7
Trabalhos Relacionados

Nesta seção são discutidos os trabalhos relacionados atuantes em modelagens de workflows dinâmicos, adaptativos ou workflows baseados em regras configuráveis. Muitas modelagens dividem os workflows em camadas onde o conjunto de regras é separado da modelagem do workflow abstrato.

As modelagens abaixo foram usadas como referência para o desenvolvimento do workflow deste trabalho.

7.1.
AgentWork

O *AgentWork* (Muller, 2004) apresenta os conceitos e implementação de um sistema de workflow suportando adaptações automáticas. Ele possui uma abordagem baseada em regras ECA para especificar ações adaptativas para as exceções do workflow. Estas ações invocam agentes para corrigir o caminho do workflow.

O *AgentWork* possui dois agentes atuantes em seu workflow:

- Agente de adaptação: decide qual regra será usada para corrigir o caminho do workflow. Todas as adaptações são sujeitas à confirmação manual.
- Agente de monitoração de workflow: verifica se as afirmações realizadas pelo agente de adaptação estão corretas. Caso contrário, ele induz a correção do workflow.

A máquina de regras ECA é utilizada para especificar quais eventos constituem uma falha lógica e como lidar com elas. Regras ECA apresentam quais ações devem ser removidas, adicionadas ou substituídas na instância do workflow. Uma abordagem baseada em regras é extremamente flexível, já que regras podem reagir a um evento em qualquer momento durante a execução de um workflow, sem verificar quando o evento ocorreu.

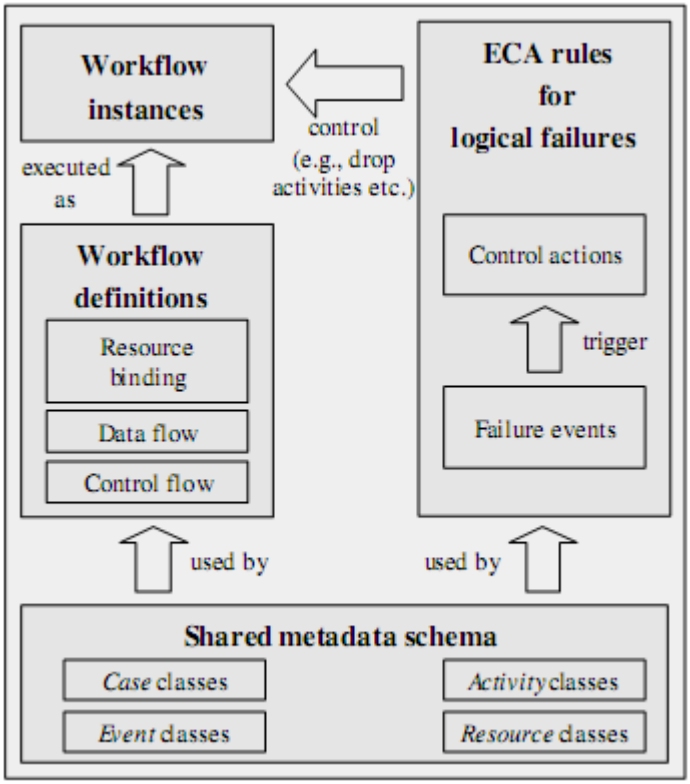

Ilustração 50 - Estrutura do AgentWork (Muller, 2004)

A Ilustração 50 apresenta a estrutura geral do *AgentWork*. O workflow é dividido entra sua definição e suas instâncias.

O conteúdo das regras ECA se localiza no esquema metadados compartilhado. Em um exemplo de aplicação, a classe *Case* representa um paciente, *Event* representa algo que possa causar uma falha no workflow, como um dado da contagem de leucócitos. *Activity* apresenta uma atividade, como a dosagem de uma droga. *Resource* representa quem executa a atividade, como um médico.

A estrutura das regras ECA foi modelada criando um framework lógico temporal com uma sintaxe orientada a objeto.

WHEN *new finding of patient P*
WITH *leukocyte count* < 1000
THEN *drop drug Etoposid for P*
VALID-TIME *during the next seven days*

Ilustração 51 - Estrutura de uma Regra (Muller, 2004)

A figura acima apresenta uma regra para administrar uma droga. A condição de entrada está definida em WHEN e WITH; THEM define qual atividade é executada; o fator temporal VALID-TIME define por quanto tempo a atividade será executada. Neste trabalho, o fator temporal não é necessário, uma vez que todas as ações são pontuais e imediatas.

O sistema verifica se existem regras que são incompatíveis. Ou seja, regras que executam ações opostas para as mesmas condições. Assim como neste projeto, o workflow utiliza uma tabela de incompatibilidade com o mapeamento das atividades. Se duas regras possuem atividades incompatíveis, então elas também são incompatíveis.

Incompatibility table for control actions (with A_1 subsuming A_2)

		1	2	3	4	5
		ca_1				
	ca_2	$drop(A_2)$	$replace(A_2, B)$	$postpone(A_2, d_2)$	$add(A_2)$	$add-repetitively(A_2, d_2)$
1	$drop(A_1)$	CP	CP[a] *ICP* if $A_1 \subset B$[b]	ICP	ICP	ICP
2	$replace(A_1, B)$	CP	CP	ICP	ICP	ICP
3	$postpone(A_1, d_1)$	CP	CP	*CP* (*ICP* only for $d_1 \neq d_2$)	CP[c]	CP[c]
4	$add(A1)$	CP	CP	CP	CP	CP
5	$add-repetitively(A_1, d_1)$	CP	CP	CP	CP	CP (ICP only for $d_1 \neq d_2$)

ICP = Incompatible, CP = Compatible.

Ilustração 52 - Mapeamento de ações incompatíveis (Muller, 2004)

O protótipo do *AgentWork* é voltado para a área médica de tratamento de câncer e muitas de suas decisões arquiteturais foram pensadas focando nesta área de atuação. Ele não permite que diferentes áreas dentro do mesmo sistema possuam workflows personalizados de maneira diferente. Não apresenta soluções para a alteração dinâmica de workflows no nível do tipo e a propagação para as instancias de workflows já em execução. (Rinderle, 2004).

7.2. KWM

KWM, ou *Knoledge-Based Workflow Model, (Lee, 1999)* é um modelo para sistemas de workflows que propõe as seguintes melhoras:

- Expressividade: deve prover a estrutura para criar um mapeamento relacional entre papeis e atores baseado no modelo da organização e em regras de negócio complexas.
- Verificação do modelo: deve prover estrutura para a análise, garantindo a corretude da especificação do workflow, incluindo verificação de inconsistência, redundância e falta de finalização
- Gerência de Mudanças: deve permitir o desenvolvimento fácil de mecanismos de propagação das alterações da estrutura organizacional e das regras de negócio.
- O modelo do workflow possui transições, regras e estados.

Um estado pode constituir tarefas, recursos, unidades organizacionais, papéis e atores. Nesta modelagem, o estado da organização é definido a partir de um conjunto de atributos do objeto organização.

As regras possuem três tipos: procedurais, de responsabilidade, e metaregras. Procedurais controlam a seqüência de tarefas, de responsabilidade condicionam atores com papeis. Metaregras combinam duas ou mais regras procedurais ou de responsabilidades

Uma regra no KWM possui uma estrutura conceitual bastante similar à deste trabalho com as seguintes informações:

- Processo: processo do workflow onde a regra atua.
- Descrição: descrição verbal da ação da regra.
- Condição: critério que precisa ser atendido para que a regra seja executada.

O KWM não permite que uma regra apresente várias soluções e nem que o usuário selecione qual deseja executar.

7.3. TriGSflow

TriGSflow (Kappel, 2000) é um framework para construção de sistemas de workflow que engloba três técnicas diferentes:

1. Modelagem em um banco de dados orientado a objetos para prover um modelo de workflow genérico, permitindo modelagem, personalização e reuso de domínios de negocio complexos.
2. Um modelo de papéis associado para desacoplar as atividades das pessoas no sistema.
3. Regras ECA usadas para coordenar, de maneira flexível, quais atividades precisam ser executadas por quais recursos.

Também possui uma máquina de regras ECA que são responsáveis por três ações no *TriGSflow*:

- Ordenação de atividades: atividades de um workflow precisam ser ordenadas de maneira que respeitem sua hierarquia temporal. Atividades novas podem ser incluídas e a ordem pode ser alterada dinamicamente durante a execução do workflow.
- Seleção de agentes: para cada atividade, deve-se selecionar um ou mais agentes para sua execução. Um exemplo de critério de

seleção de agente é: agentes que estão de férias não podem ser selecionados para executar atividades.

- Gerência de lista de tarefas: similar à ordenação de atividades, deve coordenar a lista de tarefas de um agente, uma vez que dentro de uma atividade, podem existir restrições temporais com relação aos seus passos.

O *TriGSflow* não apresenta a separação entre o workflow em um nível estrutural e em instancias. Logo, não apresenta soluções para propagar alterações da estrutura de um workflow para suas instâncias em execução.