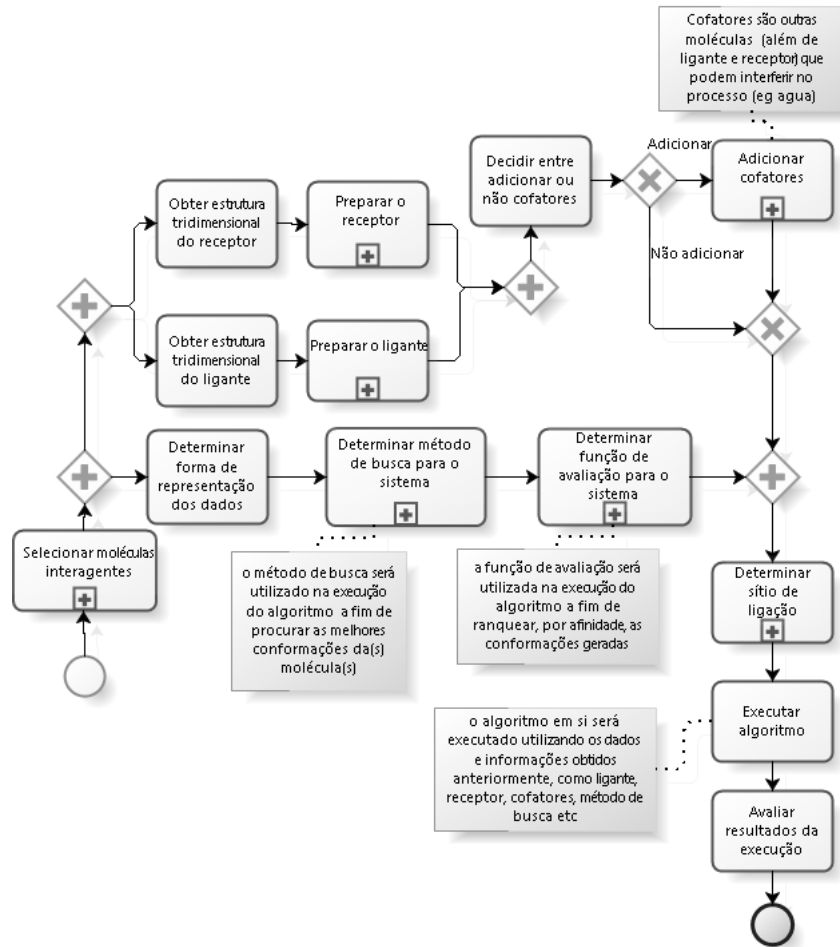


Figura 8.2 – Modelo de referência

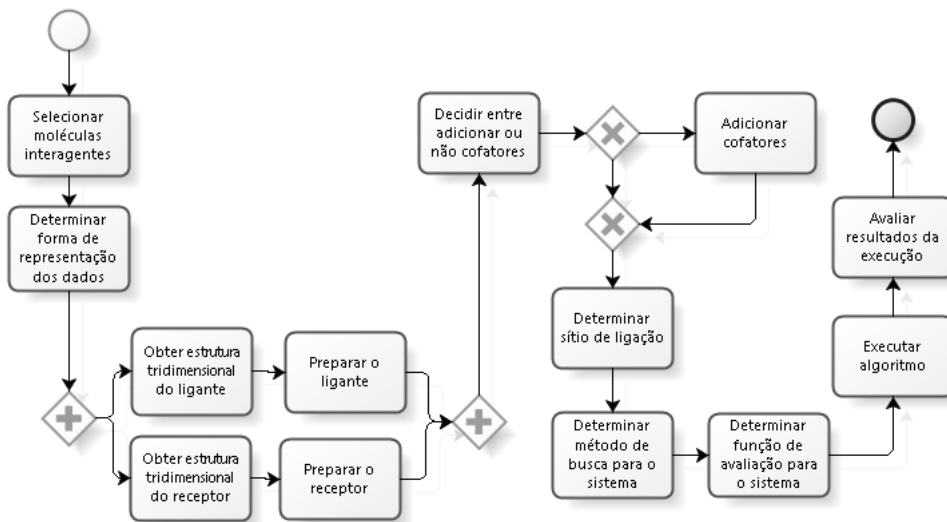


Figura 8.3 – Processo minerado do software Autodock Vina

8.1.1 Conformidade entre modelo de referência e modelo de processo do software DockThor

Em relação ao modelo de processo do software do DockThor (figura 8.1), é possível, principalmente, perceber a *ausência* da atividade “Preparar o Ligante” em paralelo à preparação do receptor. Além disso, há *ausência* das atividades “Determinar forma de representação dos dados”, “Determinar método de busca para o sistema” e “Determinar função de avaliação” como parte da execução paralela do modelo de referência (figura 8.2). Repare que tais atividades, porém, existem no modelo de processo do DockThor, ou seja, estão deslocadas em relação ao modelo de referência, criando *divergências*. Há *convergência*, porém, nas atividades relacionadas à adição de cofatores, além das atividades “Selecionar moléculas interagentes”, “Obter estrutura tridimensional do ligante”, “Obter estrutura tridimensional do receptor”, “Preparar o receptor”, “Determinar sítio de ligação”, “Executar algoritmo” e “Avaliar resultados da execução”.

Portanto, existem diversas inconformidades na ordem que as atividades são executadas, apesar de todas as atividades existentes nos *logs* de eventos do DockThor constarem no modelo de referência.

8.1.2 Conformidade entre modelo de referência e modelo de processo do software Autodock Vina

Os *logs* de eventos do Autodock Vina, em contraste aos logs do DockThor, não são completamente executáveis no modelo de referência: o principal motivo é a *divergência* da atividade “Determinar sítio de ligação”, que é realizada antes de “Determinar método de busca para o sistema”, o que não ocorre no modelo de referência. Entre as principais *ausências* no modelo de processo do Autodock Vina (figura 8.3) estão as atividades “Determinar forma de representação dos dados”, “Determinar método de busca para o sistema” e “Determinar função de avaliação” como parte da execução paralela do modelo de referência. Os dois modelos convergem, porém, nas atividades “Selecionar moléculas interagentes”, “Obter estrutura tridimensional do ligante”, “Obter estrutura tridimensional do receptor”, “Preparar o receptor”, “Preparar o ligante”, “Executar algoritmo” e “Avaliar resultados da execução”.

Analogamente ao DockThor, todas as atividades dos *logs* de eventos do Autodock Vina constam no modelo de referência. São encontradas, no entanto, diversas inconformidades na ordem que essas atividades são executadas.

9 CONCLUSÃO

No presente trabalho foi apresentado um modelo de referência na notação BPMN para o processo de atracamento molecular. Tal modelo foi refinado por especialistas da área a partir de entrevistas. Após, houve a validação externa do modelo com pesquisadores do domínio de atracamento molecular a partir de dois questionários aplicados a 9 pesquisadores do domínio, igualmente distribuídos entre conhecimento avançado, intermediário e básico do domínio de atracamento molecular.

Subsequentemente, o modelo de referência foi mapeado para Redes de Petri, um formalismo que possibilita diagnósticos e análises formais complexas, as quais verificam propriedades do modelo como *soundness*, ausência de transições mortas, entre outras. O resultado do mapeamento é o modelo de referência em BPMN representado em Redes de Petri. O modelo de referência em Redes de Petri, então, foi utilizado para análise comparativa e de conformidade com o modelo extraído a partir de mineração dos processos de dois softwares utilizados para atracamento molecular (DockThor e Autodock Vina). Tal análise mostra que os modelos dos dois softwares mencionados não estão em conformidade com o modelo de referência, afinal existem diferenças estruturais entre eles (divergência e ausência).

O conceito de BPM, além de sua utilidade para o trabalho, foi positivamente avaliado como compreensível para modelar um processo complexo com vocabulário especialista, como atracamento molecular, usando-se a literatura como referência e tendo um mínimo possível de interferência de entrevistas. O modelo de referência gerado na notação BPMN teve uma alta aceitação dos usuários, os quais avaliaram muito bem o nível de compreensão do modelo. Ainda, alguns usuários elogiaram a iniciativa e reiteraram que a modelagem utilizando BPMN é bastante clara e facilita o entendimento do processo.

Uma das limitações do trabalho foi a maneira como os *logs* de eventos foram gerados na comparação entre os processos do Dockthor, Autodock Vina e o modelo de referência. Os softwares de atracamento molecular não geram informações de dados em arquivos de *log* ou tabelas de banco de dados referentes ao processo e, portanto, os *logs* de eventos foram gerados manualmente a partir de execuções sucessivas dos processos dos softwares.

Trabalhos futuros são sugeridos a fim de processar as demais fases do ciclo de vida de um processo que não foram abordadas no trabalho. Ainda, uma maior granularidade das tarefas pode ser explorada a fim de complementar o trabalho realizado. A fim de gerar *logs* de execução de maneira automática, sugere-se o desenvolvimento de uma ferramenta que capture *logs* de execução de softwares que não geram *logs* explicitamente. Neste trabalho utilizaram-se

logs gerados pelo autor; sugere-se em trabalhos futuros, utilização de *logs* extraídos a partir de observação da utilização por especialistas dos softwares de atracamento molecular. Por último, é importante relacionar a escolha de um determinado software em relação à necessidade, em particular utilizando configuração de produtos e de processos.

Complementar à monografia, foi elaborado um poster intitulado *Process Design and Conformity Analysis of Computational Strategies for Molecular Docking*, o qual foi apresentado no X-Meeting 2015 (*11th International Conference of th AB3C + Brazilian Symposium of Bioinformatics*).

REFERÊNCIAS

- AALST, W. M. P. van der. Woflan: A petri-net-based workflow analyzer. **Syst. Anal. Model. Simul.**, Gordon and Breach Science Publishers, Inc., Newark, NJ, USA, v. 35, n. 3, p. 345–357, may 1999. ISSN 0232-9298. Disponível na Internet: <<http://dl.acm.org/citation.cfm?id=316218.316234>>.
- ADAMS, C. P.; BRANTNER, V. V. Spending on new drug development1. **Health Economics**, John Wiley Sons, Ltd., v. 19, n. 2, p. 130–141, 2010. ISSN 1099-1050. Disponível na Internet: <<http://dx.doi.org/10.1002/hec.1454>>.
- ALLWEYER, T. **BPMN 2.0**. [S.l.]: BoD, 2010. ISBN 3839149851, 9783839149850.
- ALTMAN, R. B.; DUGAN, J. M. Defining bioinformatics and structural bioinformatics. In: _____. **Structural Bioinformatics**. John Wiley Sons, Inc., 2005. p. 1–14. ISBN 9780471721208. Disponível na Internet: <<http://dx.doi.org/10.1002/0471721204.ch1>>.
- AYUTAYA, N.; PALUNGSUNTIKUL, P.; PREMCHAIWADI, W. Heuristic mining: Adaptive process simplification in education. In: **ICT and Knowledge Engineering (ICT Knowledge Engineering), 2012 10th International Conference on**. [S.l.: s.n.], 2012. p. 221–227. ISSN 2157-0981.
- BARJIS, J. The importance of business process modeling in software systems design. **Science of Computer Programming**, v. 71, n. 1, p. 73 – 87, 2008. ISSN 0167-6423. Disponível na Internet: <<http://www.sciencedirect.com/science/article/pii/S0167642308000038>>.
- BERMAN, H. M. et al. The protein data bank. *Nucleic Acids Res*, 2000.
- BRINK, T. ten; EXNER, T. E. Influence of protonation, tautomeric, and stereoisomeric states on proteinligand docking results. **Journal of Chemical Information and Modeling**, v. 49, n. 6, p. 1535–1546, 2009. PMID: 19453150. Disponível na Internet: <<http://dx.doi.org/10.1021/ci800420z>>.
- CARLEY, K. M. **Validating Computational Models**. 1996.
- DAVID, R.; ALLA, H. **Discrete, Continuous, and Hybrid Petri Nets**. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 3642106684, 9783642106682.
- DHANIK, A.; KAVRAKI, L. E. Protein–ligand interactions: Computational docking. John Wiley & Sons, Ltd, 2001. Disponível na Internet: <<http://dx.doi.org/10.1002/9780470015902.a0004105.pub2>>.
- DIJKMAN, R. M.; DUMAS, M.; OUYANG, C. Semantics and analysis of business process models in bpmn. **Inf. Softw. Technol.**, Butterworth-Heinemann, Newton, MA, USA, v. 50, n. 12, p. 1281–1294, nov. 2008. ISSN 0950-5849. Disponível na Internet: <<http://dx.doi.org/10.1016/j.infsof.2008.02.006>>.
- DINGLE, N.; KNOTTENBELT, W.; SUTO, T. Pipe2: A tool for the performance evaluation of generalised stochastic petri nets. **ACM SIGMETRICS Performance Evaluation Review (Special Issue on Tools for Computer Performance Modelling and Reliability Analysis)**, v. 36(4), p. 34–39, 2009.

DONGEN, B. F. van et al. The prom framework: A new era in process mining tool support. In: **Proceedings of the 26th International Conference on Applications and Theory of Petri Nets**. Berlin, Heidelberg: Springer-Verlag, 2005. (ICATPN'05), p. 444–454. ISBN 3-540-26301-2, 978-3-540-26301-2. Disponível na Internet: <http://dx.doi.org/10.1007/11494744_25>.

DUMAS, M. et al. **Fundamentals of Business Process Management**. [S.l.]: Springer, 2013. I-XXVII, 1-399 p. ISBN 978-3-642-33142-8.

ELOKELY, K. M.; DOERKSEN, R. J. Docking challenge: Protein sampling and molecular docking performance. **Journal of Chemical Information and Modeling**, v. 53, n. 8, p. 1934–1945, 2013. PMID: 23530568. Disponível na Internet: <<http://dx.doi.org/10.1021/ci400040d>>.

GEHRKE, N.; WERNER, M. Process mining. 2013.

GHOSH, S. et al. Structure-based virtual screening of chemical libraries for drug discovery. **Current Opinion in Chemical Biology**, v. 10, n. 3, p. 194 – 202, 2006. ISSN 1367-5931. Combinatorial chemistry and molecular diversity Scott K Silverman and Paul J Hergenrother. Disponível na Internet: <<http://www.sciencedirect.com/science/article/pii/S1367593106000536>>.

GRAAF, hris de et al. Catalytic site prediction and virtual screening of cytochrome p450 2d6 substrates by consideration of water and rescoring in automated docking. **Journal of Medicinal Chemistry**, v. 49, n. 8, p. 2417–2430, 2006. PMID: 16610785. Disponível na Internet: <<http://dx.doi.org/10.1021/jm0508538>>.

GUEDES, I. A.; MAGALHÃES, C. S. de; DARDENNE, L. E. Atracamento molecular. In: VERLI, H. (Ed.). **Bioinformática: da Biologia à Flexibilidade Molecular**. [S.l.]: SBBq, 2014.

GÜNTHER, C. W.; VERBEEK, H. M. W. **XES – Standard Definition**. [S.l.], 2014. Disponível na Internet: <<http://bpmcenter.org/wp-content/uploads/reports/2014/BPM-14-09.pdf>>.

HALGREN, T. A. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. **Journal of Computational Chemistry**, John Wiley Sons, Inc., v. 17, n. 5-6, p. 490–519, 1996. ISSN 1096-987X. Disponível na Internet: <[http://dx.doi.org/10.1002/\(SICI\)1096-987X\(199604\)17:5/6<490::AID-JCC1>3.0.CO;2-P](http://dx.doi.org/10.1002/(SICI)1096-987X(199604)17:5/6<490::AID-JCC1>3.0.CO;2-P)>.

HAWKINS, P. et al. Conformer generation with omega: Algorithm and validation using high quality structures from the protein databank and the cambridge structural database. In: . [S.l.]: J. Chem. Inf. Model, 2010. p. 572–584.

HERNÁNDEZ-SANTOYO, A. et al. 2013. Disponível na Internet: <<http://www.intechopen.com/books/protein-engineering-technology-and-application/protein-protein-and-protein-ligand-docking>>.

HUANG, S.-Y.; GRINTER, S. Z.; ZOU, X. Scoring functions and their evaluation methods for protein-ligand docking: recent advances and future directions. **Phys. Chem. Chem. Phys.**, The Royal Society of Chemistry, v. 12, p. 12899–12908, 2010. Disponível na Internet: <<http://dx.doi.org/10.1039/C0CP00151A>>.

IRWIN, J. J.; SHOICHET, B. K. Zinc a free database of commercially available compounds for virtual screening. **Journal of Chemical Information and Modeling**, v. 45, n. 1, p. 177–182, 2005. PMID: 15667143. Disponível na Internet: <<http://dx.doi.org/10.1021/ci049714+>>.

KALENKOVA, A. A.; LEONI, M. de; AALST, W. M. van der. Discovering, analyzing and enhancing bpmn models using prom. 2014.

KITCHEN, D. B. et al. Docking and scoring in virtual screening for drug discovery: methods and applications. **Nat Rev Drug Discov**, v. 3, n. 11, p. 935 – 949, 1996.

KLEINJUNG, J.; FRATERNALI, F. Design and application of implicit solvent models in biomolecular simulations. **Current Opinion in Structural Biology**, v. 25, p. 126 – 134, 2014. ISSN 0959-440X. Theory and simulation / Macromolecular machines. Disponível na Internet: <<http://www.sciencedirect.com/science/article/pii/S0959440X14000438>>.

LENGAUER, T.; RAREY, M. Computational methods for biomolecular docking. **Current Opinion in Structural Biology**, v. 6, n. 3, p. 402 – 406, 1996. ISSN 0959-440X. Disponível na Internet: <<http://www.sciencedirect.com/science/article/pii/S0959440X96800613>>.

MAGALHÃES, C. S. de et al. A dynamic niching genetic algorithm strategy for docking of highly flexible ligands. **Information Sciences**, p. 206–224, 2014.

MARTIN, Y. Let's not forget tautomers. **Journal of Computer-Aided Molecular Design**, Springer Netherlands, v. 23, n. 10, p. 693–704, 2009. ISSN 0920-654X. Disponível na Internet: <<http://dx.doi.org/10.1007/s10822-009-9303-2>>.

MATEJA, K. et al. Business process model and notation: The current state of affairs. **Computer Science and Information Systems**, v. 12, p. 509, 539, 2015.

MCCONATHY, J.; OWENS, M. Stereochemistry in drug action. **Primary Care Companion to The Journal of Clinical Psychiatry**, v. 5.2, p. 70 – 73, 2003.

MENDLING, J.; REIJERS, H. A.; AALST, W. M. P. van der. Seven process modeling guidelines (7pmsg). **Inf. Softw. Technol.**, Butterworth-Heinemann, Newton, MA, USA, v. 52, n. 2, p. 127–136, feb. 2010. ISSN 0950-5849. Disponível na Internet: <<http://dx.doi.org/10.1016/j.infsof.2009.08.004>>.

MENG, X.-Y. et al. Molecular docking: a powerful approach for structure-based drug discovery. v. 7, p. 146 – 157, 2011.

NUFFEL, D. V.; MULDER, H.; KERVEL, S. V. Enhancing the formal foundations of bpmn by enterprise ontology. In: ALBANI, A.; BARJIS, J.; DIETZ, J. (Ed.). **Advances in Enterprise Engineering III**. Springer Berlin Heidelberg, 2009, (Lecture Notes in Business Information Processing, v. 34). p. 115–129. ISBN 978-3-642-01914-2. Disponível na Internet: <http://dx.doi.org/10.1007/978-3-642-01915-9_9>.

PETERSON, J. L. Petri nets. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 9, n. 3, p. 223–252, sep. 1977. ISSN 0360-0300. Disponível na Internet: <<http://doi.acm.org/10.1145/356698.356702>>.

RECKER, J. C. Opportunities and constraints : the current struggle with bpmn. **Business Process Management Journal**, Emerald, v. 16, n. 1, p. 181–201, 2010. Disponível na Internet: <<http://eprints.qut.edu.au/20316/>>.

ROBERTS, B. C.; MANCERA, R. L. Ligandprotein docking with water molecules. **Journal of Chemical Information and Modeling**, v. 48, n. 2, p. 397–408, 2008. PMID: 18211049. Disponível na Internet: <<http://dx.doi.org/10.1021/ci700285e>>.

ROUX, B.; SIMONSON, T. Implicit solvent models. **Biophysical Chemistry**, v. 78, n. 1–2, p. 1 – 20, 1999. ISSN 0301-4622. Disponível na Internet: <<http://www.sciencedirect.com/science/article/pii/S0301462298002269>>.

ROZINAT, A. **Conformance Checking of Processes Based on Monitoring Real Behavior.**

SCHNEIDER, G. Virtual screening in drug discovery. edited by Juan Alvarez and Brian Shoichet. **ChemBioChem**, WILEY-VCH Verlag, v. 7, n. 2, p. 390–390, 2006. ISSN 1439-7633. Disponível na Internet: <<http://dx.doi.org/10.1002/cbic.200500530>>.

SEATZU, C.; SILVA, M.; SCHUPPEN, J. H. van. **Control of Discrete-event Systems: Automata and Petri Net Perspectives.** [S.l.]: Springer, 2013.

SEILER, K. P. et al. ChEMBL: a small-molecule screening and cheminformatics resource database. *Nucleic Acids Res.*, 2008.

TROTT, O.; OLSON, A. J. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading. **Journal of Computational Chemistry** **31**, p. 455–461, 2010.

VYAS, V. et al. Virtual screening: A fast tool for drug design. *Sci Pharm*, 2008.

WASS, M. N.; KELLEY, L. A.; STERNBERG, M. J. E. 3dligandsite: predicting ligand-binding sites using similar structures. **Nucleic Acids Res**, p. 469–473, 2010.

WINTER, R.; SCHELP, J. Reference modeling and method construction: A design science perspective. In: **Proceedings of the 2006 ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2006. (SAC '06), p. 1561–1562. ISBN 1-59593-108-2. Disponível na Internet: <<http://doi.acm.org/10.1145/1141277.1141638>>.

YE, J. et al. Formal semantics of bpmn process models using yawl. In: **Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on.** [S.l.: s.n.], 2008. v. 2, p. 70–74.

ZIELENKIEWICZ, P.; RABCZENKO, A. Searching for interacting surfaces of proteins—the improved method. **Journal of Theoretical Biology**, v. 116, n. 4, p. 607 – 612, 1985. ISSN 0022-5193. Disponível na Internet: <<http://www.sciencedirect.com/science/article/pii/S0022519385800904>>.

APÊNDICES

Apêndice A - Arquivos de log de eventos do DockThor no padrão XES

```

<?xml version="1.0" encoding="UTF-8" ?>
<log xes.version="1.0" xes.features="nested-attributes"
openxes.version="1.0RC7"
  xmlns="http://www.xes-standard.org/">
  <extension name="Lifecycle" prefix="lifecycle"
uri="http://www.xes-standard.org/lifecycle.xesext"/>
  <extension name="Organizational" prefix="org"
uri="http://www.xes-standard.org/org.xesext"/>
  <extension name="Concept" prefix="concept"
uri="http://www.xes-standard.org/concept.xesext"/>
  <global scope="trace">
    <string key="concept:name" value="__INVALID__"/>
  </global>
  <global scope="event">
    <string key="concept:name" value="__INVALID__"/>
    <string key="lifecycle:transition" value="complete"/>
  </global>
  <classifier name="MXML Legacy Classifier" keys="concept:name
lifecycle:transition"/>
  <classifier name="Event Name" keys="concept:name"/>
  <classifier name="Resource" keys="org:resource"/>
  <string key="source" value="Rapid Synthesizer"/>
  <string key="concept:name" value="dockthor.mxml"/>
  <string key="lifecycle:model" value="standard"/>
  <trace>
    <string key="concept:name" value="Casol.0"/>
    <event>
      <string key="org:resource" value="UNDEFINED"/>
      <string key="concept:name"
value="Selecionar moléculas interagentes"/>

```

```
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar forma de representação dos dados"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar método de busca para o sistema"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Obter estrutura tridimensional do receptor"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Preparar o receptor"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Obter estrutura tridimensional do ligante"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
```

```

    <string key="concept:name"
    value="Preparar o ligante"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
    value="Decidir entre adicionar ou não cofatores"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
    value="Adicionar cofatores"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
    value="Determinar função de avaliação para o sistema"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
    value="Determinar sítio de ligação"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
    value="Executar algoritmo"/>
    <string key="lifecycle:transition" value="complete"/>
</event>

```

```
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Avaliar resultados da execução"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
</trace>
<trace>
  <string key="concept:name" value="Caso2.0"/>
  <event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Selecionar moléculas interagentes"/>
    <string key="lifecycle:transition" value="complete"/>
  </event>
  <event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar forma de representação dos dados"/>
    <string key="lifecycle:transition" value="complete"/>
  </event>
  <event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar método de busca para o sistema"/>
    <string key="lifecycle:transition" value="complete"/>
  </event>
  <event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Obter estrutura tridimensional do receptor"/>
    <string key="lifecycle:transition" value="complete"/>
  </event>
  <event>
```

```

    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Preparar o receptor"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Obter estrutura tridimensional do ligante"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Preparar o ligante"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Decidir entre adicionar ou não cofatores"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar função de avaliação para o sistema"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar sítio de ligação"/>
    <string key="lifecycle:transition" value="complete"/>

```

```
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Executar algoritmo"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Avaliar resultados da execução"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
</trace>
<trace>
  <string key="concept:name" value="Caso3.0"/>
  <event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Selecionar moléculas interagentes"/>
    <string key="lifecycle:transition" value="complete"/>
  </event>
  <event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar forma de representação dos dados"/>
    <string key="lifecycle:transition" value="complete"/>
  </event>
  <event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar método de busca para o sistema"/>
    <string key="lifecycle:transition" value="complete"/>
  </event>
</trace>
```

```
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Obter estrutura tridimensional do receptor"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Obter estrutura tridimensional do ligante"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Preparar o receptor"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Preparar o ligante"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Decidir entre adicionar ou não cofatores"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Adicionar cofatores"/>
```

```
        <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar função de avaliação para o sistema"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar sítio de ligação"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Executar algoritmo"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Avaliar resultados da execução"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
</trace>
<trace>
    <string key="concept:name" value="Caso4.0"/>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
value="Selecionar moléculas interagentes"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
</trace>
```



```
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Determinar forma de representação dos dados"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Determinar método de busca para o sistema"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Obter estrutura tridimensional do ligante"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Obter estrutura tridimensional do receptor"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Preparar o receptor"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
```

```

        value="Preparar o ligante"/>
        <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Decidir entre adicionar ou não cofatores"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Adicionar cofatores"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar função de avaliação para o sistema"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar sítio de ligação"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Executar algoritmo"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>

```

```
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Avaliar resultados da execução"/>
<string key="lifecycle:transition" value="complete"/>
</event>
</trace>
<trace>
<string key="concept:name" value="Caso5.0"/>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Selecionar moléculas interagentes"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Determinar forma de representação dos dados"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Determinar método de busca para o sistema"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Obter estrutura tridimensional do ligante"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
```

```
<string key="concept:name"
value="Obter estrutura tridimensional do receptor"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Preparar o receptor"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Preparar o ligante"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Decidir entre adicionar ou não cofatores"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Determinar função de avaliação para o sistema"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Determinar sítio de ligação"/>
<string key="lifecycle:transition" value="complete"/>
</event>
```

```
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Executar algoritmo"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Avaliar resultados da execução"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
</trace>
</log>
```

Apêndice B - Arquivos de log de eventos do Autodock Vina no padrão XES

```
<?xml version="1.0" encoding="UTF-8" ?>
<log xes.version="1.0" xes.features="nested-attributes"
openxes.version="1.0RC7"
  xmlns="http://www.xes-standard.org/">
  <extension name="Lifecycle" prefix="lifecycle"
uri="http://www.xes-standard.org/lifecycle.xesext"/>
  <extension name="Organizational" prefix="org"
uri="http://www.xes-standard.org/org.xesext"/>
  <extension name="Concept" prefix="concept"
uri="http://www.xes-standard.org/concept.xesext"/>
  <global scope="trace">
    <string key="concept:name" value="__INVALID__"/>
  </global>
  <global scope="event">
    <string key="concept:name" value="__INVALID__"/>
    <string key="lifecycle:transition" value="complete"/>
  </global>
  <classifier name="MXML Legacy Classifier" keys="concept:name
lifecycle:transition"/>
  <classifier name="Event Name" keys="concept:name"/>
  <classifier name="Resource" keys="org:resource"/>
  <string key="source" value="Rapid Synthesizer"/>
  <string key="concept:name" value="vina.mxml"/>
  <string key="lifecycle:model" value="standard"/>
  <trace>
    <string key="concept:name" value="Casol.0"/>
    <event>
      <string key="org:resource" value="UNDEFINED"/>
      <string key="concept:name"
value="Selecionar moléculas interagentes"/>
      <string key="lifecycle:transition" value="complete"/>
    </event>
  </trace>
</log>
```

```
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Determinar forma de representação dos dados"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Obter estrutura tridimensional do receptor"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Obter estrutura tridimensional do ligante"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Preparar o receptor"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Preparar o ligante"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Decidir entre adicionar ou não cofatores"/>
```

```

    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Adicionar cofatores"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar sítio de ligação"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar método de busca para o sistema"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar função de avaliação para o sistema"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Executar algoritmo"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>

```



```

        <string key="concept:name"
        value="Avaliar resultados da execução"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
</trace>
<trace>
    <string key="concept:name" value="Caso2.0"/>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Selecionar moléculas interagentes"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Determinar forma de representação dos dados"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Obter estrutura tridimensional do receptor"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Preparar o receptor"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"

```

```
        value="Obter estrutura tridimensional do ligante"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Preparar o ligante"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Decidir entre adicionar ou não cofatores"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Determinar sítio de ligação"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Determinar método de busca para o sistema"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Determinar função de avaliação para o sistema"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
```

```
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Executar algoritmo"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Avaliar resultados da execução"/>
<string key="lifecycle:transition" value="complete"/>
</event>
</trace>
<trace>
<string key="concept:name" value="Caso3.0"/>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Selecionar moléculas interagentes"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Determinar forma de representação dos dados"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Obter estrutura tridimensional do ligante"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
```

```
<string key="concept:name"
value="Preparar o ligante"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Obter estrutura tridimensional do receptor"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Preparar o receptor"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Decidir entre adicionar ou não cofatores"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Adicionar cofatores"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Determinar sítio de ligação"/>
<string key="lifecycle:transition" value="complete"/>
</event>
```

```
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Determinar método de busca para o sistema"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Determinar função de avaliação para o sistema"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Executar algoritmo"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Avaliar resultados da execução"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
</trace>
<trace>
  <string key="concept:name" value="Caso4.0"/>
  <event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Selecionar moléculas interagentes"/>
    <string key="lifecycle:transition" value="complete"/>
  </event>
  <event>
```

```
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Determinar forma de representação dos dados"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Obter estrutura tridimensional do receptor"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Preparar o receptor"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Obter estrutura tridimensional do ligante"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Preparar o ligante"/>
<string key="lifecycle:transition" value="complete"/>
</event>
<event>
<string key="org:resource" value="UNDEFINED"/>
<string key="concept:name"
value="Decidir entre adicionar ou não cofatores"/>
<string key="lifecycle:transition" value="complete"/>
```

```
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Adicionar cofatores"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Determinar sítio de ligação"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Determinar método de busca para o sistema"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Determinar função de avaliação para o sistema"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
value="Executar algoritmo"/>
  <string key="lifecycle:transition" value="complete"/>
</event>
<event>
  <string key="org:resource" value="UNDEFINED"/>
  <string key="concept:name"
```

```
        value="Avaliar resultados da execução"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
</trace>
<trace>
    <string key="concept:name" value="Caso5.0"/>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Selecionar moléculas interagentes"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Determinar forma de representação dos dados"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Obter estrutura tridimensional do ligante"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Obter estrutura tridimensional do receptor"/>
        <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
        <string key="org:resource" value="UNDEFINED"/>
        <string key="concept:name"
        value="Preparar o ligante"/>
```



```

    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Preparar o receptor"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Decidir entre adicionar ou não cofatores"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar sítio de ligação"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar método de busca para o sistema"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>
    <string key="concept:name"
value="Determinar função de avaliação para o sistema"/>
    <string key="lifecycle:transition" value="complete"/>
</event>
<event>
    <string key="org:resource" value="UNDEFINED"/>

```

```
    <string key="concept:name"  
    value="Executar algoritmo"/>  
    <string key="lifecycle:transition" value="complete"/>  
  </event>  
  <event>  
    <string key="org:resource" value="UNDEFINED"/>  
    <string key="concept:name"  
    value="Avaliar resultados da execução"/>  
    <string key="lifecycle:transition" value="complete"/>  
  </event>  
</trace>  
</log>
```

Apêndice C - Tutorial de BPMN disponibilizado para os questionários

TUTORIAL DE BPMN

Este documento tem como principal objetivo listar um subconjunto de elementos da notação BPMN e suas definições. Tais elementos são utilizados na modelagem de atracamento molecular que será apresentada a participantes do experimento. O tutorial tem como principais referências a especificação da notação BPMN e representações gráficas do modelador de processos Bizagi.

Elemento: Task (em português, “Tarefa”).

Definição: Uma ação única realizada dentro de um processo.

Representação gráfica:



Tarefa

Elemento: Sequence flow (em português, “Fluxo de Sequência”).

Definição: Utilizado para conectar elementos dentro de um processo, definindo uma ordem de execução.

Representação gráfica:

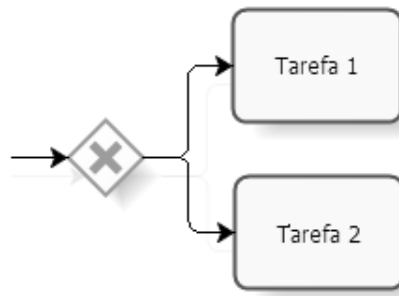


Fluxo de sequência

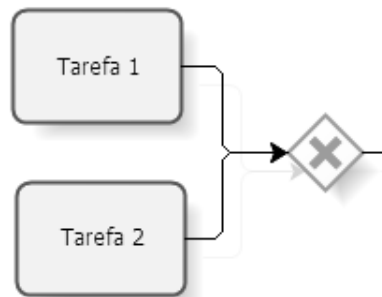
Elemento: Exclusive gateway (em português, “Desvio exclusivo”).

Definição: Um Desvio exclusivo divergente é utilizado para criar caminhos alternativos dentro de um processo e, a partir dali, apenas um caminho deve ser seguido. Um Desvio exclusivo convergente é utilizado para sincronizar (fundir) caminhos alternativos.

Representação gráfica:



Desvio exclusivo divergente

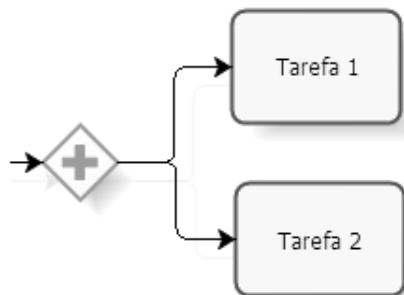


Desvio exclusivo convergente

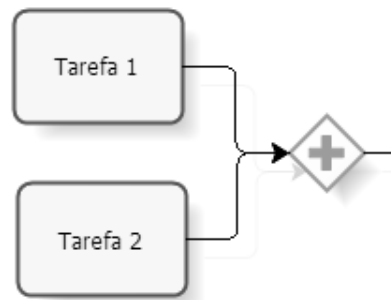
Elemento: Parallel gateway (em português, “Desvio paralelo”)

Definição: Um desvio paralelo divergente é utilizado para criar vários caminhos de execução paralelos. Um desvio paralelo convergente é utilizado para sincronizar dois ou mais fluxos paralelos, ou seja, o desvio paralelo convergente espera por todos os fluxos de sequência de chegada antes de continuar o fluxo através de seus fluxos de sequência de saída.

Representação gráfica:



Desvio paralelo divergente



Desvio paralelo convergente

Elemento: Start event (em português, “evento de início”)

Definição: O evento de início indica onde um processo em particular começa.

Representação gráfica:



Fluxo de sequência

Elemento: End event (em português, “evento de fim”)

Definição: O evento de fim indica onde um processo termina.

Representação gráfica:



Fluxo de sequência

Elemento: Subprocess (em português, “subprocesso”)

Definição: Um subprocesso é uma atividade que pode ser decomposta em um grau de granularidade maior através de um conjunto de subatividades.

Representação gráfica:



Fluxo de sequência

Elemento: Association (em português, “associação”)

Definição: Utilizada para ligar informações e artefatos à elementos gráficos da modelagem BPMN

Representação gráfica:

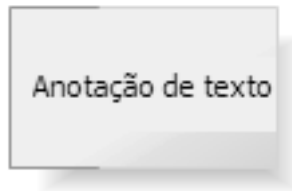
.....

Fluxo de sequência

Elemento: Text Annotation (em português, “anotação de texto”)

Definição: Utilizado para exibir informações adicionais ao leitor da modelagem. É sempre acompanhado de uma associação.

Representação gráfica:



Anotação de texto

Apêndice D - Questionário 1

Qual seu nível de conhecimento no processo de atracamento molecular?

- Nenhum
 Básico
 Intermediário
 Avançado

Qual seu nível de escolaridade?

- Superior Incompleto ou Superior Completo
 Mestrado
 Doutorado ou superior

Com que frequência você utiliza ferramentas de atracamento molecular?

Nunca Sempre

Quais os nomes das ferramentas de atracamento molecular que você já utilizou?

Apêndice E - Questionário 2

Em relação à modelagem do processo principal, você considera que a modelagem está:

- Correta
 Incorreta

Se você considerou a modelagem incorreta, comente o porquê.

Em relação à modelagem do subprocesso Selecionar Moléculas Interagentes, você considera que a modelagem está:

Correta

Incorreta

Se você considerou a modelagem incorreta, comente o porquê.

Em relação à modelagem do subprocesso Preparar o Ligante, você considera que a modelagem está:

Correta

Incorreta

Se você considerou a modelagem incorreta, comente o porquê.

Em relação à modelagem do subprocesso Preparar o Receptor, você considera que a modelagem está:

Correta

Incorreta

Se você considerou a modelagem incorreta, comente o porquê.

Em relação à modelagem do subprocesso Adicionar Cofatores, você considera que a modelagem está:

Correta

Incorreta

Se você considerou a modelagem incorreta, comente o porquê.

Em relação à modelagem do subprocesso Determinar Método de Busca do Sistema, você considera que a modelagem está:

- Correta
 Incorreta

Se você considerou a modelagem incorreta, comente o porquê.

Em relação à modelagem do subprocesso Determinar Função de Avaliação do Sistema, você considera que a modelagem está:

- Correta
 Incorreta

Se você considerou a modelagem incorreta, comente o porquê.

Em relação à modelagem do subprocesso Determinar Sítio de Ligação, você considera que a modelagem está:

- Correta
 Incorreta

Se você considerou a modelagem incorreta, comente o porquê.

Qual sua impressão geral sobre todas as modelagens acima em relação à compreensibilidade?

Fácil compreensão Dificil compreensão