

1 Introdução

1.1 Motivação

Sistemas de gerência de workflow, apesar de bastante antigos, receberam recentemente a atenção da comunidade científica, pela sua gama de aplicações. Desde então, diversas empresas organizadas na forma de consórcios e associações sem fins lucrativos, em conjunto com a comunidade acadêmica, têm se esforçado para a padronização do uso de sistemas de workflow.

Criado no ano de 1993 e atualmente o mais importante consórcio da área, o WfMC reúne várias companhias no objetivo de padronizar conceitos e tecnologias para workflow [59].

Além do WfMC, outros consórcios estão envolvidos nesses esforços de padronização, apesar de não serem estes consórcios exclusivamente vinculados ao estudo de sistemas de workflow. Podem ser citados dentre estes consórcios OASIS, WARIA, BPMI, OMG e W3C.

A associação WARIA, por exemplo, foi criada em 1992 para estudar aspectos de gerência de processos de negócio, gerência do conhecimento e comércio eletrônico e trabalha juntamente com o WfMC na busca de padrões para sistemas de gerência de workflow. O BPMI (*The Business Process Management Initiative*, ou BPMI.org) também trabalha junto com a WARIA no sentido de desenvolver a gerência de processos de negócio através do estabelecimento de padrões para o projeto, o desenvolvimento, a execução, a manutenção e a otimização de processos de negócio. Dentre as diferentes contribuições destes consórcios, destacam-se novas linguagens de definição de workflows e protocolos de coordenação.

Sistemas de workflow possuem uma extensa lista de requisitos. Dentre estes requisitos, podem ser citados execução distribuída, cooperação, coordenação e sincronização, os quais modelam a maneira com que usuários cooperam para a realização de uma determinada tarefa [5] [6] [51].

Esta tese endereça um outro requisito, aqui chamado de *execução flexível*. Geralmente, sistemas de gerência de workflow interpretam rigidamente a definição do workflow, não permitindo qualquer tipo de desvio. No entanto, existem situações reais em que usuários devem poder desviar do fluxo estático pré-definido na definição do workflow por diversas razões, incluindo a falta de informação e a indisponibilidade dos recursos necessários à execução.

Por exemplo, considere um workflow w cujo objetivo é tratar as conseqüências de um incêndio em uma área florestal. Considere, ainda que, segundo definido em w , seja necessário contactar o corpo de bombeiros mais próximo ao local do desastre. No entanto, considere a impossibilidade de contactar tal corpo de bombeiros, por exemplo, pelo fato da linha telefônica estar ocupada.

Se o tempo de espera para chamar o corpo de bombeiros for alto, pode-se configurar um desastre ainda maior. Neste caso, por exemplo, deve ser melhor contactar outro corpo de bombeiros do que aguardar até que o corpo de bombeiros definido em w seja contactado, assumindo-se que exista uma forma de comunicação entre os corpos de bombeiros.

1.2 Objetivos

Para alcançar execução flexível, esta tese propõe um mecanismo de tratamento de exceção para a flexibilização, permitindo que:

- a completa modelagem de um workflow seja adiada para o tempo de execução, possibilitando que a definição do workflow contenha referência a subworkflows ou recursos abstratos;
- a execução continue na presença de informação incompleta, ou seja, quando não estão disponíveis todos os valores de parâmetros necessários à execução. Neste caso, são utilizados valores default para os parâmetros cujos valores são ainda desconhecidos;
- a execução continue mesmo quando os recursos necessários estão indisponíveis, pelo uso de recursos ou até mesmo workflows alternativos. Se algum recurso necessário à execução encontra-se alocado para a utilização por outra instância de workflow, então recursos semanticamente próximos podem ser utilizados. Se estes não são encontrados, o workflow como um todo pode ser substituído

por outro workflow semanticamente próximo e para o qual estejam disponíveis todos os recursos.

O mecanismo de tratamento de exceção proposto pode ser entendido como uma estratégia de substituição de componentes, aplicada à execução de workflows, onde subworkflows e recursos fazem o papel dos componentes [9].

O mecanismo proposto utiliza informação semântica adicional à descrição do workflow, formalizada por meio de uma ontologia de processos e recursos, chamada aqui de *ontologia pr*, para:

- realizar a substituição em tempo de execução de objetos (recursos e workflows) abstratos por objetos concretos;
- computar valores default para parâmetros de workflows, quando seus valores são desconhecidos;
- encontrar workflows e recursos alternativos, quando existe indisponibilidade dos recursos necessários à execução;
- encontrar workflows que tratem de outras exceções levantadas por uma instância de workflow em execução.

É importante frisar que, diferentemente de algumas abordagens encontradas na literatura, o mecanismo de tratamento de exceção para a flexibilização proposto neste trabalho não visa permitir a interferência direta de usuários na execução de workflows. De fato, ele usa informação semântica adicional sobre workflows e recursos, e o estado corrente da execução, para sugerir alternativas ao usuário ou realizar suposições, permitindo que a execução continue quando, por outro lado, ela seria momentaneamente interrompida.

Para alcançar esse objetivo, esta tese endereça dois objetivos secundários, quais sejam, definir uma semântica formal para uma parte da linguagem OWL-S e especificar uma arquitetura distribuída para a execução de instâncias de workflow.

1.3 Contribuições

Como principais contribuições da tese, podemos citar:

- a construção de uma ontologia de processos e recursos que guia o mecanismo para a flexibilização da execução de uma instância de workflow. Esta ontologia foi criada estendendo as ontologias de processos e de recursos de OWL-S, sobretudo no que diz respeito à definição de timeouts e relacionamentos semânticos entre objetos (workflows e recursos);
- a especificação de uma semântica formal para a linguagem OWL-S, ainda não existente oficialmente na literatura, e que possibilita a fácil compreensão do funcionamento da linguagem. Esta semântica foi definida através do conceito de máquina abstrata;
- o próprio mecanismo de tratamento de exceção para a flexibilização, o qual aborda exceções sob diversas condições;
- a definição de uma arquitetura distribuída para sistemas de gerência de workflow.

1.4 Organização da Tese

Esta tese encontra-se organizada da seguinte forma.

O Capítulo 2 apresenta um resumo de diversos aspectos relativos aos sistemas de gerência de workflow, incluindo uma breve revisão da história dos sistemas de workflow, dos modelos transacionais e dos esforços de padronização do WfMC. Este capítulo inclui também questões diversas de projeto e implementação de sistemas de workflow, e discute a relação entre workflows e processos de negócio.

O Capítulo 3 contém uma discussão sobre os trabalhos relacionados ao proposto nesta tese, incluindo uma discussão sobre a evolução de sistemas de gerência de workflow, sobre as diversas linguagens de definição existentes, sobre a linguagem OWL-S utilizada para a modelagem de workflows neste trabalho, e incluindo também um resumo sobre os frameworks e protocolos para composição e coordenação de serviços, e sobre o problema de *matching* semântico de serviços.

O Capítulo 4 é introdutório e resume a linguagem OWL-S e a linguagem de regras do Jena, utilizadas neste trabalho.

O Capítulo 5 apresenta uma introdução ao mecanismo de tratamento de exceção proposto neste trabalho, incluindo definições de conceitos básicos utilizados ao longo de texto.

O Capítulo 6 descreve a ontologia de processos e recursos criada nesta tese, chamada aqui de *ontologia pr*, que é a base do mecanismo de tratamento de exceção, permitindo a flexibilização da execução. Esta ontologia foi construída utilizando as ontologias originais de processos e de recursos de OWL-S, acrescida de extensões que permitam a flexibilização.

O Capítulo 7 apresenta inicialmente uma proposta de semântica operacional para OWL-S, baseada no conceito de máquina abstrata. Em seguida, estende a semântica para acomodar a questão da flexibilização da execução, apresenta o conceito de log de execução de uma instância de workflow e a forma na qual dá-se a terminação de uma instância. A organização desse capítulo reflete a organização do Capítulo 6, tomando como ponto de partida os construtores básicos de OWL-S e, posteriormente, as extensões propostas.

O Capítulo 8 propõe uma arquitetura distribuída para um sistema de gerência de workflow, utilizando a linguagem ACME de descrição de arquiteturas. Esta arquitetura está fundamentada no conceito de máquina abstrata apresentado no Capítulo 7.

O Capítulo 9 descreve as conclusões e apresenta os trabalhos futuros.

O Apêndice A descreve o procedimento de execução de uma instância de processo na máquina abstrata estendida e na arquitetura distribuída. O Apêndice B apresenta as ontologias de processos e recursos de OWL-S estendidas, a ontologia *pr* completa e as regras associadas a uma ontologia de aplicação qualquer. O Apêndice C contém uma biblioteca *lib* de processos e recursos para um determinado domínio de aplicação, o código OWL de um processo, chamado *CoastalAreaOilCleaning*, e uma rápida discussão sobre a flexibilização deste processo.