

5

Introdução aos Mecanismos de Flexibilização

Este capítulo tem por objetivo introduzir os conceitos básicos utilizados ao longo deste trabalho, bem como as definições e abordagens utilizadas para a flexibilização da execução de instâncias de processos em sistemas de gerência de workflow.

A Seção 5.1 apresenta a motivação deste trabalho sobre flexibilização de instâncias de workflow. A Seção 5.2 introduz os três conceitos principais relacionados a este trabalho, quais sejam, processo, instância e recurso. A Seção 5.3 apresenta conceitos importantes associados ao mecanismo de tratamento de exceção proposto. A Seção 5.4 introduz algumas questões fundamentais a respeito da flexibilização, as quais serão apresentadas em detalhes nos próximos capítulos. A Seção 5.5 apresenta um resumo deste capítulo.

A partir deste capítulo, em função do uso da linguagem OWL-S, o termo “workflow” não mais será utilizado. Este termo será substituído pelo termo “processo”, em conformidade com a terminologia de OWL-S.

5.1

Motivação

Considere o problema de limpeza de áreas costeiras afetadas por derramamento de óleo. Considere que a solução deste problema envolva a execução de três subprocessos, quais sejam, nesta ordem:

- determinar o tipo de óleo derramado;
- determinar as áreas costeiras afetadas pelo derramamento;
- limpar as áreas afetadas.

Considere, ainda, que o óleo derramado pode ser de cinco tipos distintos, variando de óleo tipo *I* a óleo tipo *V*. O tipo de óleo determina qual o procedimento de limpeza a ser utilizado, tendo por base o impacto ambiental causado pelo procedimento.

Nesse exemplo, a flexibilização da execução poderia ocorrer de três formas distintas, visando:

- encontrar valor default para o tipo de óleo, dado que o processo que encontra o tipo de óleo derramado está demorando muito para terminar. Neste caso, pode ser necessária posterior verificação de consistência, para garantir que o valor assumido tenha sido o mesmo valor encontrado de fato para o tipo de óleo;
- encontrar melhor procedimento de limpeza das áreas costeiras, levando em consideração o tipo de óleo derramado e, portanto, o impacto ambiental causado por cada procedimento;
- escolher processos alternativos de limpeza, caso não seja possível executar o melhor processo, ou seja, aquele com menor impacto ambiental declarado.

Esse exemplo é melhor explorado no Apêndice C.

5.2 Conceitos Preliminares

Processo, instância e recurso são conceitos primordiais nesta tese. As subseções seguintes apresentam estes conceitos.

5.2.1 Processos e Instâncias

De acordo com o glossário da WfMC [130], uma *definição de um processo*, ou simplesmente um *processo*, é “uma representação do que se pretende executar, sendo composto de atividades manuais ou automatizadas”. Já uma *instância de um processo*, ou simplesmente uma *instância*, é “uma representação do que de fato está sendo executado, sendo composto de instâncias de atividades, que podem incluir atividades alocadas para participantes (humanos), ou execuções de aplicações (computadorizadas)”. Nesta tese, usaremos preferencialmente os termos *processo* e *instância*.

Um processo pode ser atômico ou composto e, ortogonalmente, abstrato ou concreto. Um processo p é atômico se é indivisível. Por outro lado, um processo p é composto se for definido a partir de dois ou mais processos, chamados de subprocessos de p ou processos componentes de p .

A definição de um processo composto p utiliza construtores de controle, dentre eles *sequence*, *split* e *if-then-else*, que determinam a forma na qual os subprocessos estão organizados em p e se relacionam uns com os outros.

A Figura 5.1 apresenta 3 processos compostos, quais sejam a , b e e , e 4 processos atômicos, c , d , f e g . Os processos a e b são definidos por meio do construtor de controle *sequence*, enquanto o processo e é definido pelo construtor *split*. Esta figura aparece no Capítulo 7 para exemplificar como uma instância de processo composto é executada segundo o mecanismo de tratamento de exceção.

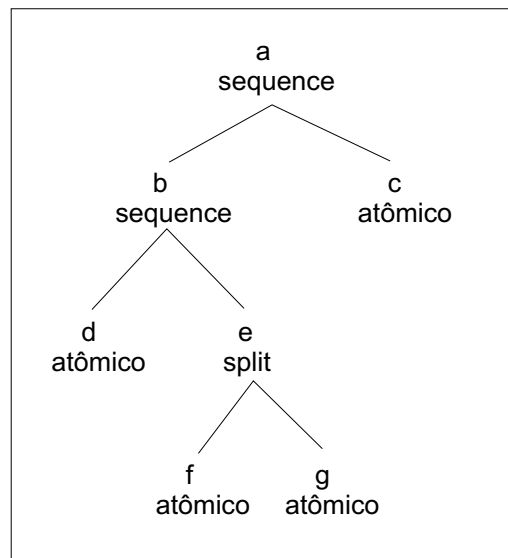


Figura 5.1: Hierarquia de processos de exemplo.

De acordo com a hierarquia apresentada, dizemos que b , c , d , e , f e g são subprocessos de a , mas que apenas b e c são subprocessos diretos de a . Analogamente, a é superprocesso de b , c , d , e , f e g . A única exceção para dizer que um processo é componente (ou subprocesso) de um processo p composto é para o caso de um processo composto através do construtor *Split*, como será visto no Capítulo 7.

Denominamos ainda P a instância gerada a partir da execução do processo p . Portanto, de acordo com a Figura 5.2, dizemos que A é a superinstância direta de B e C e que, desta forma, B e C são subinstâncias de A .

O fluxo de controle de p é determinado pelos construtores utilizados para especificar p e pelas pré-condições dos seus subprocessos. A semântica dos construtores e das pré-condições será discutida em detalhe na Seção 7.2 e nas seções subsequentes.

Brevemente, uma pré-condição determina sob qual situação um processo pode ser executado. Uma pré-condição de um processo pode dizer

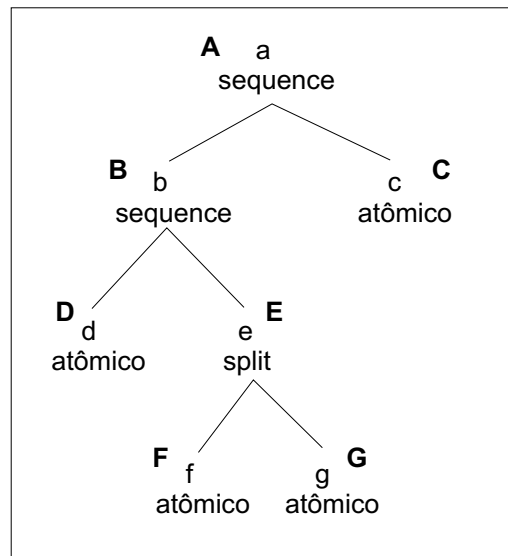


Figura 5.2: Hierarquia de processos e instâncias de exemplo.

respeito a um ou mais de seus parâmetros. Além disso, assumimos que são pré-condições implícitas a disponibilidade dos recursos necessários à execução de um processo e o conhecimento dos valores de seus parâmetros de entrada.

Como não existe definição exata com relação ao fluxo de dados entre processos em OWL-S, assumimos que, se um processo p recebe valores de entrada provenientes de parâmetros de um outro processo p' , então p e p' possuem um superprocesso em comum, ou seja, pertencem à mesma hierarquia de processos.

Ortogonalmente ao conceito de processo atômico e processo composto, um processo pode ser abstrato, o que significa que não pode ser diretamente executado, porque não tem implementação, ou concreto, quando possui implementação associada. Um ou mais processos concretos, por definição, podem estar associados a um processo abstrato, definindo como o implementam. Vale ressaltar que um processo abstrato por si só não gera instâncias, já que a execução é sempre relativa ao processo concreto.

A definição de um processo composto p contém, além da especificação do fluxo de controle, por meio dos construtores de controle, a especificação do fluxo de dados, ou seja, da forma com que os subprocessos componentes trocam dados entre si.

5.2.2 Recursos

Um recurso no sistema de gerência de workflow pode ser a representação de algo físico (um carro, uma bola, uma pá, por exemplo) ou de algo lógico (um arquivo, um software, por exemplo).

Analogamente a processos, recursos também podem ser abstratos ou concretos. Um recurso abstrato é como um tipo de recurso, por exemplo, um carro. Um recurso concreto é um recurso que pode ser diretamente utilizado e que é identificado unicamente. Por exemplo, o carro Gol com a placa *NNN2222* é um recurso concreto, que por sua vez pode estar associado ao recurso abstrato carro. Detalhes de como recursos abstratos e concretos são modelados podem ser encontrados na Seção 6.4.2.

5.3 Introdução à Flexibilização

Este trabalho define um mecanismo de tratamento de exceção para a flexibilização permitindo que:

- a completa modelagem de um processo seja adiada para o tempo de execução, possibilitando que a definição do processo contenha referência a subprocessos ou recursos abstratos, para posterior concretização;
- a execução continue na presença de informação incompleta, ou seja, quando não estão disponíveis todos os valores de parâmetros necessários à execução, e também que continue quando os recursos necessários estão indisponíveis ou quando alguma exceção é levantada.

Esse mecanismo de tratamento de exceção utiliza ontologias, conforme descrito no Capítulo 6, para:

- realizar concretização de processos e recursos;
- computar valores default para parâmetros, quando estes valores são desconhecidos ou não foram gerados em um tempo pré-determinado para o fim da execução da instância do processo;
- encontrar processos e recursos alternativos, que possam ser utilizados quando o processo ou o recurso original não pode ser utilizado;
- encontrar processos que tratem de determinadas exceções levantadas por instâncias de processos.

O conceito de exceção é de fundamental importância para o funcionamento do mecanismo de tratamento de exceção para a flexibilização da execução de processos. Consideramos uma exceção uma situação não prevista na definição estática do processo e que, anteriormente definida, possa ser tratada pelo mecanismo de tratamento de exceção. As exceções são definidas de forma hierárquica ou aninhada, de modo que uma exceção no segundo nível da hierarquia, por exemplo, apenas pode ocorrer depois de uma exceção já ter sido levantada no primeiro nível e não ter sido resolvida.

A Figura 5.3 apresenta um esquema simplificado da hierarquia das exceções definidas neste trabalho.

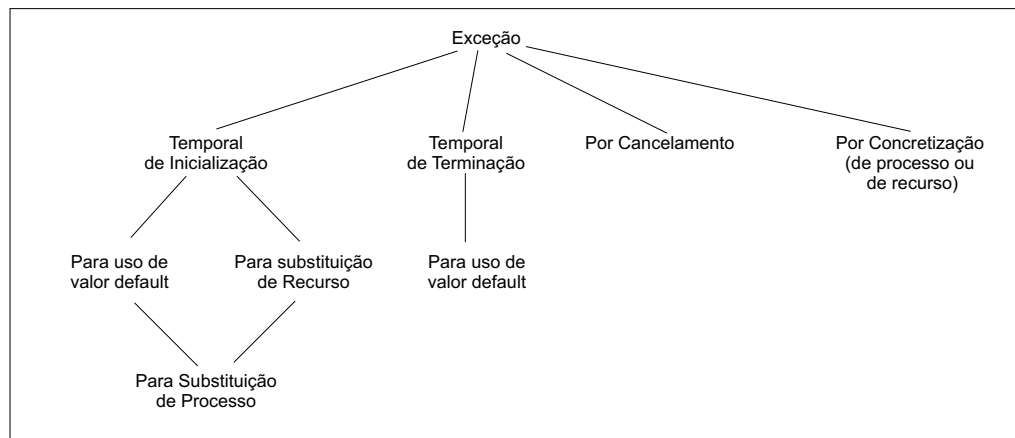


Figura 5.3: Hierarquia das exceções definidas.

No primeiro nível da hierarquia de exceções estão três exceções:

- *temporal de inicialização*: é levantada quando uma instância não consegue iniciar sua execução, porque não são conhecidos todos os seus parâmetros de entrada ou porque não estão disponíveis todos os recursos dos quais necessita. Neste caso, o tratamento da exceção corresponde a executar um processo previamente definido que possa resolver o problema de inicialização;
- *temporal de terminação*: é levantada quando uma instância não consegue terminar sua execução, não gerando os valores dos parâmetros de saída definidos no fluxo de dados do processo correspondente. Neste caso, o tratamento da exceção corresponde a executar um processo previamente definido que possa resolver o problema de terminação;
- *para concretização*: é levantada pela instância relativa ao superprocesso p do processo abstrato ou do processo concreto que possui a ele associado um ou mais recursos abstratos. Neste caso, o

tratamento da exceção corresponde a encontrar um processo concreto para o processo abstrato alcançado no fluxo de controle da instância P de p , ou a encontrar recursos concretos disponíveis relativos aos recursos abstratos encontrados;

- *por cancelamento*: é levantada quando uma instância de processo atômico é abortada. Neste caso, o tratamento da exceção corresponde em desfazer os efeitos da instância abortada.

Uma exceção do tipo temporal de inicialização indica que a instância alcançou o tempo previamente definido para ela iniciar sua execução. Neste caso, o tempo para iniciar a execução pode ter sido alcançado pela instância por dois motivos distintos. Independentemente de qual tenha sido o motivo, a máquina de execução procurará por processos para tratamento de exceção temporal de inicialização previamente definidos para o processo cuja instância sofreu a exceção. Caso nenhum processo deste tipo tenha sido encontrado, uma nova exceção, agora de segundo nível, é levantada, dado que a execução não pode ser iniciada. Deste fato surgem duas novas exceções de segundo nível, subordinadas à exceção temporal de inicialização:

- exceção temporal de inicialização para uso de valor default: é levantada quando a exceção temporal de inicialização não pode ser tratada e uma vez que o motivo que gerou esta exceção foi o fato da máquina de execução não conhecer os valores de todos os parâmetros de entrada da instância em execução. Neste caso a exceção é tratada pela busca de valores default para todos aqueles parâmetros que têm seus valores desconhecidos;
- exceção temporal de inicialização para substituição de recurso: é levantada quando a exceção temporal de inicialização não pode ser tratada e uma vez que o motivo que gerou esta exceção foi o fato de nem todos os recursos necessários à execução estarem disponíveis. Neste caso, a exceção é tratada pela busca de recursos alternativos, semanticamente próximos, àqueles que não puderam ser utilizados.

Uma exceção do tipo temporal de terminação indica que a instância alcançou o tempo previamente definido para ela finalizar sua execução, sem ter gerado todos os valores dos parâmetros de saída definidos. Neste caso, surge uma nova exceção de segundo nível, agora subordinada à exceção temporal de terminação, a exceção temporal de terminação para o uso de valor default. Esta exceção é tratada por meio da busca de valores default

para todos aqueles parâmetros de saída para os quais os valores não foram gerados.

A exceção para concretização é levantada diante de duas situações: (i) foi encontrado um processo abstrato na definição do seu superprocesso p em execução; ou (ii) o subprocesso de p a ser executado contém referências a recursos abstratos. No primeiro caso, se um processo concreto correspondente não pode ser encontrado, mesmo se a partir de processos abstratos semanticamente próximos (veja definição na seção seguinte) ao processo abstrato original, a superinstância P do superprocesso p do processo abstrato é abortada, porque não pode continuar sua execução.

Por outro lado, se um processo concreto relativo ao abstrato encontrado pode ser inicializado, a execução da superinstância P continua normalmente. Observe, neste caso, que ainda existe a possibilidade do processo concreto referenciar recursos abstratos e, desta forma, torna-se necessário que a máquina de execução primeiro encontre os recursos concretos referentes aos abstratos encontrados para que depois possa de fato inicializar a instância do processo concreto.

Quando a exceção por concretização é levantada porque o subprocesso de p a ser executado na instância P em execução contém referências a recursos abstratos, se recursos concretos correspondentes não podem ser utilizados, a instância P é abortada.

No Apêndice A será mostrado que, quando não são encontrados objetos (processos ou recursos) concretos relativos a um objeto abstrato (processos ou recursos, respectivamente), um outro objeto abstrato semanticamente próximo é buscado para que os seus correspondentes concretos sejam buscados.

As exceções de inicialização, tanto devido a parâmetro quanto devido a recurso, ainda podem gerar uma exceção em terceiro nível, que chamamos exceção para substituição de processo. Esta exceção é levantada quando a estas exceções de inicialização não puderam ser tratadas, ou seja, quando a máquina de execução não conseguiu valores default para todos os parâmetros que tinham seus valores desconhecidos, ou quando a máquina não encontrou recursos para substituir aqueles que não estavam disponíveis. Neste caso, o tratamento da exceção para substituição consiste em encontrar processos alternativos que possam ser executados, no lugar dos processos cujas instâncias sofreram a exceção por inicialização.

A exceção por cancelamento é levantada quando uma instância de processo atômico é abortada. Para tratar esta exceção, é executado um processo associado ao processo cuja instância levantou a exceção para

desfazer os efeitos das mesmas. Esta nova instância, por definição, não pode sofrer flexibilização. Como será apresentado nos capítulos seguintes, uma instância de processo composto é abortada ou tem seus efeitos desfeitos de acordo com as demais subinstâncias de sua hierarquia.

Dizemos, ainda, que uma instância P' é de flexibilização se ela está sendo executada para tratar um exceção temporal, uma exceção para concretização ou uma exceção para substituição levantada por uma instância P .

5.4 Possibilitando a Flexibilização

A flexibilização da execução de processos é garantida por meio do tratamento das exceções levantadas durante a execução das instâncias na máquina de execução.

Além das exceções, o conceito de proximidade semântica precisa ser definido.

Para se realizar a concretização, para tratar da exceção por aborto ou para tratar as exceções temporais e suas descendentes, é levado em consideração um valor que chamamos de *proximidade semântica*, definido entre um par de objetos, sejam eles processos ou recursos. Por exemplo, é definido um valor para o relacionamento entre um processo p e um processo p' , no qual p' pode representar o processo que trata uma exceção temporal de inicialização levantada por uma instância P de p .

De acordo com a proximidade semântica, a máquina de execução define o melhor processo a ser escolhido, de acordo com a situação específica, por exemplo, no caso de uma concretização de processo. Medimos então o grau de relacionamento entre dois processos de acordo com o valor de proximidade semântica atribuído ao relacionamento entre eles.

O valor da proximidade semântica entre dois objetos (processos/recursos) pode ser determinado diretamente na ontologia, ou pode ser determinado de modo dinâmico, de acordo com regras que denominamos *regras de proximidade semântica*.

Além disso, é importante notar que quando um processo é escolhido para ser instanciado e tratar da exceção temporal gerada por uma determinada instância P , a instância gerada P' , por definição, substitui a instância P . Neste caso, como p de P pode definir um fluxo de dados qualquer, é necessário que haja um mapeamento entre os parâmetros de p e os parâmetros de p' .

Ainda, no caso de processos abstratos, seus recursos abstratos também precisam ser mapeados em recursos abstratos do processo concreto escolhido no momento da concretização.

Os valores default utilizados pelo mecanismo para parâmetros de instâncias de processo que tenham seus valores desconhecidos ou que não tenham ainda valores, podem ser definidos, de forma análoga ao valor de proximidade semântica, de forma estática, rígida na definição do processo, ou de forma dinâmica. Neste último caso, *regras de suposição* podem ser definidas para que, levando em consideração por exemplo o estado da computação, possam determinar dinamicamente o valor default do parâmetro associado.

É importante deixar claro que as instâncias de processo executadas segundo a abordagem de flexibilização proposta possuem um comportamento transacional. Por comportamento transacional, entendemos a propriedade garantindo que ou todas as ações de uma instância terminam corretamente, ou todas são abandonadas e identificadas como tal.

Assumimos, ainda, que qualquer instância de flexibilização P' também pode sofrer flexibilização, exceto instâncias que estão desfazendo os efeitos de uma instância de processo atômico abortada.

Outro ponto importante é que, com a flexibilização, as subinstâncias de uma determinada instância P não são definidas diretamente pela estrutura de controle do processo p de P , mas são sim aquelas que estão de fato executando sob o “controle” de P , sem terem sofrido flexibilização. Instâncias que estavam realizando “undo” de instâncias atômicas abortadas, no entanto, não possuem superinstância e não podem sofrer flexibilização.

A flexibilização, dessa forma, força a geração dinâmica de uma árvore de instâncias, em oposição à árvore de processos estaticamente definida na definição de um processo, a partir da qual as instâncias são geradas.

Cabe ainda lembrar que, durante a busca de processos para o tratamento das exceções levantadas por uma determinada instância P , aplica-se um modelo de custo previamente definido, conforme será apresentado na Seção 7.6.4. Este modelo de custo visa ordenar as alternativas encontradas pelo mecanismo de tratamento de exceção, conforme critério definido.

5.5 Resumo

Este capítulo introduziu inicialmente os conceitos de processo, instância, recurso, e composição e descrição de processos, utilizados ao longo da tese.

Em seguida, apresentou a hierarquia de exceções definida como base para a atuação do mecanismo de tratamento de exceção. Segundo esta hierarquia, as exceções de primeiro nível são levantadas quando uma instância de processo não consegue inicializar sua execução no tempo previsto, quando não consegue terminar sua execução no tempo previsto, quando é encontrada uma referência para um processo ou recurso abstrato na definição do processo a ser executado e quando uma instância de processo atômico é cancelada. As demais exceções são levantadas apenas quando estas não podem ser tratadas.

Por fim, para facilitar a compreensão do resto do texto, este capítulo apresentou a definição de proximidade semântica, regras de proximidade semântica e regras de suposição.