

Automação de Processos por meio de *Workflow*: o caso da PROCempa

CARLOS GOMIDE¹

JOSÉ PALAZZO M. DE OLIVEIRA²

(recebido em 14/04/05, aprovado em 19/07/05)

PALAVRAS -CHAVE

Workflow - Automação de processos - Modelagem de workflow

RESUMO

Este artigo apresenta o desenvolvimento e implementação de automação de processos na Prefeitura Municipal de Porto Alegre pela Procempa³, Empresa de Tecnologia da Informação e Comunicação de Porto Alegre, com expressiva redução de complexidade e custo. A criação de aplicações em *workflow* é realizada com auxílio de metodologia de modelagem com uma arquitetura que utiliza a plataforma de desenvolvimento de sistemas preexistentes e um motor de *workflow*, o **Wideflow**. Este motor de *workflow* executa a semântica de um Diagrama de Atividades da UML estendido, construído para suportar a coordenação e execução de atividades. O artigo mostra, também, um estudo de caso de um processo piloto cujos requisitos servem de base para a discussão da solução.

1. INTRODUÇÃO

Este artigo detalha a construção de uma aplicação de automação de processos em um ambiente de engenharia de software. O objetivo é apresentar extensões das construções do Diagrama de Atividades da UML, apresentar um motor de *workflow* construído para executar a semântica do diagrama de atividades estendido, e de descrever a especificação de uma aplicação real onde os principais requisitos de automação de processos discutidos aqui são atendidos.

A experiência relatada desenvolveu-se como um projeto de conclusão do mestrado profissionalizante realizado na UFRGS e desenvolvido na Procempa. Na PMPA os processos de redesenho administrativo desenvolvidos nos últimos anos têm demandado na automação de muitos destes processos. O desenvolvimento de aplicações de *workflow* revelou alguns problemas que conduziram a uma abordagem diferente daquelas geralmente utilizadas no desenvolvimento desta classe de sistemas em muitas organizações. Os problemas encontrados no desenvolvimento de aplicações para atender à automação de processos vão desde a confusão conceitual devida à complexidade dos modelos até a dificuldade na efetiva implementação das abstrações modeladas nas ferramentas existentes.

¹ E-mail: carlos.gomide@procempa.com.br

² E-mail: palazzo@inf.ufrgs.br

³ www.procempa.com.br

Um aspecto do problema se materializa na grande parcela de analistas e projetistas que não conseguem delimitar claramente a fronteira entre *Business Process Reengineering* (BPR) e *workflow*. Uma ambigüidade é associada com o conceito de *workflow-enabled* que muitos produtos anunciam. Estes são produtos como planilhas, processadores de textos e outros que buscam integração com ferramentas de *workflow*. Esse tipo de aplicação não pretende nem possui as características necessárias para suportar automação de processos. Também o Gerenciamento Eletrônico de Documentos (GED) e outras tecnologias têm contribuído para aumentar essa confusão. Georgakopoulos (1996) já listava uma série de tecnologias, todas sob o guarda-chuva da designação *Workflow*, exemplificando o espectro de cada uma delas: desde àquelas orientadas a suportar o trabalho humano, como *Computer Support Cooperative Work* (CSCW), até às orientadas a sistemas automatizados, tais como os sistemas de transacionais.

Além disso, do ponto de vista da modelagem, a semântica das abstrações possui comportamento essencialmente diferente daquelas encontradas em sistemas de informação tradicionais, forçando um reestudo sobre as possibilidades de utilização dos métodos clássicos de Engenharia de *software* para uso na análise e modelagem de *workflow*. Um ponto importante a ser salientado é a necessidade de alinhamento entre o ambiente de modelagem e o de execução do *workflow*. Um maior cuidado no processo de derivação de modelos é necessário para que o motor de *workflow* execute exatamente o que a semântica expressa nos modelos determina, isto é, a arquitetura da ferramenta deve suportar as construções semânticas expressas no modelo.

Por outro lado, a maioria das ferramentas comerciais para automação de processos administrativos possui características sofisticadas que as transformam em verdadeiras plataformas de desenvolvimento de *software*. Isso requer um significativo investimento que vai além do custo das licenças de uso, custo em treinamento no ambiente específico da ferramenta, em seus conceitos, em consequência das limitações e alternativas de solução padronizadas, etc. Esses fatores acabam redundando num grande esforço de criação das aplicações, geralmente sem reuso do código e da modelagem previamente existentes, sendo as novas aplicações construídas do “zero” em ferramentas totalmente integradas do tipo “tudo ou nada” (MANOLESCU, 2001), onde a solução tem que abranger o escopo total do problema sob pena de inviabilizar o uso da ferramenta.

As razões acima que originam uma outra faceta do problema. As aplicações resultantes da automação de processos administrativos⁴ manipulam dados de um caso, uma instância particular de um determinado tipo de processo, dados estes informados pelos atores ou recuperados de arquivos anexos ao processo ou, ainda, dados oriundos de bases de dados de sistemas legados.

Os participantes digitam dados pertinentes ao caso, como valores monetários, nomes e dados de pessoas, informações sobre bens móveis ou imóveis. Alternativamente, arquivos em diversos formatos podem ser anexados ao caso para que fiquem disponíveis aos próximos atores e sejam armazenados historicamente. Outra classe de entrada é constituída por dados típicos de sistemas legados. Para solucionar esses problemas, uma extensão do Diagrama de Ativi-

⁴ Processos administrativos se caracterizam por possuírem passos repetitivos e conhecidos a priori, modelagem simples, forte integração com legado em diversas tecnologias e por tratarem da coordenação do fluxo de trabalho entre pessoas.

dades da UML para modelar e elicitar o controle de atividades, os dados necessários em cada caso e as interfaces para interação entre o *workflow* e os sistemas legados bem como para elicitar os grupos de participantes foi desenvolvida e validada experimentalmente. Um motor de *workflow*, o **Wideflow**, para executar a máquina de estados correspondente ao modelo conceitual foi especificado e desenvolvido. Ainda no escopo deste artigo o diagrama e o motor são validados por meio de uma utilização em um projeto-piloto real da Prefeitura Municipal de Porto Alegre.

2. O WIDEFLOW NA AUTOMAÇÃO DE PROCESSOS DA PMPA

O **Wideflow** é um motor de *workflow* criado para suportar processos de produção tipicamente encontrados em administrações públicas, onde uma pasta contendo documentos trafega entre várias pessoas num fluxo previamente conhecido e com poucas variações, sendo alvo de verificações de conformidade, autorizações e despachos.

Neste motor de *workflow* o controle do fluxo dos processos é dado suportado pelo polimorfismo existente nas especializações de uma atividade, isso é, para o **Wideflow** todas as abstrações que representam *nós* no fluxo de trabalho, inclusive *forks* e *joins* (CASATI et al, 1995) bem como as transições com condição de guarda (OBJECT MANAGEMENT GROUP, 2003) tornam-se atividades que o motor de *workflow* deve executar, cada uma associada a um evento específico. O pressuposto para habilitar a transição entre um nó e outro é o término da execução do nó precedente. Assim a dinâmica básica do **Wideflow** é dada pela procura por transições possíveis.

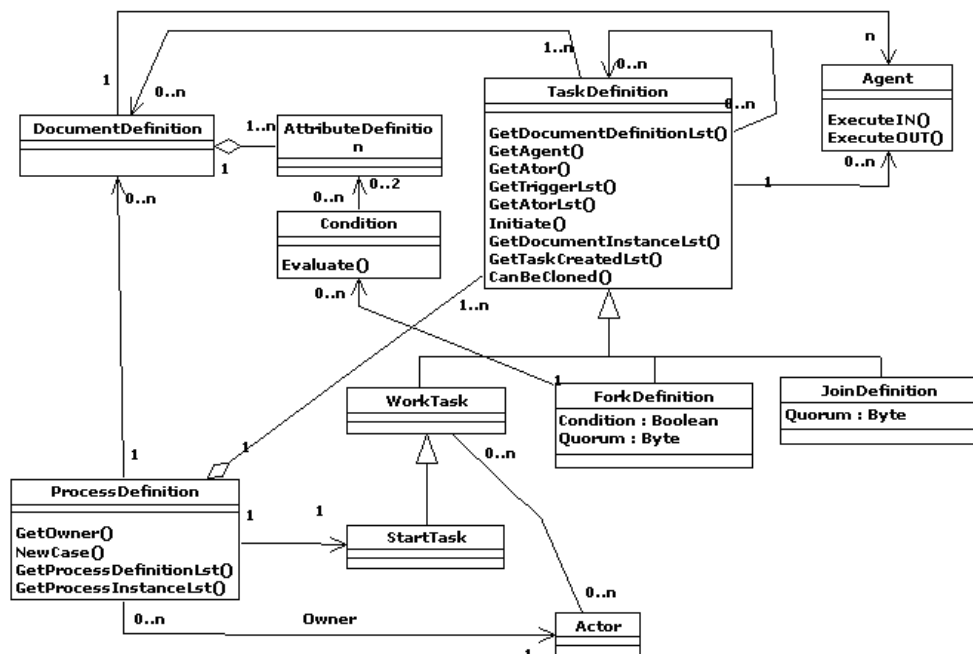


Figura 2.1 – Meta-modelo do Wideflow

Os eventos que alteram o estado de um processo são comumente disparados pelos participantes do processo na interação destes com os casos em suas *tasklists* (WORKFLOW MANAGEMENT COALITION, 1995). Estes eventos são responsáveis por indicar ao motor de *workflow* que a atividade pode ser finalizada – são as atividades modeladas pelo analista, do tipo *WorkTask*. Outra classe de evento é composta pela ocorrência de períodos determinados de tempo, atividades temporais e executadas pelo motor de *workflow* de acordo com um período determinado. Existem ainda atividades denominadas *Fork*, derivadas de transições com condição de guarda e *Joins*, atividades internas que o motor de *workflow* executa para validar a transição ou para controlar e sincronizar fluxos convergentes.

No motor **Wideflow** são definidos dois eventos no ciclo de vida de uma instância de atividade: o evento de inicialização e o evento de finalização. O mecanismo do motor de *workflow* interpreta a definição do processo a cada ocorrência de evento de inicialização, quando as classes de definição criam instâncias de objetos das classes de execução e seus relacionamentos com atores e documentos associados. Permitir alterações no esquema significa permitir alterações no modelo da definição, desenvolvido pelo analista, sem que isso implique nas costumeiras manutenções trabalhosas e sujeitas a erros para que as instâncias existentes mantenham integridade em relação à definição vigente quando da sua criação.

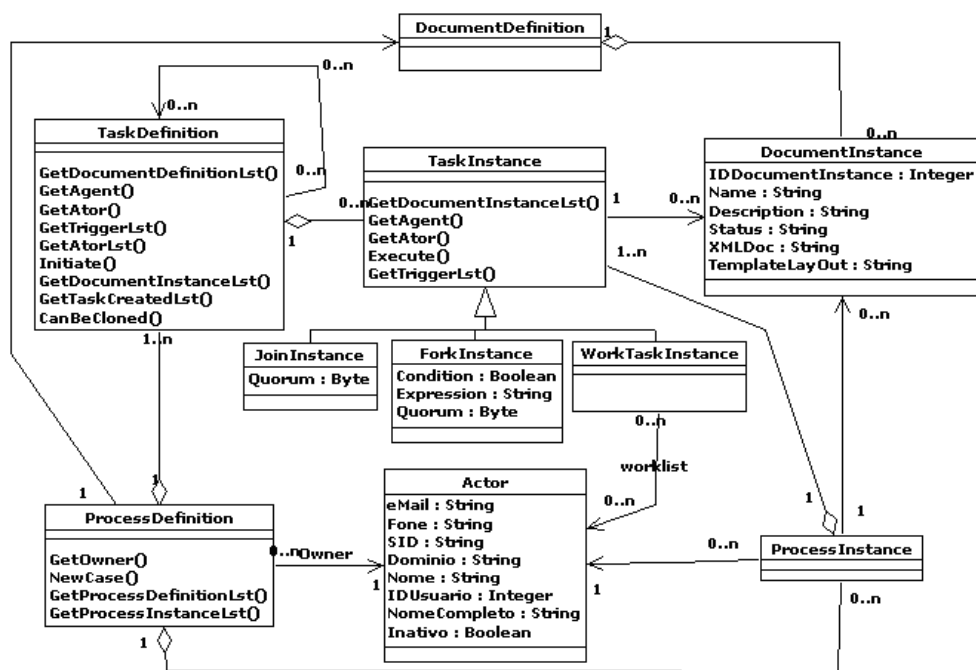


Figura 2.2 – Meta-modelo do Wideflow

Condições de guarda (OMG, 2003) controlam a seqüência no fluxo do processo. Tratadas também como atividades internas cuja responsabilidade de execução cabe ao motor de *workflow*, como na figura 2.3, elas podem impor condicionantes para a transição entre origem e o destino de atividades ou sobre cada transição existente entre uma divisão do fluxo e um destino, permitindo concorrência ou paralelismo no fluxo do processo. No meta-modelo do **Widflow** elas, as condições de guarda, são implementadas por um *Fork*. Aqui novamente pode ser observado que a especialização de atividades simplifica o motor do workflow.

O assinalamento de atores, atribuição de uma atividade a um ator, é feito pela associação entre um ator e uma instância de atividade (veja figura 2.2). Os atores são dispostos em grupos aos quais são atribuídas as permissões de acesso à atividade. Tipicamente um grupo permite o acesso de seus membros a uma atividade num relacionamento 1:1. Adicionalmente a estratégia do assinalamento determina uma das seguintes dinâmicas:

- a) PUSH ou PULL (CASATI et al, 1995);
- b) Ao iniciador do processo – a atividade é colocada na lista de trabalho da instância de ator que iniciou o caso;

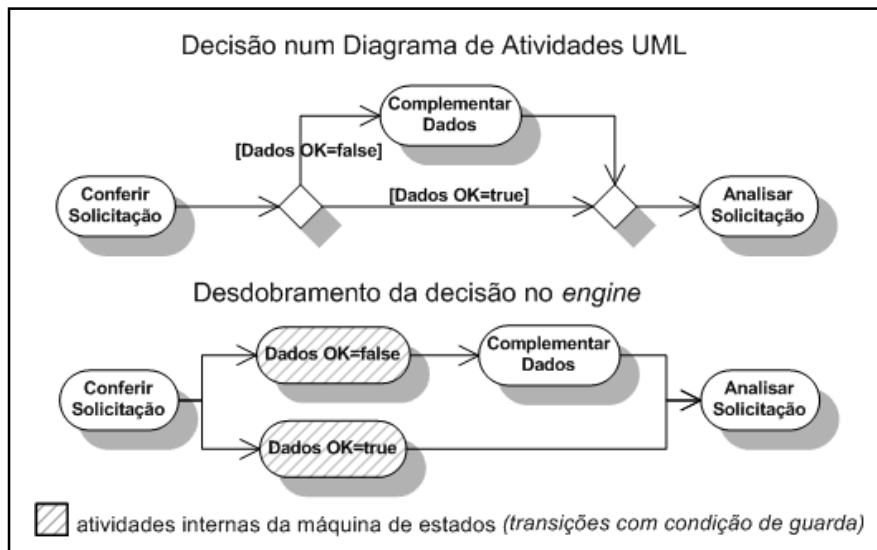


Figura 2.3 – Desdobramento de condições de guarda

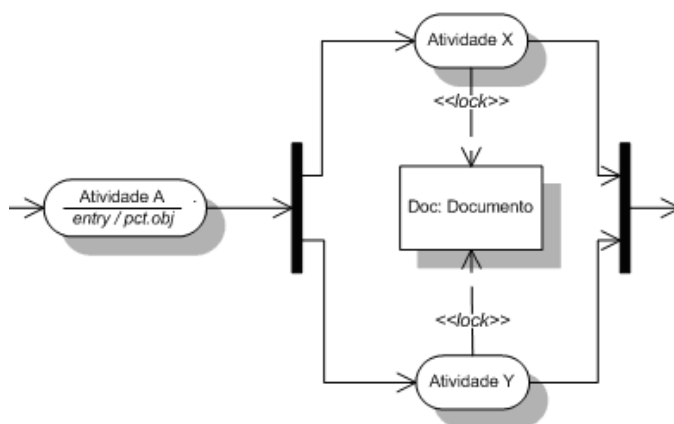
- c) Ao executor da tarefa X – a atividade é colocada na lista de trabalho da instância de ator que executou a atividade X no mesmo caso.

Os usuários do *workflow*, apesar de possuírem atribuições variadas, na maioria das vezes ocupam o mesmo cargo. Assim, um ator inserido em um grupo

possui as permissões daquele grupo. A estratégia de assinalamento é imposta pela definição da atividade e avaliada na dinâmica de execução, quando as atividades são depositadas nas *tasklists*.

Para o motor de *workflow* interessa o conjunto de permissões de cada ator frente ao processo da mesma forma como feito no controle de acesso às funcionalidades da maioria dos sistemas de informações atuais. A identificação do ator, que permite o acesso à sua *tasklist* e às atividades nela depositadas, é feita a partir do *login* de rede, utilizando os componentes do sistema de segurança.

As informações dos casos, instâncias de processo, são oriundas dos dados digitados em formulários eletrônicos. Podem ser utilizadas as interfaces de apresentação do **Wideflow**, que interpretam os documentos modelados na definição do processo, apresentando seus atributos para inserção, alteração ou visualização,



ou interfaces de apresentação customizadas contendo uma lógica de apresentação mais refinada, tratando a interação do ator com *scripts* no cliente ou com listas combinadas originadas em bases legadas.

Os atributos do processo são agrupados em documentos manipulados e tornados permanentes pelo **Wideflow**. Com isso é possível modelar documentos com seu conjunto de atributos e impor restrições de acesso concorrente (*lock* de um documento entre atividades) como na figura 2.4, determinar a visibilidade de documentos em determinadas atividades e manipular a propagação de dados entre atividades como na figura 2.5. Eles são tornados permanentes em formato XML (WORLD WIDE WEB CONSORTIUM, 2003) o que facilita a reflexão sobre sua estrutura e conteúdo e a evolução do esquema de maneira independente. É possível ainda modelar um documento impondo o estereótipo *occur* no relacionamento entre este documento e uma atividade, de modo que o documento possua várias ocorrências em uma determinada atividade, permitindo a inserção de um lote de informações do mesmo tipo como na figura 2.5.

Os documentos representam um conjunto de dados agregados numa mesma entidade do processo e são os únicos dados sob responsabilidade do motor de *workflow*. As condições de guarda são atividades internas que avaliam expressões com estes dados como operandos, da expressão tomando as decisões que

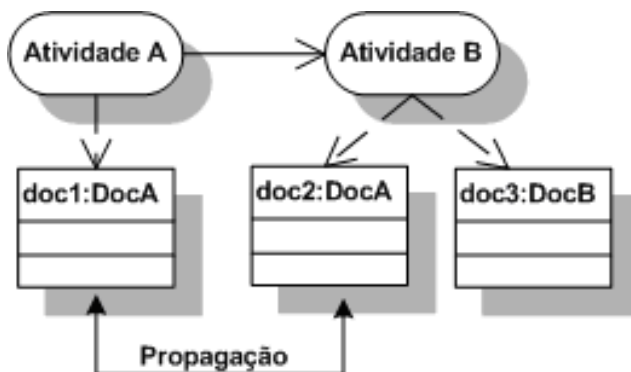


Figura 2.5 – Propagação de documentos

direcionam o fluxo do processo. Assim, para que possam ser utilizados no caso, os dados devem ser encapsulados em um documento modelado explicitamente no diagrama de atividades (ESHUIS; WIERINGA, 2001).

As informações que serão necessárias no fluxo do processo e que são mantidas por sistemas de informações devem, antes da utilização no motor de workflow, ser apropriadas ao documento. A integração com outros sistemas é feita ao nível de programa (SILVEIRA; BARROS; PEREIRA, 2000), preservando os domínios de cada sistema transacional. Os agentes, interfaces COM+ (KIRTLAND, 2000), com propriedades e um método padrão são o mecanismo para realizar esta integração. O **Widflow** obtém a referência para a interface do agente e a percorre, como em uma reflexão, atribuindo os valores dos documentos à propriedade na interface. Após, é feita a execução do método padrão que implementa o algoritmo definido para interagir com o sistema legado e que também atualiza nas propriedades da interface do agente os novos valores que deverão ser apropriados.

Um agente pode ser associado a uma *action* (OMG, 2003) a ser executada na inicialização de uma atividade (*entry*) ou após a execução da atividade (*exit*), quando itera os atributos existentes na interface de um documento, percorre a interface de todos os documentos visíveis àquela atividade. Também é possível associar um agente a um documento específico em uma determinada atividade, o que significa que somente os dados daquele documento irão interagir com o agente, como mostra a figura 2.1. A correta execução do agente é o pressuposto para a troca de estado, o término da atividade ao qual ele está ligado.

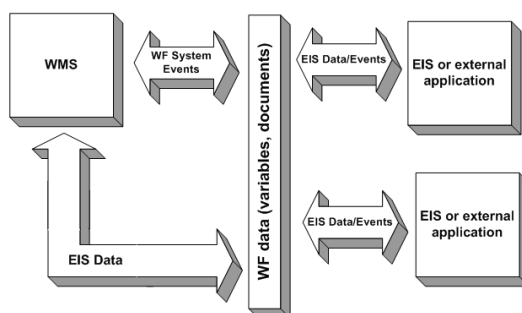


Figura 2.6 – Interface entre workflow e sistemas de informação

Com isso é possível trocar dinamicamente dados, transformando-os e enviando-os ou obtendo-os de aplicações legadas sem invadir seus domínios, como exemplifica a figura 2.6, com o que muitas construções são possíveis na modelagem dos processos.

Arquivos anexados aos processos são armazenados em uma estrutura de diretórios cuja *url*, indicando a localização do arquivo, é tornada permanente em uma instância de documento. Toda manipulação ocorre por meio da especialização dos métodos da classe que transportam o arquivo da aplicação cliente para o motor de *workflow* e vice-versa.

O interessante na arquitetura do **Wideflow** é que ela foi projetada com o mesmo *framework* destinado ao desenvolvimento de aplicações então utilizado na Procempa. O motor de *workflow* se constitui na camada de negócio forman-

WideFlowExec
UsuarioAtivo : IGRDUsuario
<pre> getProcessDefinitionList() : TWFWProcessDefinitionLst initProcess(IDProcessDefinition : Long, Descricao : String) : Boolean getTaskInstanceList(IDProcess : Integer) : TaskDefinitionLst getTaskInstanceDocLst(IDTaskInstance : Integer) : TWFWDocumentInstanceLst getDocumentInstance(IDDocumentInstance : Integer) : ADORecordSet updateDocumentInstance(IDDocumentInstance : Integer, Doc : ADORecordSet) : Boolean executeTaskInstance(IDTaskInstance : Integer) : Boolean getTaskInstanceExpiredList() : TWFWTaskInstanceLst getTaskInstanceInDeadLockList() : TWFWTaskInstanceLst getActorList(IDTaskInstance : Integer) : ActorLst getProcessParticipationList(IDUsuarioAtivo : Integer) : TWFWProcessInstanceLst toDelegate(IDTaskInstance : Integer, IDActor : Integer) : Boolean getTaskInstance(IDTaskInstance : Integer) : TaskInstance </pre>

Figura 2.7 – API do componente de execução

do o cerne funcional do **Wideflow**. Os componentes implementam abstrações do meta-modelo das figuras 2.1 e 2.2 e realizam a semântica do diagrama de atividades estendido.

Utilizar estes componentes exigiria que a aplicação cliente conhecesse a estrutura do meta-modelo, sabendo exatamente para qual objeto apontar, o método a ser invocado e, principalmente, como navegar pelos relacionamentos estruturais. Para facilitar o uso do motor de *workflow* foi criado um componente que fornece uma API pública de serviços, inspirada nas *Workflow API* (WAPI) 2 e 3 (WMC, 1995), apresentado na figura 2.7, para que os clientes o utilizem. Tanto os formulários eletrônicos do Wideflow quanto as aplicações cliente utilizam este componente.

A modelagem e definição do processo, *Workflow API 1* (WMC, 1995), é feita no diagrama de atividades estendido utilizando o *Rational Rose Enterprise* como ferramenta CASE. Este diagrama já é utilizado pelos analistas e desenvolvedores da Procempa na elicitação e modelagem de sistemas de informação (ARAÚJO; GOMIDE; PETRILO, 2003), o que facilita a disseminação, treinamento e manejo da ferramenta.

A definição do processo é obtida deste diagrama e tornada permanente no domínio do **Widelflow** para que seja acessada e interpretada a cada passo no ciclo de vida das instâncias dos processos. Um *script* desenvolvido em *Rational Rose Script Language*^a percorre o gráfico e traduz suas construções para o domínio do motor de *workflow*. A automação do processo de Solicitação de Declaração Municipal

Na administração pública do município de Porto Alegre, como em muitas outras cidades, as demandas individuais dos cidadãos geram o registro em papel que passa a constituir um processo administrativo ou expediente: uma pasta onde se registram todos os pareceres, tramitações, providências, adendos, documentos e decisões de todas as partes da estrutura que atuam operacionalmente no atendimento ou devem tomar conhecimento do teor do processo.

O processo para solicitação de uma Declaração Municipal (DM) é utilizado pelos cidadãos na etapa que antecede à submissão, ao órgão municipal competente, de projetos para modificação do solo urbano, seja tanto pela construção ou reforma de imóveis quanto pelo parcelamento e divisão territorial privadas. A DM, regulada pelo Decreto 12.715/2000 do Município de Porto Alegre, informa as condições urbanísticas de ocupação do solo.

A automação prevê a solicitação da DM pelo cidadão através da Internet com a anexação dos documentos necessários e a tramitação eletrônica pelos diversos órgãos e secretarias municipais responsáveis pela avaliação das condições de uso do solo em sua área de competência específica para após produzir o documento final contendo as prescrições, recomendações e impedimentos relativos à área territorial em análise. A DM gerada é enviada pelo correio eletrônico diretamente ao solicitante. O grau de automação de cada atividade do processo depende diretamente da qualidade das informações eletrônicas existentes. Um alto nível de automação só seria possível com a existência de bases georeferenciadas, sem as quais a maioria do trabalho em cada atividade ainda estará sob a responsabilidade de um funcionário que faça a análise do caso.

As figuras 3.1 e 3.2 apresentam um modelo com a semântica dos usuários do processo, os funcionários dos setores envolvidos na análise da solicitação. O solicitante, cidadão interessado na DM, é tratado como um *Business Actor* (JACOBSON; ERICSON; JACOBSON, 1994). O *Business Use Case* (JACOBSON; ERICSON; JACOBSON, 1994) a seguir descreve o processo em alto nível.

Para iniciar o processo o requerente acessa a aplicação no seu navegador e informa a inscrição do imóvel no IPTU, o que identifica uma economia.

A aplicação deve verificar todos os endereços que compõem o bloco de endereços ao qual o imóvel pertence. Após ela deve selecionar o número de um Expediente Único previamente existente que englobe o conjunto de endereços selecionados anteriormente.

Se existir uma DM válida, com data de emissão menor que um ano, ela deve ser apresentada para que o requerente possa optar pela re-emissão de uma 2ª via, caso em que não será necessário enviar os arquivos como descrito abaixo.

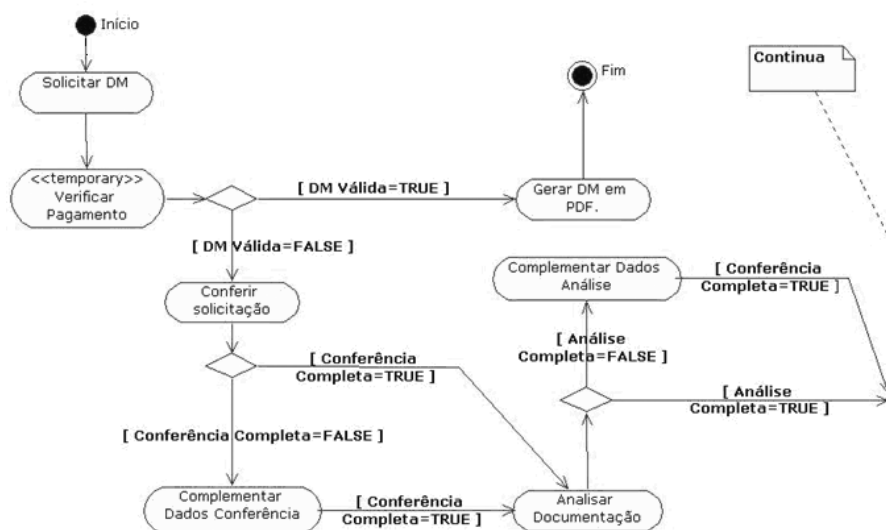


Figura 3.1 – Processo Solicitação de Declaração Municipal

O requerente deve fazer o *upload* dos arquivos contendo a planta de situação e o título de propriedade digitalizados, para que sejam armazenados e tramitem com o processo.

Após , ele emite o documento para pagamento da taxa relativa à expedição da DM, o DAM – Documento de Arrecadação Municipal, e a solicitação está completa.

A segunda atividade prevê que o requerente efetue o pagamento do DAM e após o aviso do crédito bancário o processo tem seguimento.

Se o requerente tiver optado por receber uma 2ª via de DM, como descrito na atividade de Solicitação da DM, ela é enviada e o processo termina. Caso contrário, a solicitação é encaminhada para a atividade de Conferência da Solicitação onde o conferente verifica todos os dados e pode alterar o conjunto de endereços e/ou o número do Expediente Único da solicitação. Se o conferente perceber a falta de informações ou de arquivos ele envia um *e-mail* ao requerente que deve encaminhar as informações em falta enquanto a solicitação permanece aguardando a complementação de informações.

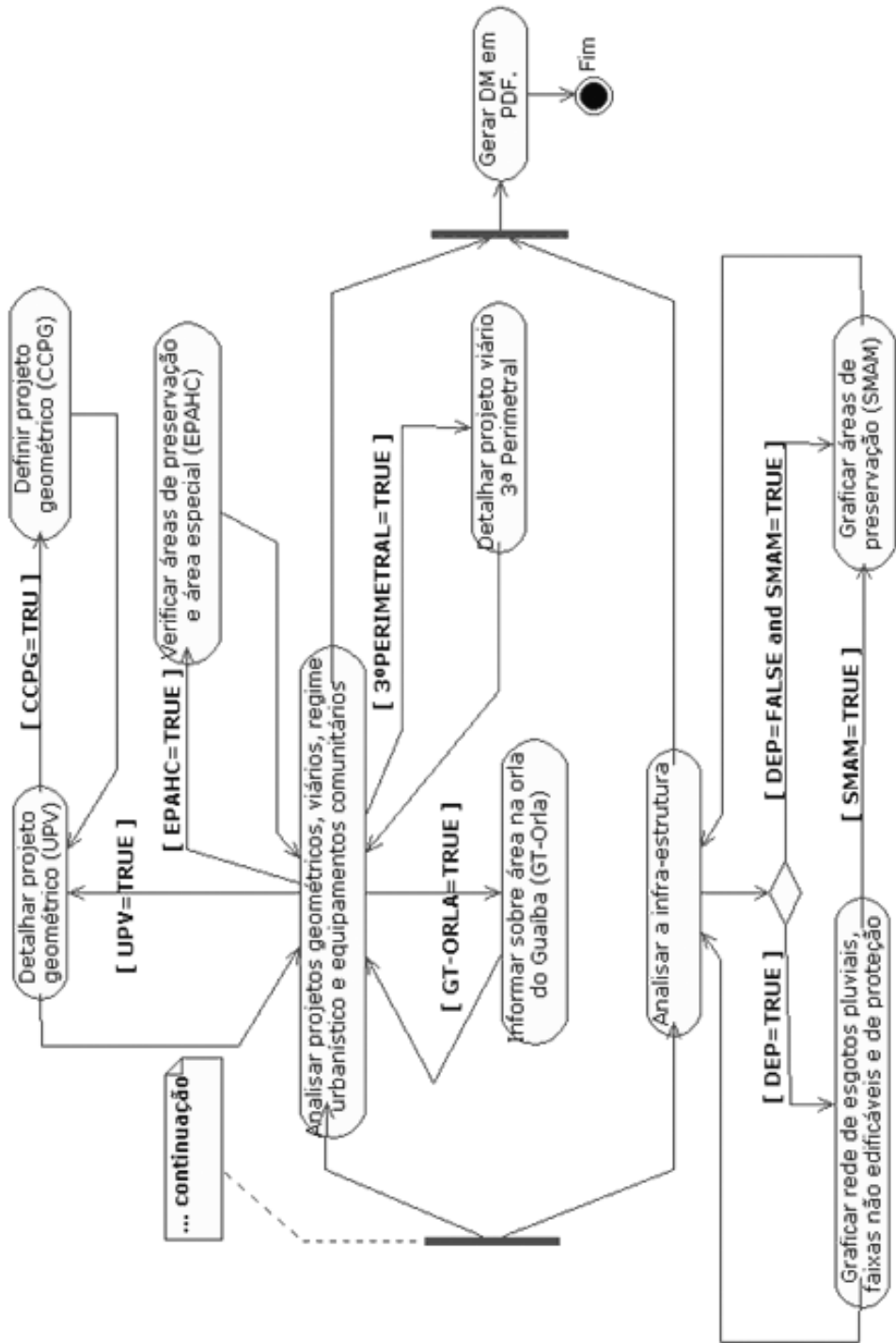


Figura 3.2 – Definição do processo Solicitação de Declaração Municipal

Quando o conferente receber a resposta do requerente com a complementação ele as anexa à solicitação e encaminha para a atividade de Análise da Documentação.

É nesta atividade que os arquivos com planta de situação e título de propriedade são analisados, em conjunto com as outras informações. Desta atividade

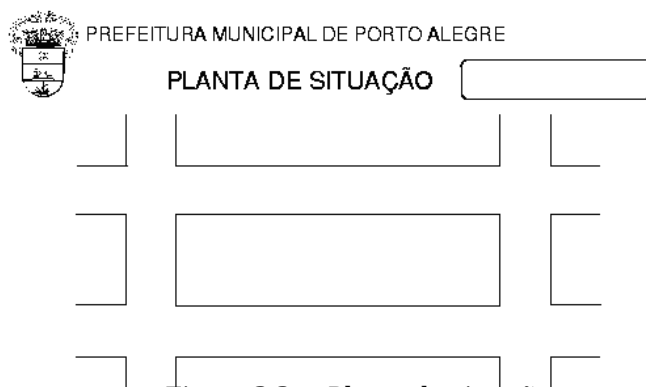


Figura 3.3 – Planta de situação

pode resultar a impropriedade da solicitação, quando o analista pode pedir por *e-mail* que o requerente complemente ou altere as informações técnicas, fazendo com que a solicitação permaneça aguardando a resposta do requerente.

Quando as informações são complementadas o processo segue para a atividade de Análise de Projetos e paralelamente para a atividade de Análise de Infra-Estrutura.

Além de analisar sob perspectiva própria os detalhes da solicitação, os participantes nestas duas atividades, técnicos especializados, decidem os próximos passos da solicitação.

Ela pode ser enviada para que outros técnicos em setores, órgãos ou secretarias diversas, analisem e aponham informações, condicionantes ou vetos às possibilidades de uso da área geográfica em exame.

Após esta circulação o processo deve retornar às atividades de origem – Análise de Projetos e Análise de Infra-Estrutura – quando os analistas podem decidir pelo reenvio para algum daqueles setores ou órgãos para nova análise ou podem decidir pela finalização de seu trabalho.

Para que o processo realmente possa ser finalizado, tanto o técnico responsável pela atividade de Análise de Projetos quanto o técnico responsável pela atividade de Análise de Infra-Estrutura têm de considerar o processo finalizado.

Quando isso ocorrer a DM deve ser gerada e enviada ao requerente e também ser armazenada para consultas ou emissões posteriores.

Se o requerente não pagar o DAM em 72 horas a solicitação é encerrada sem avisos. Igualmente se ele não responder aos pedidos de complementação de informações em 15 dias, no caso do pedido feito pelo conferente, ou não responder em 90 dias, no caso do pedido ser feito pelo analista da documentação.

A planta de situação, figura 3.3, é um arquivo gráfico em formato simples (JPG) anexado, que tramita entre os vários órgãos. Ela será manipulada em diversas atividades por órgãos e setores diferentes. Em algumas situações é possível que exista concorrência sobre o arquivo, com um setor executando sua ativi-

Pré-Condição:

Requerente autenticado.

Fluxo Básico:

- a) O sistema solicita as inscrições de IPTU dos endereços para os quais deseja uma DM e a área total titulada destes imóveis.
- b) O Requerente informa uma ou mais inscrições e a área total titulada e submete a solicitação.

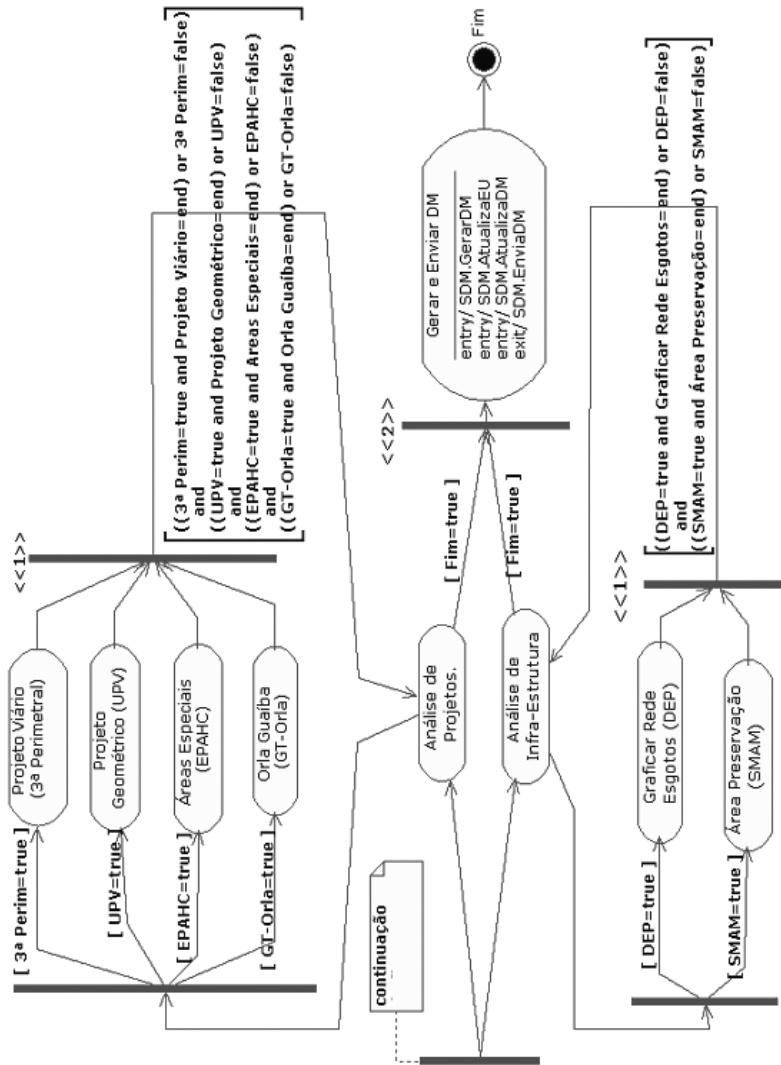


Figura 3.5 – Definição do Processo de Solicitação de DM

- c) O sistema valida os dados informados e dispara no Wideflow os eventos que executam os agentes responsáveis por preencher os dados nos documentos: os endereços para cada inscrição no banco do Ca-

distro Imobiliário, a descrição dos logradouros no banco do Cadastro de Logradouros e o número de Expediente.

Único se já existir um para o conjunto de endereços. Se faltar alguma inscrição, que também pertença ao imóvel ou bloco de imóveis ela deverá ser obtida. Apresentar para confirmação dos endereços.

- d) O Requerente confirma os endereços.
- e) Após, exibe a tela com a opção “Enviar Documentos” habilitada e a opção “Emitir DAM” desabilitada, com a seguinte mensagem: “Solicitação enviada. Aguardando o envio da Planta de Situação e do Título de Propriedade. O código desta solicitação é 99”.
- f) O Requerente seleciona a opção de *upload*, selecionando os arquivos e iniciando a transferência.
- g) O sistema recebe os arquivos e dispara no Wideflow os eventos que executam os agentes responsáveis pela atualização dos respectivos documentos.
- h) O sistema DESABILITA a opção “Enviar Arquivos” e habilita a opção “Emitir DAM”, com a seguinte mensagem: “Documentos enviados. Aguardando a emissão e o pagamento do DAM. O código desta solicitação é 99.”.
- i) O Requerente seleciona a opção de emissão de DAM.
- j) O sistema dispara no Wideflow o evento que executa o agente que fornece os dados para o DAM (ver use case “Emitir DAM”) e emite-o. O sistema marca como *true* o atributo “Solicitação Completa”, dispara o evento de atualização do documento correspondente no Wideflow e dispara a execução da atividade Solicitar DM para realizar a transição para a próxima atividade do processo. Após, exibe a tela com somente a opção “Voltar” habilitada.

Pós-Condição:

Uma nova instância de processo para solicitação de uma DM, com os documentos previstos, inicia a tramitação.

Fluxo Alternativo:

No passo “c”, se o sistema verificar que existe uma DM válida, isto é, com data de emissão menor que um ano, deverá se comportar como segue:

- a) O sistema mostra esta DM e oferece a opção de emissão de uma segunda via.
- b) O Requerente opta por emitir a segunda via.
- c) O sistema marca a informação “DM Válida” modelada no objeto “Solicitação de DM” como *true* e executa o método de atualização de documentos do Wideflow para este documento.
- d) O fluxo segue para o passo i

Se o sistema SDM, ao interagir com o requerente, a qualquer momento considerar que a documentação para solicitação do processo está completa, disparará o evento de execução da atividade.

A atividade Verificar Pagamento, uma atividade temporal, é instanciada e sua execução é feita pelo próprio motor de *workflow* diariamente às 06:00, pois o processamento diário dos avisos de crédito bancário ocorre antes deste horário. O motor de *workflow* altera o estado da tarefa para terminada, o que dispara o agente, previsto na *action* **SDM.Pagto** como discutido anteriormente. Este agente executa um algoritmo que acessa o sistema de Controle de Arrecadação Municipal, verificando se o pagamento do documento emitido já foi apropriado. Em caso positivo o Agente retorna *true*, em caso contrário, *false*.

A execução do Agente com sucesso (*true*) faz com que o motor de *workflow* mantenha o estado da atividade como terminado, de outro modo, o motor de *workflow* faz com que a atividade retorne para o estado anterior, pronta para executar novamente quando for fechado mais um período.

Na seqüência é feita uma avaliação: se o requerente optou por emitir uma segunda via, ainda na primeira atividade. Em caso afirmativo o fluxo segue para a atividade onde a DM será gerada e transformada em *Portable Document Format* (PDF) e enviada ao requerente e o processo tem seu fim.

Caso contrário - ou não existia uma DM válida ou o requerente tenha optado por emitir uma nova DM, o fluxo se encaminha para a atividade de Conferência da Solicitação. Neste passo um conferente fará diversas análises entre a solicitação e os dados, podendo decidir por utilizar um outro código para o Expediente Único que considere mais apropriado para o processo ou; ainda, inserir novos endereços para aquele Expediente Único. Com isso o SDM dispara eventos contra o **Widelflow** que fazem com que ele atualize os dados sobre os logradouros e endereços da solicitação.

O conferente pode decidir pela complementação de documentação marcando a decisão num atributo do processo e disparando a execução da atividade. O motor de *workflow* passa o estado da instância da atividade para “terminada” e inicia a atividade Complementar Dados Conferência. Um e-mail é enviado ao requerente pelo agente **SDM.E-mail**, solicitando os dados desejados com o texto redigido pelo conferente. O requerente envia os dados para o conferente que os anexa ao processo substituindo os anteriores. Estas duas atividades serão executadas pelo mesmo ator e sua divisão ocorre por decisão de análise, para marcar dois estados distintos no processo. Se o requerente não responder em um período de 15 dias o motor de *workflow* irá encerrar o processo. Ao receber a complementação dos dados, o conferente encerra a atividade de complementação e o fluxo segue para a atividade seguinte.

Uma instância de atividade Analisar Documentação é iniciada e um ator assinalado. A interação ator *versus* atividade é análoga à atividade anterior de conferência, com análise e envio de *e-mail* ao requerente para possível complementação de documentação. Quando a análise estiver completa, ou na atividade Analisar Documentação ou na atividade Complementar Dados Análise, será executado o agente **SDM.AtualizaCEU**, que atualiza o sistema de Cadastro de Expedientes Únicos com as informações deste processo.

Após esta primeira análise o fluxo segue paralelamente para a atividade Análise de Projetos Geométrico, Viário, do Regime Urbanístico, e de Equipamentos Comunitários e para a atividade Analisar Infra-Estrutura.

A atividade Analisar Projetos Geométricos, Viários, Regime Urbanístico e Equipamentos Comunitários e a atividade Analisar a Infra-Estrutura são executadas também com apoio de interfaces próprias do SDM introduzindo a lógica própria que os atores utilizam para realizar seu trabalho. A planta de situação é utilizada de forma concorrente nessas atividades.

Na primeira, por decisão do ator com base no contexto, o processo pode ser enviado para detalhamento, verificação ou inclusão de informações em outros setores ou órgãos. Ao fazer o encaminhamento a atividade é encerrada. Um detalhe em relação às diversas atividades em paralelo é a concorrência entre os documentos enquanto as atividades são realizadas.

As atividades Detalhar Projeto Geométrico, Verificar Áreas de Preservação, Detalhar Projeto Viário, Informar Área da Orla e Definir Projeto Geométrico, ao modificarem dados nos documentos do processo, devem modificar a mesma instância de documento, preservando a integridade das informações que, ao retornarem para a atividade Analisar Projetos Geométrico, Viário, Regime Urbanístico e Equipamentos Comunitários, estarão completos, permitindo ao ator dar seguimento à solicitação.

Interação semelhante sofrerá a atividade Analisar Infra-Estrutura, onde também os documentos serão compartilhados concorrentemente. Se existirem documentos comuns nas atividades de análise de projetos e análise de infra-estrutura e suas seqüências, eles deverão observar o requisito de concorrência.

O participante assinala os atributos correspondentes aos setores, departamentos ou secretarias por onde o processo deverá tramitar. Ao executar a atividade, o Wideflow avalia as condições impostas pelo ator e realiza o fluxo desejado, instanciando as atividades cuja transição de origem tenham as correspondentes condições de guarda resultado como verdadeiras.

Após a execução de cada atividade, o *join* é instanciado, pois o *quorum* definido é igual a 1. A condição de guarda na transição de saída do *join* faz o Wideflow avaliar se todas as atividades que foram instanciadas já encerraram, quando a transição ocorre e o fluxo faz com que o motor de *workflow* crie uma nova instância da atividade Analisar Projetos Geométricos, Viários, Regime Urbanístico e Equipamentos Comunitários ou Análise Infra-Estrutura, conforme o seqüenciamento.

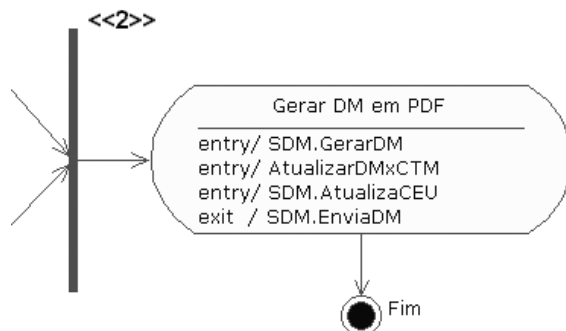


Figura 3.6 – Atividade Gerar DM

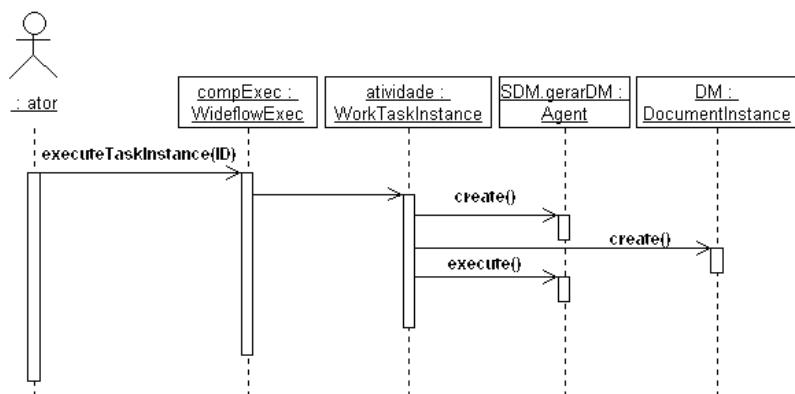


Figura 3.7 – Trecho do diagrama de sequência da atividade Solicitar DM

Este conjunto de atividades de análise e verificação é feito com auxílio das informações recolhidas dos sistemas de informação e também pela análise não automatizada de plantas, mapas e outras informações gráficas que cada órgão mantém. A automação destas atividades necessita da conclusão de projeto de informações georreferenciadas em execução pela administração municipal. As atividades são realizadas com o auxílio das IHM do próprio Wideflow que incorporam o padrão lista à detalhe, muito utilizado em sistemas de informação que têm sua camada de apresentação baseada em interfaces Web.

Quando considerarem finalizados seus respectivos trabalhos, os atores responsáveis pelas atividades Analisar Projetos Geométricos, Viários, Regime Urbanístico e Equipamentos Comunitários ou Análise Infra-Estrutura marcarão o atributo “Finalizado” em suas respectivas atividades e dispararão o evento de execução da atividade. Desta feita o único caminho verdadeiro para cada atividade é a transição que leva ao *join* final que possui *quorum* igual a 2, significando que o Wideflow somente executará a transição de saída do *join* quando as duas transições de origem ocorrerem.

A atividade Gerar DM executa três agentes em sua inicialização (*entry*) como na figura 3.6. Dois deles atualizam sistemas de informação, o outro dispara o evento para execução do programa que gera a DM no domínio do sistema de informação, recebe-a e transforma-a em PDF, armazenando o arquivo resultante na árvore de diretórios do motor de *workflow* e o endereço no objeto “DM em PDF”. O quarto agente, executado na finalização (*exit*) da atividade, envia a DM via *e-mail*.

O processo de desenvolvimento de aplicações de automação de processos conta com profissionais que desempenham papéis diversos. A seguir, os principais papéis e suas responsabilidades são descritos.

Ao analista de negócios cabe fazer a análise do processo, identificando os requisitos que os clientes, usuários e patrocinadores pretendem ver atendidos. Ele se utiliza basicamente do Diagrama de Atividades da UML, sem extensões, modelando o processo de negócio, e elabora um *Business Use Case* descrevendo as interações de cada atividade com a estrutura organizacional e os produtos necessários como entrada e gerados na saída de cada uma delas. Uma vez compreendidos os requisitos, ele elabora o documento que compreende a visão do produto, com o detalhamento do escopo da aplicação.

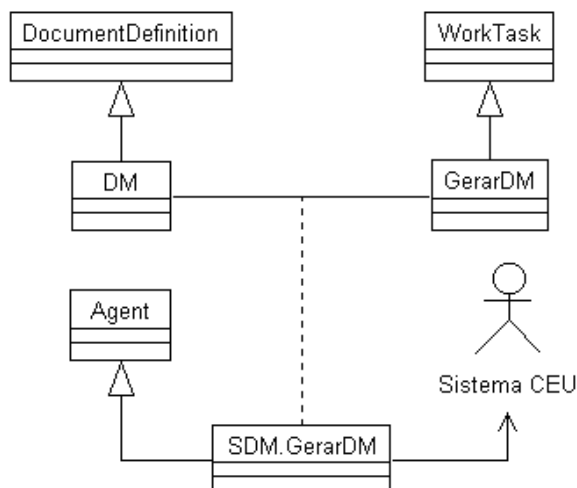


Figura 3.8 – Modelo de classes da atividade Gerar DM

Mais adiante o analista de sistemas irá detalhar a aplicação, modelando o Diagrama de Atividades estendido discutido aqui. Ele também elabora os diagramas de sequência com todas as entidades envolvidas na execução de cada atividade, como exemplifica a figura 3.7 e um modelo demonstrando o relacionamento entre as classes envolvidas, como o da figura 3.8. Também ele é o responsável por enviar a definição do processo para o **Widflow** e por especificar o trabalho dos programadores.

Ao programador cabe codificar os agentes conforme descrito na especificação do objeto no modelo. Em suma, este é o processo de trabalho utilizado para desenvolver aplicações de *workflow*.

3. CONCLUSÕES

Como resultado deste trabalho, podemos afirmar que a proposta de desenvolvimento de aplicações de *workflow* suportadas por um processo de Engenharia de Software adequado, utilizando um motor de *workflow* como suporte para a aplicação permite a aplicação de uma metodologia consistente e eficaz para esta classe de sistemas. Por outro lado permite, também, a criação de aplicações mistas: sistemas de informação integrados com *workflow*, num ciclo de vida de desenvolvimento integrado, para atender demandas por automação de processos de produção, muitos deles como parte primordial em projetos de *e-government* na administração pública de Porto Alegre.

Com a utilização de uma solução de TI como o **Widflow**, é possível uma enorme economia ao orçamento público em pagamento de licenças de ferramentas proprietárias para clientes de *workflow*, em treinamento e consultoria pagos aos representantes dos fornecedores e em infra-estrutura, uma vez que o **Widflow** utiliza o mesmo ambiente que os sistemas de informação atuais elaborados e executados pela Procempa.

Com o **Widflow** e o Diagrama de Atividades da UML estendido foi possível definir um processo de desenvolvimento de aplicações de *workflow* com um

alto grau de aderência ao cenário encontrado na Procempa, a organização responsável pela execução dos projetos de automação. Eles possibilitam a larga disseminação desse tipo de solução tanto pela simplicidade quanto pelo baixo custo adicional num cenário onde os recursos para implementar soluções de TI são escassos e a proliferação desse tipo de aplicação ocorrerá rapidamente nos próximos anos.

Resumindo, como resultado do trabalho da criação do **Wideflow** e da modelagem de *workflow* com o diagrama de atividades e seus estereótipos no ambiente de engenharia de software da Procempa, podem ser salientadas as seguintes contribuições:

- a) foi criado um motor de *workflow* executando a semântica de uma máquina de estados do Diagrama de Atividades da UML;
- b) foram delimitadas construções semânticas para o Diagrama de Atividades da UML capazes de suportar requisitos fundamentais em aplicações de *workflow*;
- c) foi implementado o motor de *workflow* baseado num meta-modelo com o paradigma da orientação a objetos;
- d) foi reduzida a modelagem organizacional utilizando conceito de grupo de acesso já utilizados comumente nos sistemas de informação atuais;
- e) foram construídas aplicações de *workflow* integrando automação de processos e sistemas de informação sem invasão de domínios;
- f) foi realizada a modelagem da aplicação tomando o motor de *workflow* como arquitetura para a aplicação, num componente distribuído com uma API claramente definida;
- g) foram implementadas aplicações de *workflow* de produção modeladas com uma extensão do diagrama de atividades e implementadas com o uso de um motor de *workflow* próprio;
- h) foi desenvolvida a modelagem de dados do *workflow* em abstrações de documentos aumentando o poder de expressão das operações possíveis em torno dos dados.

Com o desenvolvimento deste projeto acreditamos que os objetivos esperados foram alcançados, contribuindo para que *workflow* se popularize de forma efetiva e atenda de forma eficaz aos processos de automação discutidos aqui, projetos de desenvolvimento de aplicações de *workflow* a serem utilizados na administração pública de Porto Alegre.

KEYWORDS

Workflow - Automation of processes - Modeling of workflow.

ABSTRACT

The present article presents the development and implementation of automation of processes in the Porto Alegre's Data Processing Company –Procempa – with expressive reduction of complexity and cost. The creation of workflow applications is made with aid of a modeling methodology using an architecture that make use of the preexisting systems development platform and an engine of workflow, the Wideflow, this last executing the semantics of a extended UML Diagram of Activities, build to support the coordination and execution of activities. This article also shows to a case study based on a pilot process whose requirements serve as base for the solution discussion.

REFERÊNCIAS BIBLIOGRÁFICAS

ARAUJO, A. DE; GOMIDE, C. F.; PETRILLO, F. **Metodologia e Projeto de Software Orientados a Objetos**. São Paulo: Ed. Érica, 2003.

BASTOS, R.M; RUIZ, D.D.A. Extending UML Activity Diagram for Workflow Modeling in Production Systems. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEMS SCIENCES, 35. 2002. **Proceedings...** Island of Hawaii: [s.n], 2002.

CASATI, F. et al. Conceptual Modeling of Workflow. In: OO-ER CONFERENCE, 1995. **Proceedings...** Gold Coast: [s.n], 1995.

ESHUIS, R.; WIERINGA, R. A real-time execution semantics for UML activity diagrams. In: FUNDAMENTAL ASPECTS OF SOFTWARE ENGINEERING (FASE), 2001. **Proceedings...** Genova: Springer, 2001. p. 76-90 (Lecture Notes in Computer Science, v. 2029)

GEORGAKOPOULOS, D. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. **ACM Distributed and Parallel Databases**, Nova Iorque, n.3, p.119-153, Mar. 1996.

JACOBSON, I.; ERICSON, M.; JACOBSON, A. **The Object Advantage – Business Process Reengineering with Object Technology**. New York: ACM Press, dezembro, 1994.

KIRTLAND, M.; **Projetando Soluções Baseadas em Componentes**. Rio de Janeiro, Campus, 1999.

MANOLESCU, D. A. **Micro-Workflow**: a workflow architecture supporting compositional object-oriented software development. 2001. Tese (Doutorado). – University of Illinois, Champaign, 2001.

OBJECT MANAGEMENT GROUP, **Unified Modeling Language Specification**, Versão 1.5, [S.l.], 2003.

SILVEIRA, M. DE L. DA; BARROS, L. C. DA S.; PEREIRA, R. B. Acesso a sistemas legados através da Internet. **Revista de Informática Pública**. Belo Horizonte, v.02, p. 147-162, Mai. 2000.

VAN DER AALST, W.M.P. et al. Advanced Workflow Patterns. In: INTERNATIONAL CONFERENCE ON COOPERATIVE INFORMATION SYSTEMS (COOPIS, 7.), 2000. **Proceedings.....**, Berlin: Springer-Verlag, 2000. p. 18-29. (Lecture Notes in Computer Science, v. 1901).

VAN DER AALST, W.M.P. et al. **Workflow Patterns**. Eindhoven, Netherlands: Eindhoven University of Technology, 2000. (Beta Working Paper Series, WP 47).

WORKFLOW MANAGEMENT COALITION. **The Workflow Reference Model**, Disponível em: <<http://www.wfmc.org>>. Acesso em: Jun. 2002.

WORLD WIDE WEB CONSORTIUM. **Extensible Markup Language (XML) 1.0 (Second Edition)**. W3C Recommendation. Oct 2000. Disponível em: <<http://www.w3c.org/XML/>>. Acesso em: Jun. 2003.

SOBRE OS AUTORES

CARLOS FRANCISCO GOMIDE

Mestre e especialista em Sistemas de Informação pela Universidade Federal do Rio Grande do Sul

Bacharel em Ciências Contábeis pela PUCRS

Professor do curso de Administração com Habilitação em Análise de Sistemas na UNIVATES, em Lajeado-RS

Supervisor de Tecnologia e Serviços na Procempa - Empresa de Tecnologia da Informação e Comunicação de Porto Alegre

Áreas de interesse: automação de processos, bussines intelligence, gestão de informação e gestão de conhecimento

JOSÉ PALAZZO M. DE OLIVEIRA

Doutor em Informática pelo Instituto Nacional Politécnico de Grenoble em 1984

Professor Titular da Universidade Federal do Rio Grande do Sul

Áreas de interesse: modelagem conceitual, ensino a distância, banco de dados, sistemas de informação, automação de escritórios, engenharia de software, sistemas distribuídos, descoberta de conhecimento, sistemas na WEB e modelagem temporal