

2

Sistemas de Gerenciamento de Workflow

O termo *workflow* consiste na utilização de elementos computacionais para auxiliar a execução de processos [2]. *Workflows* são tipicamente usados para descrever a interação entre empregados e recursos materiais e computacionais para execução desses processos. Sistemas de gerenciamento de *workflow* permitem a automação de processos que envolvem atividades humanas e computacionais, incluindo interação com ferramentas e aplicações auxiliares [2]. Nesses sistemas os processos são modelados em *workflows* que permitem a definição da sequência das atividades realizadas durante um processo, bem como os responsáveis e recursos de cada uma. Além de auxiliar a descrição do modelo, sistemas de gerenciamento de *workflow* são usados para realizar os processos, acessando bases de dados ou aplicações auxiliares quando necessário. Com isso, atividades complexas e repetitivas podem ser feitas com menor ou nenhuma intervenção humana, reduzindo a suscetibilidade dos processos a erros. Dessa forma, a realização dos processos pode ser simplificada e unificada, permitindo que aspectos produtivos e administrativos sejam planejados em detalhe.

Sistemas de gerenciamento de *workflow* são caracterizados por darem suporte a três funções principais [2]: construção, controle da execução e interação. Cada uma dessas funções pode ser desempenhada por uma ou mais ferramentas. Para a função de construção existem ferramentas para definição de *workflows* e para modelagem de processos. A função de controle de execução é desempenhada pelo motor do sistema, ele que instancia os processos e sequencia as ações a serem realizadas. Finalmente, a função de interação consiste em gerenciar a interação de ações com usuários e outras aplicações para realização das atividades descritas.

Em um mesmo sistema pode ser necessário usar ferramentas de diferentes fabricantes. Sendo assim, para padronizar a interação entre ferramentas que realizem diferentes funções foi formado o WFMC (Workflow Management Coalition) [2]. A padronização consiste no estabelecimento de interfaces para interação entre diferentes partes do gerenciamento de *workflows*. Dessa forma o gerenciamento de *workflows* fica dividido em: modelagem de processos, execução de processos, interação com usuário, interação com aplicações auxiliares,

monitoração da execução dos processos e interação com outros sistemas de gerenciamento de *workflow*. Com isso, um fabricante pode se dedicar ao desenvolvimento de uma ferramenta que se responsabiliza por apenas uma parte do gerenciamento de *workflow* e, por outro lado, uma empresa não precisa comprar todas as ferramentas de um mesmo fabricante.

A interação entre as ferramentas pode ser analisada a partir da descrição dos passos de utilização de um sistema de gerenciamento de *workflow*. A entrada do sistema é feita a partir da modelagem do processo ou apenas a definição do *workflow*. Uma vez que o *workflow* é descrito, cabe ao motor do sistema gerenciar a instanciação e execução dos processos.

Durante a execução de um *workflow*, o motor interpreta as ações definidas e provê os recursos necessários para a sua conclusão, sejam esses recursos aplicações externas ou a realização de tarefas por empregados da empresa. Uma interface específica pode ser usada para listar as tarefas pendentes de um empregado ou simplesmente informar o empregado da sua tarefa e aguardar a sua conclusão. Nesse caso é preciso treinar o usuário para acessar as tarefas listadas, realizá-las e informar o sistema da conclusão da tarefa e de possíveis resultados. Por outro lado, para interagir com outros sistemas, a interação implica na troca de dados por meio de uma interface pré-estabelecida. Essa interface pode ser um protocolo de comunicação comum ou uma ferramenta intermediária que saiba encaminhar pedidos e resultados entre o motor e o sistema em questão.

Durante a operação de um sistema de gerenciamento de *workflow*, pode ser necessário acompanhar o andamento de um processo ou o histórico de execuções de determinado processo para fins administrativos, como auditoria ou um relatório de produção. Para esses casos o sistema de gerenciamento de *workflow* precisa de uma ferramenta de monitoração. Essa ferramenta disponibiliza então, as informações da execução dos processos em tempo real ou por meio de históricos de execução dos processos.

As aplicações usadas em um sistema de gerenciamento de *workflow* podem ser classificadas em clientes de modelagem, clientes de execução ou servidor de execução. Um exemplo de interações possíveis entre essas aplicações pode ser visto na figura 2.1. Clientes de modelagem permitem tanto a modelagem dos processos quanto a descrição desses modelos em *workflows*. A modelagem e descrição são tipicamente feitas por meio de programação visual em que ações são encadeadas em grafos. Além disso, são determinadas as condições para realização das ações e descritos os recursos necessários para sua realização. Essas condições se baseiam no valor dos dados do *workflow* e determinam se uma sequência de ações será realizada, enquanto o encadeamento

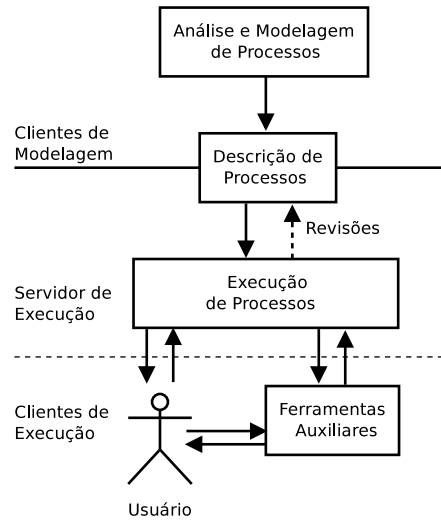


Figura 2.1: Elementos de um Sistema de Gerenciamento de *Workflow*

indica a ordem em que as ações devem ser realizadas, descrevendo o fluxo da execução do processo. Além do fluxo de execução, é possível determinar o fluxo de dados entre as ações. Esse fluxo de dados define quais variáveis e resultados serão usados nas ações do *workflow*.

O servidor de execução possui a função de motor do sistema de gerenciamento de *workflow*. Ele é responsável pela a realização das ações e avaliação de condições de transição, determinando a execução do processo. Finalmente, clientes de execução são utilizados para acompanhar a execução dos processos. Esses clientes se comunicam com o motor e obtêm informações do funcionamento dos *workflows*, podendo ser usados para, por exemplo, listar tarefas pendentes de um empregado, monitorar a execução de processos, gerenciar a execução de processos ou gerenciar a comunicação com outros sistemas.

Os processos empresariais descritos podem ser classificados em administrativos ou produtivos. Processos administrativos estão associados a tarefas de gerenciamento da empresa como gerenciamento de estoque, faturamento de serviços, recursos humanos e folha de pagamento. Já os processos produtivos estão associados à fabricação de produtos ou a prestação de serviços. Esses dois tipos de processos possuem recursos bem característicos. Enquanto processos administrativos dependem de dados, documentos e empregados, processos produtivos dependem de máquinas, aplicações especialistas e matéria-prima.

A execução do processo descrito no *workflow* é o principal elemento do sistema de gerenciamento de *workflow*. Durante a operação do sistema, um *workflow* pode estar em um dos seguintes estados [2]:

instanciado instância do processo foi criada assim como dados associados,

mas as condições para execução ainda não foram cumpridas;

executando a execução do processo foi iniciada, mas nenhuma de suas ações foi ativada;

ativo uma ou mais ações do processo foram iniciadas;

suspenso o processo foi interrompido e suas ações não vão de ser iniciadas até que retorne ao estado executando;

abortado a execução do processo foi interrompida antes do seu término normal e operações de registro ou de recuperação de dados são executadas e suas instâncias removidas; e

concluído o processo atingiu suas condições de conclusão, as operações de registro são realizadas e a instância do processo é removida.

Uma vez que o *workflow* está executando, o motor determina quais ações podem ser iniciadas, ativando-as. Os estados das ações de um *workflow* são [2]:

inativa o processo já foi instanciado, mas as condições para realização da ação ainda não foram cumpridas;

ativa a ação está sendo executada por um elemento do sistema de gerenciamento de *workflow*;

suspensa a ação foi interrompida e não será retomada até que seja desativada e receba uma novo dado de entrada; e

completa a ação foi concluída e suas condições de transição são ativadas.

Ao iniciar o seu funcionamento o motor do sistema não executa nenhum *workflow*. É necessário um comando para que os *workflows* sejam instanciados. Uma vez que os *workflows* são instanciados, o motor inicia a avaliação das condições para sua execução. Quando essas condições são atendidas, a primeira ação do *workflow* é iniciada e as demais ações podem ser ativadas. A conclusão dessa ação irá ativar as ações seguintes de acordo com suas condições de transição e, assim em diante, até que a última ação seja concluída e o *workflow* atinja suas condições de conclusão. Enquanto possui *workflows* instanciados, o motor atua em ciclos de verificação para determinar quais ações já atingiram suas condições de ativação. Esse ciclos irão controlar o fluxo da execução dos processos, realizando as ações na ordem em que foram descritas. Essa ordem pode não ser apenas linear, ou seja, ações podem ser realizadas em paralelo ou determinar bifurcações no fluxo da execução, de acordo com a construção definida para o *workflow*. Na próxima seção iremos apresentar os padrões para essas construções e exemplos de casos em que são usados.

2.1

Padrões de Construção de Workflows

Ao executar o *workflow*, o sistema deve ser capaz de interpretar o modelo descrito de forma a realizar as ações na ordem em que as atividades modeladas seriam realizadas no processo modelado. Sendo assim, o *workflow* precisa representar de forma mais próxima o processo modelado. Para isso, clientes de modelagem disponibilizam elementos para descrição de modelos que permitem construções equivalentes às do processo modelado. Observando os sistemas de gerenciamento de *workflow* disponíveis é possível identificar padrões para essas construções [3].

Os padrões de construção básicos devem descrever a **sequência das ações**, **condições para ativação**, **convergência de sequência**, **execução em paralelo** e **sincronização de ações**. O padrão de **sequência das ações** é determinado pela conexão do término de uma ação com o início da próxima. Além desse padrão, é possível usar o padrão de **condição para ativação** para que uma ação seja iniciada quando uma condição, indicada pelo valor de uma informação de controle ou resposta do usuário, for observada. O padrão de **convergência de sequências** indica que uma ação pode ser realizada após o término de uma de várias ações, permitindo a convergência de várias sequências de ações. Alternativamente a esses, o padrão de **execução em paralelo** permite que ações de sequências independentes entre si possam ser executadas ao mesmo tempo sem que aguardem o término de ações que não interferem na sua execução. Complementar a esse padrão está o padrão de **sincronização de ações**. Esse padrão permite que uma ação só seja executada após a conclusão de todas as ações conectadas.

Sistemas de gerenciamento de *workflow* que oferecem suporte aos padrões básico possibilitam a modelagem de processos simples. Entretanto, alguns processos podem precisar de padrões avançados para construções complexas. A seguir iremos descrever os padrões avançados para modelagem de processos em *workflows*.

O padrão de **escolha múltipla** permite descrever que, ao chegar em determinado ponto do fluxo, o motor deve verificar o valor de uma informação de controle ou resposta do usuário para determinar o início de uma ou mais ações. De forma complementar, existe o padrão de **sincronização de ações ativas** que aguarda conclusão de todas as sequências de ações conectadas. Diferente do padrão de **sincronização de ações** esse padrão não obriga que todas as sequências ações conectadas estejam executando, mas todas as que estiverem precisam ser concluídas para que a próxima ação seja realizada. Analogamente, existe o padrão de **mesclagem múltipla** que permite o encadeamento arbitrário

de ações, de forma que fluxos de execução diferentes realizem a mesma atividade seguindo sequências de ações diferentes. Em casos que uma ação deve ser iniciada apenas uma vez após o término da primeira de várias sequências de ações, existe o padrão do discriminador. Esse padrão permite que sequências de ações independentes, mas com o mesmo objetivo, sejam realizadas em paralelo e, a que for concluída mais rapidamente, inicie a ação seguinte, sem que essa seja realizada novamente quando a outra sequência de ações for concluída.

Como a conexão de ações de um *workflow* são semelhantes a de grafos de ligação direcionados, *workflows* podem descrever estruturas como ciclos e saltos. Essas construções são representadas por dois tipos de padrões: ciclos arbitrários e término implícito de processo. Padrões de ciclos arbitrários definem uma ou mais ações do *workflow* que podem ser feitas repetidamente. Entretanto, alguns sistemas só permitem o uso de ciclos estruturados que apresentam apenas um ponto de entrada e um ponto de saída. Um exemplo de ciclo estruturado pode ser visto na figura 2.2 onde a ação C é repetida para que o ciclo tenha apenas uma entrada. Os padrões de término implícito tratam da possibilidade do motor interpretar a conclusão das ações como término do *workflow*. Sistemas de gerenciamento de *workflow* que não implementam esse padrão usam elementos específicos para marcar o término do fluxo. Uma vez que o elemento de término for alcançado, as demais ações do *workflow* não poderão mais ser ativadas até que o fluxo seja instanciado novamente.

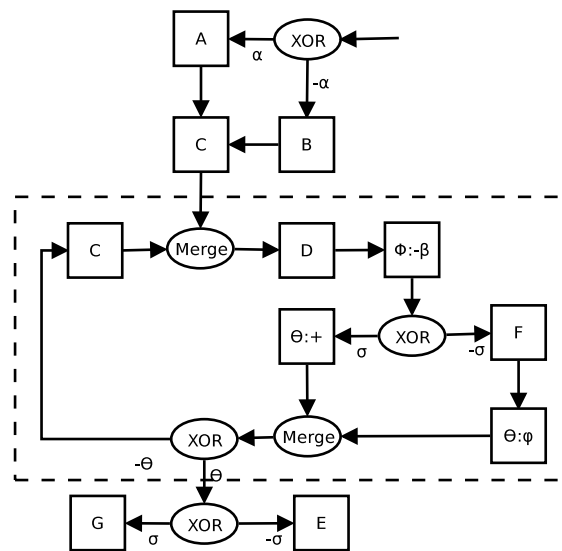


Figura 2.2: *Workflow* adaptado ao padrão de ciclos estruturados.

O padrão de múltiplas instâncias permite que a mesma sequência de ações seja instanciada para diferentes fluxos de dados de entrada. Esse padrão apresenta dois subtipos de acordo com o conhecimento prévio dos fluxos

de dados de entrada, podendo ser **pré-determinados** ou **descobertos durante a execução**. Dados descobertos durante a execução são mais flexíveis, pois não amarram o *workflow* a um conjunto de fluxos de dados específico, e permitem a reutilização do modelo de processo caso que novos fluxos de dados sejam adicionados. Por exemplo, com esse padrão é possível definir um *workflow* em que os fluxos de dados representam equipamentos da planta industrial e as ações definem um controle que deve ser realizado, de forma independente, sobre todos os equipamentos.

A execução de um *workflow* por um motor está associada a um estado que é formado por um conjunto de informações relevantes para execução, como a disponibilidade de recursos e variáveis do próprio fluxo. Esse estado pode ser usado para determinar como ações vão ser ativadas em alguns padrões avançados. Esses padrões podem ser usados em três tipos de construção: **escolha adiada**, **execução intercalada** e **baseada em marcos pré-definidos**. O padrão de **escolha adiada** permite agrupar ações de forma que só a primeira que tiver suas condições de ativação obedecidas será ativada. A ação que vai ser ativada só é determinada em tempo de execução, tornando esse padrão útil para casos em que mais de um recurso pode ser usado para a mesma atividade. Dessa forma, apenas a ação associada com o primeiro recurso disponível será iniciada. O padrão de **execução intercalada** permite a definição de um conjunto de ações que devem ser executadas exclusivamente, mas não estabelece a ordem entre elas e ação seguinte ao conjunto só será realizada quando todas forem concluídas. Esse padrão permite que ações possam ser reordenadas durante a execução de acordo com a disponibilidade de recursos, mas que todas as ações listadas tenham que ser concluídas para que a execução do *workflow* possa continuar. O último padrão baseado no estado do *workflow* é o **baseado em marcos**. Esse padrão permite que uma ação esteja ativa apenas enquanto um valor ou conjunto de valores do fluxo de dados do *workflow* corresponda a valores pré-determinados. Nesse tipo de construção é possível, por exemplo, desabilitar o cancelamento de um pedido uma vez que o pedido seja enviado.

O último padrão identificado para *workflows* é o padrão de **cancelamento**. Esse padrão permite que, de alguma forma, a realização de uma ação ou a execução de uma instância de um *workflow* seja interrompida ou desfeita. Construções desse tipo são usadas para evitar que uma ação seja realizada em uma condição indesejada ou após a conclusão de uma ação equivalente. Esse tipo de padrão não necessariamente tem um suporte gráfico, mas pode ser obtido por meio de uma interface de programação que habilite a remoção de uma ação da base de dados. Sistemas que não possuam o cancelamento de ações podem agrupar ações de cancelamento com ações críticas utilizando o

padrão de escolha adiada. Uma vez que a ação de cancelamento for ativada a escolha adiada impedirá que a ação crítica seja iniciada.

2.2

Limitações

A utilização de sistemas de gerenciamento de *workflow* viabiliza a otimização e coordenação de processos. Entretanto, a realização desses processos acabou se tornando dependente da infra-estrutura computacional do sistema utilizado. Esses sistemas foram inicialmente projetados para serem usados em redes pequenas com um conjunto pequeno de processos e de usuários [1]. Por esse motivo, o armazenamento das ações, gerenciamento de listas de tarefas de usuários, obtenção de dados relevantes para execução e geração de relatórios foram originalmente projetados para bases de dados locais e centralizadas. Como apenas um elemento distribui as ações e indica o que está sendo realizado, essa característica simplifica a cooperação entre usuários e o compartilhamento de tarefas. Entretanto, essa característica acaba deixando sistemas de gerenciamento de *workflow* menos escaláveis e mais suscetíveis a gargalos durante a execução, pois todas as requisições acabam sendo direcionadas para uma mesma base de dados [1].

Sistemas usados para execução de processos críticos não podem ficar inoperantes por muito tempo, seja por falha ou para manutenção. Como uma falha em uma base de dados centralizada implica na falha do sistema como um todo e na interrupção dos processos, esses sistemas devem estar associados a bases de dados que possam se recuperar ou ser substituídas rapidamente. Uma forma de obter isso é manter uma cópia reserva da base de dados do sistema que seja utilizada quando uma falha na base de dados original for identificada. Nesse caso, para manter a cópia e a base de dados original consistentes é necessário realizar comandos adicionais que, por sua vez, implicam em uma sobrecarga e impacto no desempenho do sistema.

Outra limitação conhecida dos sistemas de gerenciamento de *workflow* está associada à forma como os sistemas lidam com erros encontrados durante a realização de ações [4]. Quando ocorre a falha de uma aplicação auxiliar, uma falha na comunicação, o vencimento de um prazo ou a falta da resposta de um usuário do sistema, a execução do processo é interrompida. Em alguns desses casos o processo não pode continuar e a interrupção é inevitável. Entretanto, podem existir ações que contornam a falha e por meio de caminhos ou recursos alternativos. Apesar de existirem construções para *workflows* que permitam algum nível de contorno, os sistemas tipicamente não disponibilizam ao usuário ferramentas visuais dedicadas para descrever como lidar com esses erros.

Sistemas que conseguem superar essas limitações são capazes de atender a demanda em relação à segurança e confiabilidade da realização dos processos.

2.3

Exemplos de Sistema de Gerenciamento de Workflow

Existem diversos sistemas de gerenciamento de *workflow* disponíveis no mercado. Esses sistemas diferem no foco de uso e no nível de interação que dispõem com outros sistemas. A seguir descreveremos alguns exemplos desses sistemas e suas características.

O SAP ERP tem o foco empresarial e é um dos mais conhecidos para o gerenciamento de processos administrativos. Esse sistema tem três interfaces principais: apresentação, execução e base de dados. Cada uma das interfaces é responsável por uma parte do gerenciamento de *workflows*. Além disso, esse sistema possui soluções específicas para gerenciamento financeiro, pessoal, de operações e de serviços corporativos.

Com o foco em pesquisa, o Exotica/FMQM [5] foi construído buscando resolver o problema da continuidade da execução dos processos. Esse sistema permite a distribuição das ações do *workflow* em diferentes máquinas que utilizam um meio de mensagem persistente para comunicação. Dessa forma, a falha de um membro do sistema não interrompe a execução do processo e as ações definidas para a máquina em falha serão retomadas uma vez que essa se recupere. Entretanto, nesse sistema não é possível o acompanhamento da execução de um processo específico de forma simples. Para isso, um elemento teria que centralizar as informações da distribuição das atividades ou navegar pelas máquinas do sistema e identificar o estado do processo. Na figura 2.3 é possível observar essa característica, onde o cliente de modelagem distribui o processo pelos nós disponíveis e cada nó apresenta uma tabela de instâncias do processo e uma fila de *threads* em execução.

Um outro sistema que se destaca é o Kepler [6]. Esse sistema possui o código aberto e tem como foco a cooperação no meio científico. O objetivo é disponibilizar uma representação executável dos passos necessários para gerar resultados de experimentos científicos. Projetos que utilizam esse sistema vão de gerenciamento de dados gerados por redes de sensores ao acompanhamento e análise da população de espécies selvagens. Na figura 2.4 é possível observar o ambiente de desenvolvimento e um *workflow* no Kepler. O *workflow* representado consiste em ações para o operações com os valores de entrada e para exibição do sinal de saída. Esse tipo de *workflow* é tipicamente usado para tratamento de sinais.

Assim como o Kepler, o Taverna [8] também é um sistema de código

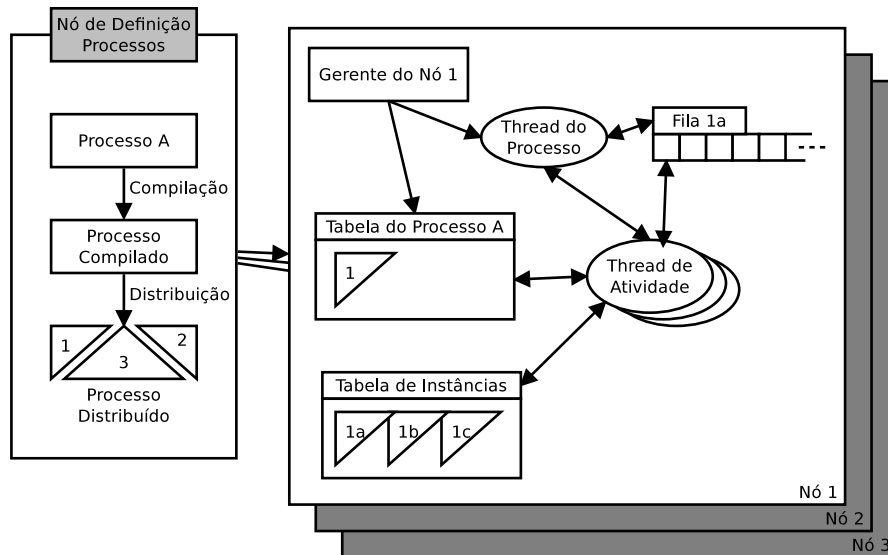


Figura 2.3: Exemplo de distribuição e um nó do Exotica/FMQM.

aberto que permite a automação de métodos experimentais utilizando serviços de vários domínios científicos. O foco desse sistema é disponibilizar para cientistas, com pouco conhecimento em computação, ferramentas para criação de análises complexas de forma que possam ser gerenciadas a partir de um computador pessoal. O sistema é composto por duas aplicações principais: o servidor, com o motor e gerenciamento dos processos e uma aplicação cliente que funciona como uma bancada de trabalho para descrição dos processos. Além dessas aplicações, existem ferramentas auxiliares que permitem a realização de ações como: acesso a base de dados, iniciar processos por linha de comando e interação com um processo em execução. Na figura 2.5 é possível observar o ambiente de desenvolvimento e um *workflow* do Taverna. Nesse ambiente é possível observar os serviços disponíveis para o ambiente, a configuração das ações, entradas e saídas do *workflow* e como é feito sequenciamento de ações para utilizar esses serviços.

2.4

Considerações Finais

Ao fim deste capítulo já temos informações o suficiente para identificar quais características devemos considerar na especificação de um mecanismo de tolerância a falhas para sistemas de gerenciamento de *workflows*. Observado os padrões de construções comumente utilizados, podemos observar que operações não-determinísticas são comuns para sistemas de gerenciamento de *workflow*. Em especial, os padrões que podem definir espera pela disponibilidade de um recurso externo, como *espera adiada* ou a *execução intercalada*, dependem clara-

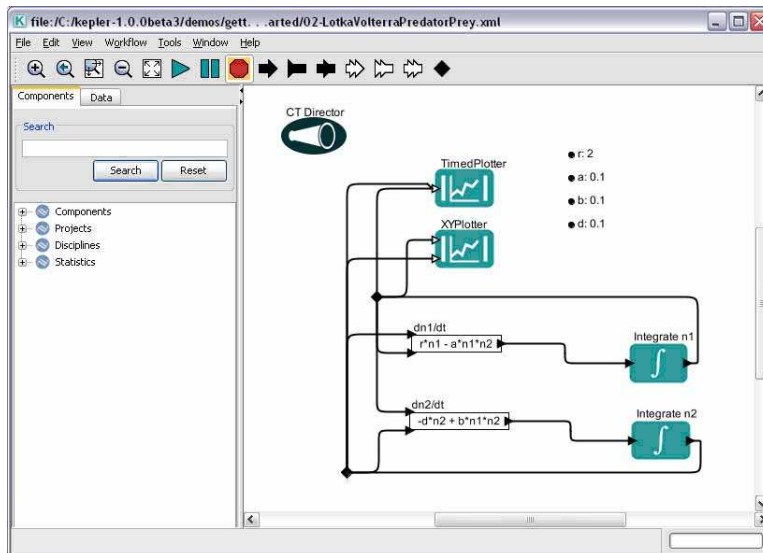


Figura 2.4: Ambiente de desenvolvimento do Sistema Kepler (fonte [7])

mente de informações que só estão disponíveis durante a execução. Além disso, informações provenientes de elementos externos podem mudar de forma arbitrária, ou seja, uma operação que dependa dessas informações pode apresentar resultados diferentes se realizada em momentos distintos.

Por outro lado, cabe ressaltar que neste trabalho iremos focar apenas na limitação da execução centralizada dos *workflows*. A limitação relacionada ao tratamento de exceções e de erros identificados apenas durante a execução, requer alterações nas ferramentas disponíveis para a modelagem do *workflow* e foi usada apenas para ilustrar possíveis melhorias em sistemas desse tipo.

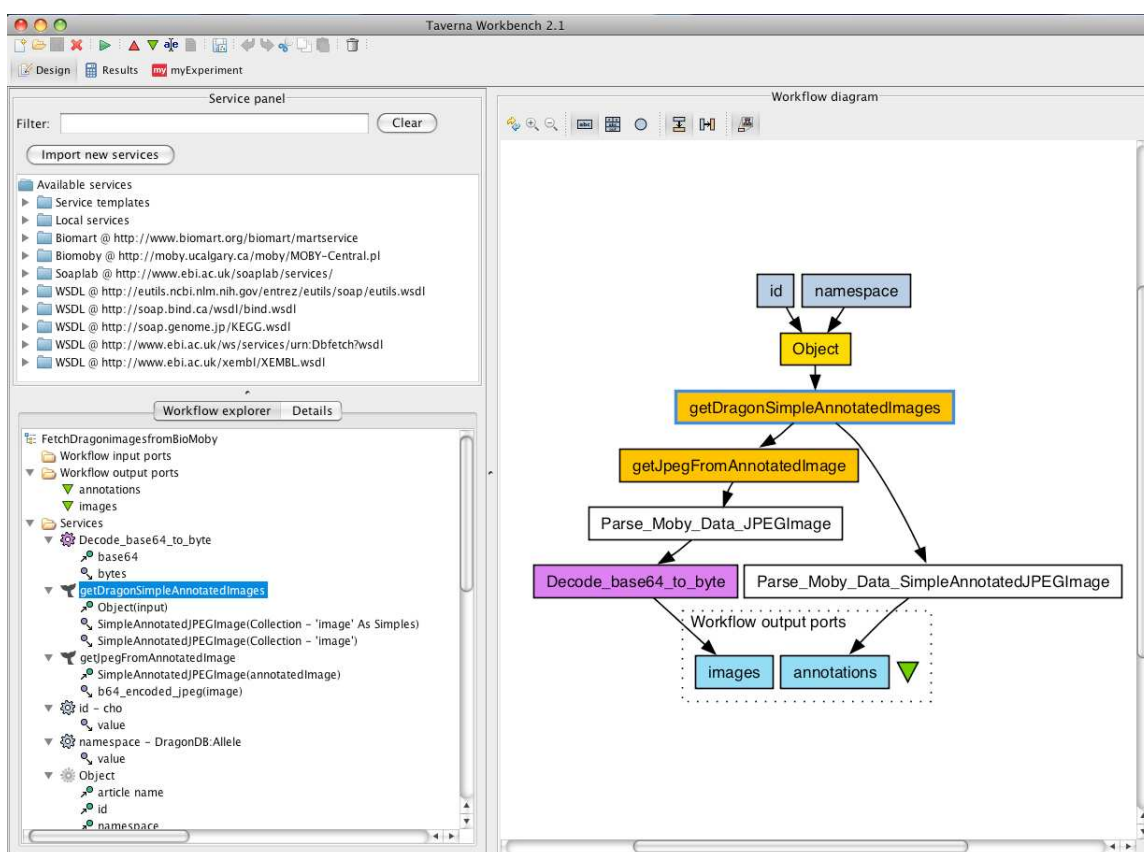


Figura 2.5: Ambiente de desenvolvimento do Sistema Taverna (fonte [9])