

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Ambiente de Modelação e Configuração de Processos

Marcelo Pedro Fernandes Cerqueira

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Hugo José Sereno Lopes Ferreira (PhD, Assistant Lecturer)

Co-orientador: Ademar Manuel Teixeira de Aguiar (PhD, Assistant Professor)

28 de Junho de 2010

Ambiente de Modelação e Configuração de Processos

Marcelo Pedro Fernandes Cerqueira

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas publicas pelo júri:

Presidente: Ana Cristina Ramada Paiva Pimenta (Professor Auxiliar)

Vogal Externo: Feliz Alberto Ribeiro Gouveia (Professor Associado)

Orientador: Hugo José Sereno Lopes Ferreira (Assistente Convidado)

14 de Julho de 2010

Resumo

Esta dissertação foi desenvolvida no âmbito da disciplina de Dissertação do 5º ano do Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto (FEUP). Foi realizada em parceria com a empresa Siemens Sector Healthcare, uma empresa líder de mercado no sector da saúde.

O objectivo desta dissertação consiste na descrição de uma arquitectura assente numa filosofia de workflows, permitindo a sua modelação e configuração, e enquadrada no âmbito do desenvolvimento de uma solução orientada ao suporte das tarefas administrativas/financeiras.

Ao longo dos últimos anos a necessidade de uma gestão eficiente de processos de negócio tem crescido significativamente, devido ao enorme valor acrescentado no melhoramento dos processos de negócio das organizações.

Processos de negócio definem o método de operação de uma empresa e a maneira como ela se diferencia em relação a outras empresas competidoras. O aumento da globalização e da competitividade aumenta a necessidade de otimizar esses mesmos processos para alcançar um excelente método de operação. Uma grande parte da vantagem competitiva reside na capacidade de adaptação à mudança rápida e eficiente dos processos de negócio. Em muitos domínios de negócio, os processos podem ser automatizados e implementados num sistema computacional para facilitar a troca de informação entre pessoas e computadores.

Esta dissertação descreve a especificação e os respectivos detalhes de implementação de uma solução com a capacidade de: i) Criar e modelar processos de negócio; ii) Modificar e publicar processos de negócio; iii) Disponibilizar uma biblioteca com blocos de actividades pré-definidas.

Neste documento está descrita uma pesquisa bibliográfica sobre conceitos, notações de modelação e aplicações no domínio da modelação de processos de negócio.

A descrição da implementação do ambiente de modelação e configuração de processos inclui a descrição da utilização da tecnologia Window Workflow 4.0 como plataforma base, através da funcionalidade de *re-hosting*, adaptando o ambiente de modelação e as respectivas funcionalidades às necessidades funcionais da solução. O ambiente de modelação do Windows Workflow 4.0 é modificado de modo a permitir uma simplificação do método de atribuição de argumentos entre actividades, e o desenvolvimento de uma interface agradável e intuitiva para o utilizador final.

Abstract

The current work was developed within the subject Dissertation of the 5th year of the MSc in Computer Engineering of Faculdade de Engenharia da Universidade do Porto (FEUP). This dissertation was conducted in partnership with Siemens Sector Healthcare, a market leader in the health sector.

The purpose of this thesis is the architecture description based on a workflows philosophy, allowing its modeling and configuration, and framed in the context of developing a solution-oriented support of administrative/financial tasks.

Business processes define the company operating method and how it differs in relation to other competing companies. The globalization and competitiveness growth increases the need to optimize those processes to achieve an excellent operation method. A large part of the competitive advantage lies in the ability to adapt quickly and efficient of business processes change. In many areas, business processes can be automated and implemented in a computer system to facilitate the exchange of information between people and computers.

This dissertation consisted in specification and implementation of a solution with the ability of: i) Create and model business processes; ii) Modify and publish business processes; iii) Providing a library of pre-defined activities blocks.

Described in this paper is a literature search about concepts, notations and applications in business processes modeling domain.

The implementation description of the modeling and configuration process environment include the use of technology Window Workflow 4.0 as base platform by re-hosting feature, by adapting the modeling environment and its features to the solution functional needs.

For development work, was used the Windows Workflow 4.0 technology as a base platform, using the re-hosting functionality, adapting it to the solution functional needs. The Windows Workflow 4.0 modeling environment is modified to allow a parameter assignments simplification between activities, and developing a nice and intuitive interface for the end user.

Agradecimentos

Em primeiro lugar quero agradecer à Siemens Sector Healthcare, pela oportunidade concedida de realizar a dissertação nesta empresa.

Agradeço ao meu orientador na empresa, o Engenheiro Rodrigo Moreira, o apoio prestado, a sua disponibilidade e pela visão prática e objectiva com que me ajudou a encarar as dificuldades que foram surgindo ao longo da dissertação.

Agradeço ao meu orientador na faculdade, o Professor Hugo Ferreira, pela disponibilidade, paciência, competência e brilhantismo reveladas, assim como pelos seus importantes conselhos e a sua dedicação.

Também quero aqui deixar uma palavra de gratidão aos meus colegas de trabalho da Siemens Sector Healthcare por todo o apoio, orientação e conselhos que me concederam.

Quero agradecer aos meus pais, aos meus irmãos e à minha namorada, pelo seu apoio incondicional durante toda a minha vida, pelo exemplo de dedicação, responsabilidade e perseverança, pela compreensão, carinho e amor. Sem eles não teria chegado até este ponto e por isso esta dissertação deve ser dedicada principalmente a eles.

E por último, agradeço a todos os meus amigos e a todos aqueles que, de uma forma ou outra, contribuíram para o desenvolvimento desta dissertação, não só pelo apoio directo no trabalho, mas também no meu crescimento como pessoa.

A todos, um MUITO OBRIGADO!

Marcelo Pedro Fernandes Cerqueira

Conteúdo

Agradecimentos	iii
1 Introdução	1
1.1 Apresentação da Empresa	1
1.2 Contexto	2
1.3 Motivação e Objectivos	3
1.4 Estrutura da Dissertação	4
2 Revisão Bibliográfica	7
2.1 Conceitos gerais	7
2.1.1 Sistemas Incompletos por Natureza	7
2.1.2 Designing for Incompleteness	8
2.1.3 Sistemas de Gestão de Workflows	8
2.1.4 Modelação de Processos de Negócio	11
2.2 Notações de Modelação de Processos de Negócio	12
2.2.1 Flowcharts	13
2.2.2 Diagramas de Actividade UML	14
2.2.3 Business Process Modeling Notation	15
2.3 Tecnologias	18
2.3.1 Alfresco - JBoss jBPM	18
2.3.2 Alfresco - Activiti	21
2.3.3 Microsoft Sharepoint 2010	22
2.3.4 TIBCO Business Studio	23
2.3.5 IBM Websphere Business Modeler	24
2.3.6 BizAgi	25
2.3.7 Windows Workflow 4.0	27
2.4 Conclusão	33
3 Especificação	36
3.1 Definição do Problema	36
3.2 Visão de Alto-Nível	37
3.3 Requisitos	38
3.3.1 User Stories	39
3.3.2 Diagramas de Casos de Uso	46
3.3.3 Requisitos Não-Funcionais	46
3.4 Objectivos	46
3.5 Metodologia	48

CONTEÚDO

3.6	Estratégia	49
3.7	Conclusão	50
4	Implementação	53
4.1	Tecnologia Base	53
4.2	Notação Base	55
4.2.1	Fluxo de Dados	55
4.2.2	Actividades	59
4.3	Ambiente Gráfico	60
4.3.1	Painel de Configuração	60
4.3.2	Designer Runtime Behavior	62
4.3.3	Object Browser	64
4.3.4	Atribuição de Business Components	65
4.3.5	Navegação	65
4.4	Integração com o Workflow Engine	66
4.5	Conclusão	69
5	Conclusões e Trabalho Futuro	71
5.1	Satisfação dos Objectivos	71
5.2	Discussão	72
5.3	Trabalho Futuro	72
5.4	Conclusão Final	73
	Referências	74

Lista de Figuras

1.1	Modelo de workflow - pagamento no início do check-out	5
1.2	Modelo de workflow - pagamento no final do check-in	6
2.1	Limite do software em relação ao domínio [Fer11]	8
2.2	Características Básicas dos WMS [Hol95]	11
2.3	Tipos de modelos processuais [Von07]	12
2.4	Diagrama utilizando notação Flowchart [Flo11b]	14
2.5	Tipos de nós da notação dos diagramas de actividade UML [Esh02]	15
2.6	Exemplo de diagrama de actividade UML	16
2.7	Elementos principais de um diagrama BPMN	18
2.8	Exemplo de diagrama utilizando a notação BPMN	19
2.9	Visualização gráfica de workflow utilizando o JBoss jBPM	21
2.10	Visualização gráfica da modelação no Microsoft Office Visio	23
2.11	Exemplo de Diagrama BPMN utilizado no TIBCO Business Studio	24
2.12	Modelo de workflow utilizando o estilo BPMN no IBM Websphere Business Modeler [IBM11]	27
2.13	Modelo Sequencial do Windows Workflow 4.0 [Mil11]	28
2.14	Modelo Flowchart do Windows Workflow 4.0 [Mil11]	29
2.15	Quadro-Resumo das características das tecnologias	34
3.1	Diagrama da arquitectura do Configurador	38
3.2	Diagrama da arquitectura do componente de gestão de workflows	39
3.3	Diagrama de Casos de Uso	47
3.4	Diagrama de Casos de Uso (continuação)	51
3.5	Processo de desenvolvimento iterativo e incremental Scrum [Sof11]	52
4.1	Fluxo de controlo de um workflow	56
4.2	Fluxo de dados de um workflow	57
4.3	Fluxo de dados de um workflow utilizando os argumentos globais	58
4.4	Representação gráfica de uma actividade	59
4.5	Representação gráfica de um workflow anteriormente modelado e reutilizado no modelo de outro workflow	60
4.6	Painel de configuração na área do DRB no momento da modelação	62
4.7	Painel de configuração na área do DRB no momento da reutilização	63
4.8	Organização em árvore dos argumentos das actividades/workflows do Object Browser	64
4.9	Representação do Flowchart visível no ambiente de modelação	67

LISTA DE FIGURAS

4.10	Representação do Flowchart lógico representativo do ambiente de modelação	68
4.11	Acesso ao objecto <i>Message</i> para integração com o fluxo de dados do <i>Workflow Engine</i>	69
4.12	Fluxo de execução do workflow	70

Lista de Tabelas

2.1	Informação detalhada sobre os elementos da notação BPMN [Gro11a] . . .	17
2.2	Funções dos vários modos de modelação do IBM Websphere Business Modeler [IBM11]	26
3.1	User Stories	40

Abreviaturas e Símbolos

ACSS	Administração Central de Serviços de Saúde
AOM	Adaptive Object Models
API	Application Programming Interface
BPD	Business Process Diagram
BPDM	Business Process Definition Metamodel
BPEL	Business Process Execution Language
BPEL4WS	Business Process Execution Language For Web Services
BPM	Business Process Modeling
BPMI	Business Process Management Initiative
BPMN	Business Process Modeling Notation
BPR	Business Process Re-engineering
CAD	Computer-Aided Design
DRB	Designer Runtime Behavior
ECM	Enterprise Content Management
EPR	Electronic Patient Record
GDH	Grupos de Diagnóstico Homogéneo
GPD	Graphical Process Designer
HIS	Health Information Systems
IDE	Integrated Development Environment
JPA	Java Persistence API
JAR	Java Archive
JTA	Java Transaction API
MVVM	Model-View-ViewModel
OMG	Object Management Group
PNG	Portable Network Graphics
ROI	Return Of Investment
SCA	Service Component Architecture
TI	Tecnologias de Informação
UML	Unified Modeling Language
UI	User Interface
WCF	Windows Communication Foundation
WF	Windows Workflow
WF4	Windows Workflow 4.0
WF3	Windows Workflow 3.0

LISTA DE TABELAS

WFM	Workflow Management
WfMC	Workflow Management Coalition
WMS	Workflow Management Systems
WPF	Windows Presentation Foundation
WSDL	Web Services Description Language
WWF	Windows Workflow Foundation
WWR	Windows Workflow Runtime
XAML	Extensible Application Markup Language
XML	Extensible Markup Language
XOML	Extensible Object Markup Language
XSD	XML Schema Definition

Capítulo 1

Introdução

1.1 Apresentação da Empresa

Com 500 centros de produção em 50 países e presença em 190 países a Siemens está representada em todo o mundo. Em Portugal, a Siemens S.A. dispõe de duas unidades fabris, centro de investigação, desenvolvimento de software (Lisboa e Porto) e presença em todo o país, através dos seus parceiros e das suas instalações. A empresa está desde 2008 organizada em três grandes sectores de actividade: Industry, Energy e Healthcare.

O Sector Industry dispõe de soluções para a indústria nas vertentes de produção, transporte e edifícios, segmentando-se em cinco áreas: Industry Automation and Drive Technologies, Building Technologies, Industry Solutions, Mobility e OSRAM.

O Sector Energy disponibiliza produtos e soluções para a geração, transmissão e distribuição de energia eléctrica, segmentando-se em seis áreas: Fossil Power Generation, Renewable Energy, Oil gas, Energy Service, Power Transmission e Power Distribution.

O Sector Healthcare oferece um conjunto de produtos inovadores e soluções integradas bem como serviços e consultadoria na área da saúde, segmentando-se em três áreas: Imaging IT, Workflow Solutions e Diagnostics.

A área Imaging IT disponibiliza sistemas de imagem para diagnóstico precoce e intervenção, bem como para prevenção efectiva, nomeadamente Sistemas de ressonância magnética (MR), Sistemas de tomografia axial computadorizada (CT), Sistemas de radiografia, Sistemas angiográficos digitais, Sistemas de tomografia por emissão de positrões (PET/CT) e tomografia por emissão de fóton único (SPECT e SPECT/CT), Unidades de ecografia, entre outros. Todos os sistemas estão interligados por tecnologias de informação de elevada performance possibilitando uma optimização dos processos a nível dos prestadores de cuidados de saúde (sistemas de gestão hospitalar como o Soarian, sistemas de processamento de imagem como o Syngo, e tecnologias *knowledge-based* como auxiliares de diagnóstico).

A área Workflow Solutions disponibiliza soluções globais para especialidades como a cardiologia, a oncologia e a neurologia. Esta área fornece ainda soluções, por exemplo, para saúde da mulher (mamografia), a urologia, a cirurgia e a audiologia, englobando igualmente a vertente de consultadoria e soluções globais. Simultaneamente, a área de Workflow Solutions engloba a prestação de serviços pós-venda e gestão de clientes.

A área Diagnostics encerra a vertente de diagnóstico in-vitro, incluindo imunodiagnóstico e análise molecular. As soluções da área vão desde os aplicativos *point-of-care* até à automatização de grandes laboratórios.

Desta forma, a Siemens Sector Healthcare é hoje a primeira empresa a nível mundial a disponibilizar um portefólio integrado de tecnologias que permitem responder a todas as fases do ciclo de cuidados de saúde.

Em Portugal, o Sector Healthcare da Siemens S.A. é um dos líderes de mercado no ramo dos cuidados de saúde, reconhecido pelas suas competências e força de inovação em diagnóstico e tecnologias terapêuticas, assim como engenharia de conhecimento, incluindo tecnologias de informação e integração de sistemas [SA11].

1.2 Contexto

Presentemente, as oportunidades que emergem no mercado da saúde e em que a Siemens SA, como instituição de referência e prestígio reconhecido, tende a responder, são cada vez mais globalizantes e assentes nas tecnologias de informação, nomeadamente no que respeita ao suporte do registo electrónico de paciente (EPR - *Electronic Patient Record*). O registo electrónico de paciente consiste num conjunto ordenado de registos contendo toda a informação clínica (e.g. história clínica, exames objectivo, diagnósticos, procedimentos, terapêuticas), administrativa (e.g. identificação do paciente, data de nascimento, laços familiares, demografia) e financeira (e.g. dados financeiros, histórico de transacções financeiras) recolhida junto de um paciente [Rog00]. Pela multidisciplinaridade da informação englobada, o sistema administrativo/financeiro assume um papel central na integração entre as envolventes internas (intra-departamentais) e envolventes externas (e.g. seguradoras, entidades reguladoras) existentes e/ou associadas às instituições de saúde.

Portugal encontra-se numa situação particular em relação a outros países europeus, porque a maioria dos hospitais públicos assentam a sua gestão administrativa/financeira sobre o Sistema Integrado de Informação Hospitalar (SONHO), desenvolvido e distribuído pela Administração Central de Serviços de Saúde (ACSS). O SONHO é uma plataforma com mais de 20 anos e que apresenta algumas limitações ao nível da interface gráfica, usabilidade, integração com outros sistemas, normalização e utilização de *standards* da área da saúde, falta de flexibilidade na adaptação a novos processos de negócio e inexistência de suporte multi-entidade [AL11] [eAMB05].

Um sistema adequado à actual realidade das vertentes de gestão administrativa/financeira no contexto de uma organização de saúde, deve ser capaz de oferecer a possibilidade de modificação expedita dos modelos de negócio pré-definidos promovendo, simultaneamente, a auto-aprendizagem por parte do utilizador, a partilha de tarefas entre utilizadores (trabalho colaborativo), a utilização de modelos dinâmicos de apresentação (*wizards*), a utilização de novos dispositivos físicos, entre outras. Estes conceitos não são suportados pelo SONHO por compreensivas limitações da arquitectura base.

1.3 Motivação e Objectivos

A constante evolução das organizações aliada ao inevitável aumento da complexidade dos processos organizacionais associados, determina a necessidade de dispor de sistemas flexíveis, orientados a processos dinâmicos e capazes de responder eficazmente à mutabilidade do sector. Neste âmbito, a Siemens Sector Healthcare encontra-se a desenvolver uma solução clínica de última geração, diferenciadora em relação às soluções existentes no mercado.

Esta solução está focada nas actividades administrativas/financeiras das organizações de saúde. Tem como objectivo fundamental o suporte das tarefas administrativas que suportam o fluxo de processos administrativo/financeiros associados a um paciente. A solução está pensada para ser integrada transparentemente no panorama heterogéneo e disperso dos sistemas TI de um prestador de cuidados de saúde, usufruindo de uma arquitectura modular (módulos funcionais) e conseqüente adaptabilidade à diversidade existente nos serviços prestados. Consegue-se assim fornecer de uma forma transversal uma infra-estrutura de serviços e módulos integrados capazes de suportar globalmente os processos clínicos, administrativos e financeiros. A implementação desta solução possuiu uma motivação fundamentalmente alicerçada nos seguintes conceitos:

- A eliminação do desfasamento existente entre sistemas administrativos e sistemas clínicos, que se traduz na evidente necessidade de incorporação de novos mecanismos de integração da informação e colaboração entre sistemas;
- O aumento da transparência entre processos de negócio e sistemas de informação;
- O incremento da flexibilidade e adaptabilidade a novos fluxos de trabalho e regras de negócio (e.g. modificações do foro legislativo, normas e requisitos de qualidade);
- A garantia de registo e integridade da informação temporal durante o período em que uma actividade se encontra sob execução;
- A redução de custos de manutenção e de desenvolvimento adicional;

- A disponibilização de serviços que possibilitem a comunicação e interacção com os pacientes utilizando a Internet;
- A monitorização em tempo-real de todos os processos da instituição de saúde;
- A monitorização em tempo-real de um processo específico de um qualquer utente;
- O aumento significativo da usabilidade e qualidade na apresentação gráfica da informação, tornando a experiência de utilização agradável, promovendo o treino *on-the-job* e potenciando a eficácia dos processos associados;

Esta dissertação descreve a implementação de uma arquitectura assente numa filosofia de workflows (automatização de processos de negócio), permitindo a sua modelação e configuração, e enquadrada no âmbito do desenvolvimento da solução administrativo/financeira referida anteriormente. Ou seja, descreve a implementação de um ambiente de modelação de configuração de processos de negócio.

Para um melhor entendimento dos assuntos relacionados com a modelação de processos de negócio, são descritos conceitos, notações de modelação e aplicações, com um conjunto de funcionalidades semelhantes à solução apresentada.

O objectivo do ambiente de modelação e configuração de processos é agradar o utilizador final em termos de facilidade e intuitividade de utilização. Esta simplificação engloba a modificação de aspectos relacionados com a modelação de processos de negócio e da atribuição de argumentos entre actividades (fluxo de dados). Este ambiente permite ainda a total abstracção do desenvolvimento de código, e a correcta definição de um modelo adaptável a diferentes tipos de processos organizacionais.

Um exemplo de um processo dinâmico, no contexto de um serviço hospitalar, é o *check-in* (conjunto de actividades feitas no anteriormente à admissão) e o *check-out* (actividades envolvidas no processo de alta administrativa). Neste exemplo, a actividade de pagamento reflecte a flexibilidade do processo, permitindo que a actividade de pagamento seja realizada quer numa quer noutra fase de acordo com as necessidade do serviço hospitalar, como ilustram as figuras 1.1 e 1.2.

1.4 Estrutura da Dissertação

Para além da introdução, esta dissertação contém mais cinco capítulos. No capítulo 2 é descrita uma revisão bibliográfica, evidenciando conceitos gerais, notações de processos de negócio e uma descrição sobre as diferentes tecnologias existentes no mercado, no domínio da modelação dos processos de negócio.

O capítulo 3 descreve uma análise teórica do problema, envolvendo a sua definição, requisitos, metodologias, objectivos e estratégias necessárias para a implementação do ambiente de modelação e configuração de processos.

Introdução

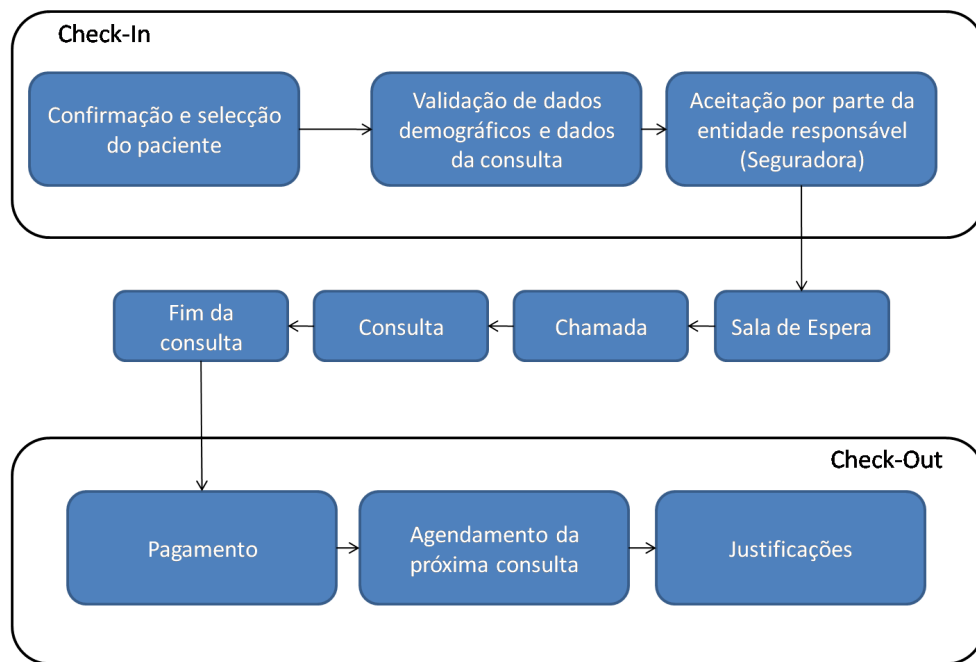


Figura 1.1: Modelo de workflow - pagamento no início do check-out

O capítulo 4 descreve os aspectos de implementação com maior profundidade, e no capítulo 5 é apresentada uma breve análise final dos resultados obtidos, com uma abordagem ao respectivo trabalho futuro.

Introdução

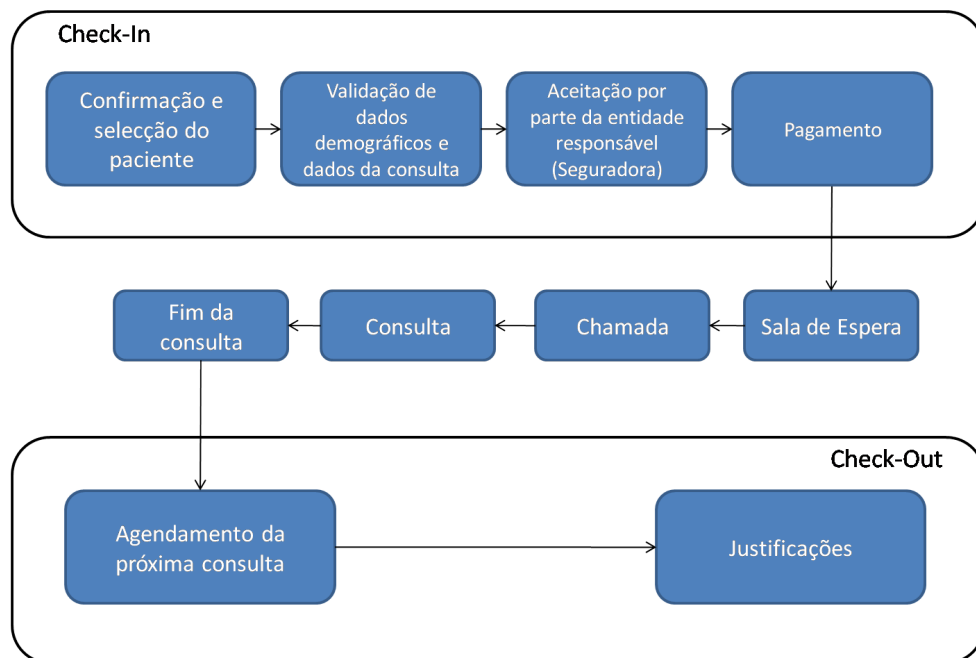


Figura 1.2: Modelo de workflow - pagamento no final do check-in

Capítulo 2

Revisão Bibliográfica

2.1 Conceitos gerais

Neste sub-capítulo são apresentados conceitos gerais relacionados com a modelação de processos de negócio.

2.1.1 Sistemas Incompletos por Natureza

Sistemas que necessitam de alteração constante, para as funcionalidades estarem em conformidade com as necessidades da organização, designam-se sistemas incompletos por natureza.

Estes sistemas tornam-se inúteis quando implementam soluções em que os requisitos mudam mais rapidamente que a própria implementação. Neste sentido, será extremamente difícil e dispendioso manter este tipo de sistemas quando os processos de uma organização estão em constante mudança.

Um sistema que auxilia e possibilita a concretização dos processos apresentados nas figuras 1.1 e 1.2, trata-se de um sistema incompleto por natureza, porque é alvo de uma constante evolução e adaptação às novas realidades, tanto a partir do momento em que é projectado como durante a sua utilização.

Como ilustra a figura 2.1 o software não coincide na totalidade com a região do domínio. Este factor deve-se à imprecisão dos domínios, à rígida estrutura artificial que o software impõe, e à mudança constante da própria realidade.

Em ambientes de constante mudança, adoptar um método de concepção que tenta fixar limites, objectivos e finalidades torna-se potencialmente perigoso. Considerando que tal abordagem pode produzir um sistema que apesar de ser óptimo num determinado período de tempo e dada a mudança contínua, o sistema rapidamente torna-se obsoleto com o tempo [Fer11] [FCYA10] [GJT07].

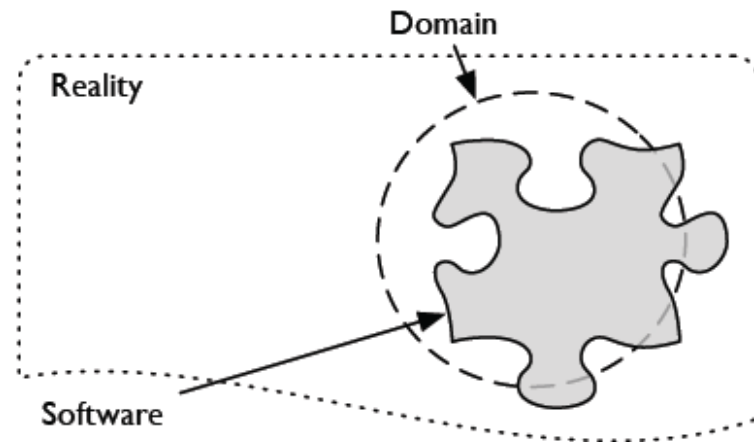


Figura 2.1: Limite do software em relação ao domínio [Fer11]

2.1.2 Designing for Incompleteness

Para solucionar problemas evidentes em sistemas incompletos por natureza surge um raciocínio bastante eficaz: *designing for incompleteness*. Este conceito é utilizado para designar a necessidade de adoptar um mecanismo que permitirá responder adequadamente a essa incompletude do sistema.

Sistemas que são incompletos por natureza devem ser desenvolvidos para se ajustarem facilmente às mudanças efectuadas nos processos. É fácil constatar que se um processo é considerado dinâmico, então esse dinamismo precisará de ser oferecido pelo sistema que o implemente.

Assim, em sistemas incompletos por natureza devem ser criados mecanismos geradores de sequências dinâmicas de soluções.

No âmbito desta dissertação, e de acordo com este raciocínio, surge a necessidade da construção de sistemas capazes de implementar processos dinâmicos, e que correspondem às mudanças processuais das organizações, respondendo adequadamente às suas necessidades [Fer11] [GJT07].

2.1.3 Sistemas de Gestão de Workflows

Sistemas que implementam processos de negócio categorizam-se como sistemas incompletos por natureza, porque necessitam estar constantemente em mudança para corresponderem às necessidades processuais das organizações. Será necessária uma solução que permita que esses sistemas sejam *Designing for Incompleteness*. Essa solução denomina-se por gestão de workflows, e os sistemas que ajudam a sua função denominam-se por Sistemas de Gestão de Workflows.

Um workflow é definido como um conjunto de tarefas organizadas para a realização de um processo de negócio (e.g. processamento de ordens de compra por telefone, processamento de reclamações dos seguros). As tarefas envolvidas poderão ser executadas por um ou mais sistemas de software, uma ou mais equipas de indivíduos, ou então pela combinação entre estes.

As tarefas humanas incluem a interacção com computadores aproximadamente (e.g. utilizando a inserção de comandos) ou folgadoamente (e.g. utilizando computadores apenas para indicar o progresso de uma ou várias tarefas). Exemplos de tarefas podem ser a actualização de um ficheiro ou base de dados, geração ou envio de uma conta, ou colocação de um cabo de rede. Adicionalmente ao conjunto de tarefas, um workflow define a ordem de invocação de uma tarefa ou a condição sobre a qual essas tarefas devem ser invocadas. Define também a sua sincronização e o fluxo de informação trocado pelas tarefas [GHS95].

O Workflow Management Coalition (WfMC), um consórcio formado para definir *standards* para a interoperabilidade de sistemas de gestão de workflows, define um workflow da seguinte forma:

“Automatização de um processo de negócio, no seu todo ou apenas uma parte, do qual documentos, informação e tarefas são passados de um participante para outro, através de uma acção e de um conjunto de regras processuais”
[AtH03].

A definição de workflow especifica quais as tarefas a serem executadas e qual a sua ordem de execução. Desde que as tarefas sejam executadas numa ordem específica, torna-se útil identificar condições que correspondem às suas dependências causais. Assim, cada tarefa necessitará de uma pré-condição (que será avaliada antes da execução da tarefa) e uma pós-condição (que será avaliada após a execução da tarefa) [Aal97].

A gestão de workflows (Workflow Management - WFM) é um conceito utilizado na engenharia de processos de negócio. Trata-se de:

- Definir workflows, ou seja, descrever os aspectos de um processo, no controlo e ordenação da execução das suas tarefas, e possivelmente as habilidades necessárias dos utilizadores e dos sistemas de informação para a realização de cada tarefa;
- Fornecer uma concepção e implementação rápida dos processos através da mudança de sistemas de informação;

Para suportar efectivamente WFM, as organizações deverão evoluir constantemente os seus ambientes computacionais para um novo ambiente distribuído, com as seguintes características:

- Integração e interoperabilidade entre componentes *loose coupling*;

- Suporte à aplicação de workflows, que correspondem a implementações de processos de negócios;
- Garantia da exactidão e a confiança na presença de concorrência e avarias;
- Suporte à evolução, substituição e adição de outras aplicações de workflows e componentes [GHS95];

Um workflow está várias vezes associado à Business Process Re-engineering (BPR), que é uma área que preocupa-se com a avaliação, análise, modelação, definição e posteriormente com a implementação dos principais processos de negócio organizacionais. Apesar de nem todas as actividades BPR resultarem em implementações de workflows, a tecnologia é muita das vezes uma solução apropriada, uma vez que permite a separação da lógica dos processos de negócio e do suporte operacional das TI [Hol95]. Deste modo, surge a necessidade de criar sistemas de gestão de workflows (Workflow Management Systems - WMS). O WfCM define um sistema de gestão de workflows da seguinte forma:

”Sistema que define, gere e executa workflows através da execução de software, cuja ordem de execução é conduzida por uma representação lógica do workflow” [Hol95]

Apesar de haver sempre alguma variedade entre WMS, existem características comuns a todos, que fornecem a base para a integração e capacidade de interoperabilidade entre diferentes produtos. Em alto-nível, todos os WMS podem ser caracterizados por disponibilizar suporte a três áreas funcionais:

- **Funções em tempo de compilação** - orientada para a definição e modelação do workflow e das suas actividades constituintes;
- **Funções de controlo em tempo de execução** - orientada para a gestão dos workflows num ambiente operacional e da sequencialização das várias actividades a serem executadas;
- **Interacções em tempo de execução** - orientada para a interacção com os utilizadores finais e ferramentas TI no processamento dos vários passos de um workflow [Hol95].

A figura 2.2 ilustra as características básicas dos WMS e as relações entre as suas principais funções.

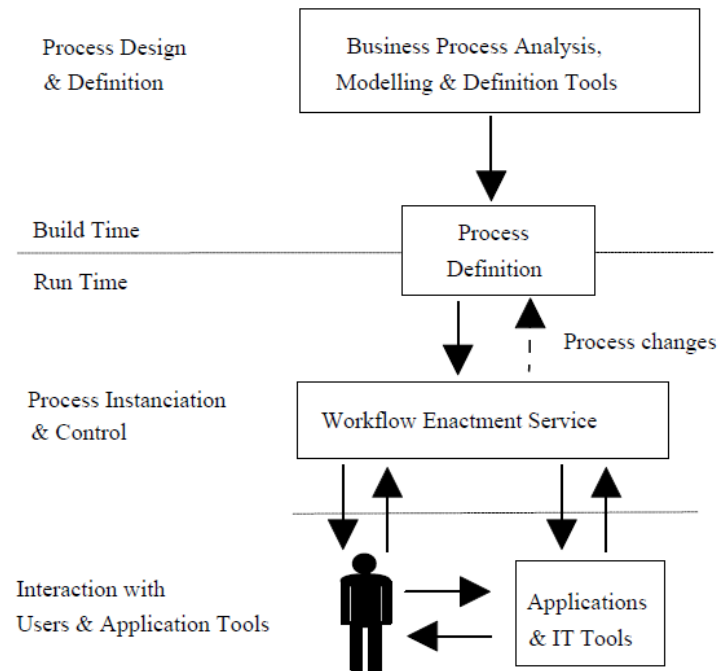


Figura 2.2: Características Básicas dos WMS [Hol95]

2.1.4 Modelação de Processos de Negócio

Para representar graficamente os workflow existem inúmeras notações gráficas, ou seja, definições de modelos que representam as actividades e o respectivo fluxo de controlo do processo de negócio. Assim, um modelo de processo de negócio é uma representação do mesmo numa forma que suporta manipulação automatizada, como a modelação, através de um sistema de gestão de workflows.

Um método de modelação de processos de negócio (*Business Process Modeling - BPM*) pode ser caracterizado como um instrumento que permite a criação de um modelo formal e visual, utilizado para representar um processo com precisão suficiente para a sua simulação e controlo. A abordagem visual da modelação permite aumentar a capacidade e a clareza, facilitando a comunicação entre *stakeholders* [Von07].

A figura 2.3 ilustra três tipos diferentes de modelos de workflows.

O principal objectivo de um modelo funcional (*Functional Model*) é a identificação da arquitectura de um processo de negócio, bem com a identificação dos processos dos clientes e produtos. Engloba saber quais os processos que são utilizados por uma organização, e qual a estrutura dos mesmos.

Um modelo de objecto (*Object Model*) identifica a estrutura estática de todas as entidades (objectos) que são essenciais para a aplicação do processo. Este modelo tenta capturar todos os objectos activos responsáveis pela execução das actividades e objectos

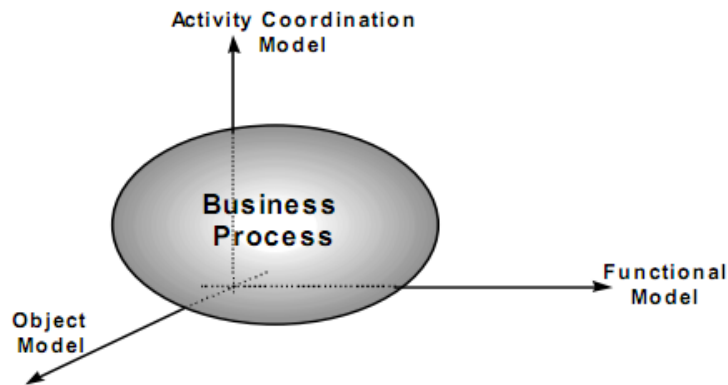


Figura 2.3: Tipos de modelos processuais [Von07]

passivos, para que possam ser percebidos como produtos materiais ou documentos que serão manipulados pelo processo.

O modelo de coordenação de actividades (Activity Coordination Model) é baseado nos anteriores modelos e o seu objectivo é mostrar como o processo será decretado, ou seja, especifica as interacções entre objectos (activos e/ou passivos) e define a sincronização entre actividades [Von07].

Para tornar a modelação de workflows mais intuitiva para o utilizador, é geralmente utilizada uma interface gráfica. Essa interface gráfica, além de possuir a uma notação gráfica do modelo, possibilita o *drag-and-drop* dos componentes que o constituem, que podem ser acedidos através de painéis de configuração [SZ05].

2.2 Notações de Modelação de Processos de Negócio

Produzir um modelo de processo de negócio é uma peça fundamental para a compreensão do mesmo. Modelos servem dois propósitos fundamentais: facilitam a comunicação entre *stakeholders* por serem recursos visuais vitais para a compreensão e entendimento dos processos de negócio, e actua como uma ferramenta de validação dos mesmos. A utilização de um modelo adequado torna mais simples a identificação de diversas lacunas existentes no processo de negócio.

Existe um vasto número de notações BPM que permitem representar graficamente o modelo pretendido, ficando mais simples para a equipa interpretar e desenvolver um modelo padronizado. Ang Che [Ang09] define linguagens/notações de processos de negócio da seguinte forma:

”Linguagem de modelação de processos de negócio é uma linguagem de domínio específico que descreve processos de negócio de uma maneira perceptiva para humano e/ou para a máquina.”

Algumas notações de modelação de processos de negócio conhecidas são:

- Flowcharts;
- Diagramas de actividade UML;
- Business Process Modeling Notation;

Neste sub-capítulo serão apresentadas as principais características destas notações.

2.2.1 Flowcharts

Flowcharts permitem separar um processo em eventos individuais ou actividades, e apresentá-los de forma abreviada, mostrando as suas relações lógicas. Define-se como uma representação simbólica de um processo. Cada passo no processo é representado por um diferente símbolo, contendo uma curta descrição, e estão ligados por setas que representam a direcção do fluxo de controlo do processo [Flo11a].

Existem duas principais funções. A primeira função é a possibilidade de ser utilizado para analisar um processo complexo, separando o processo em pequenos passos. Serve de base a uma posterior discussão, possibilitando a identificação de pontos onde alguns dados podem ser recolhidos e analisados, identifica ineficiências no processo, e permite facilitar a explicação do processo a outras pessoas. A outra função é a sua utilização para definir um processo ou projecto a ser implementado, tornando-se bastante útil por expor claramente os passos que devem ser analisados, identificando potenciais problemas na implementação, e clarificando a separação de responsabilidades nas diferentes partes do processo [Tip11] [Tec11].

A figura 2.4 ilustra um exemplo de um processo utilizando a notação em *flowchart*.

Existe um enorme desacordo na comunidade mundial em perceber a denominação de *flowcharts*, porque existir um vasto conjunto de nomes que são utilizados para os designar, nomeadamente: *process flow chart*, *process map*, *business process model*, *process flow diagram*, *workflow diagram*, *business flow diagram*, ou apenas *flow diagram*. Cada um destas designações são aplicadas em vários contextos. Por exemplo, o termo *process map* é várias vezes utilizado para referir *flowcharts* que trocam informação (*inputs* e *outputs*) em cada passo do processo. *flow diagram* é frequentemente um sinónimo de *data flow diagram*, que é uma representação do fluxo de informação num programa de computador. *Business Process Modeling* (BPM), como veremos mais à frente, é normalmente uma abordagem mais sofisticada e detalhada para a modelação de processos, utilizada na engenharia de processos de negócios [Flo11a].

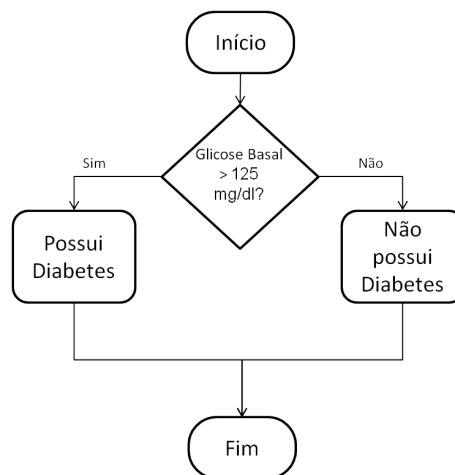


Figura 2.4: Diagrama utilizando notação Flowchart [Flo11b]

Os quatro símbolos mais utilizados nos *flowchart* são: Terminador, Processo, Decisão e Conector. De facto, se forem utilizados símbolos adicionais coloca-se em causa o entendimento dos *stakeholders* no que está representado, necessitando de informação adicional ao diagrama para ficar mais perceptível.

Em *flowcharts*, a forma e a informação que está no interior da actividade indica qual a sua função. Por exemplo, para designar uma entrada num *flowchart* normalmente é utilizada a palavra *Start* ou *Begin*, e para indicar um ponto de saída deverá ser utilizadas as palavras *End*, *Exit* ou *Return*. Na maior parte dos *flowcharts*, o rectângulo é a forma mais comum para representar uma tarefa, actividade, acção ou operação.

Para a leitura de um *flowchart* é utilizada a orientação das setas que ligam as actividades, que determinam o fluxo de controlo do diagrama. Numerar as formas é uma maneira opcional de desenhar o *flowchart*, e torna-se útil para referenciar uma actividade numa discussão em grupo. Um símbolo de decisão é um elemento que coloca uma questão na sua avaliação. A resposta à questão determina a próxima seta que será executada do conjunto de setas que saem do símbolo de decisão [Flo11b].

2.2.2 Diagramas de Actividade UML

Os diagramas de actividade são definidos pelo UML (*Unified Modeling Language*), e possui várias semelhanças com os *flowchart*.

Um diagrama de actividade é um grafo dirigido, constituído por nós e arestas direccionadas, capazes de representar o comportamento de um sistema. Um nó representa o estado de um sistema e é constituído por sete tipos, como ilustra a figura 2.5.

O nó de actividade atómica caracteriza-se pela espera da terminação da actividade, que corresponde a um passo no processo envolvido. Contrariamente, o nó de espera

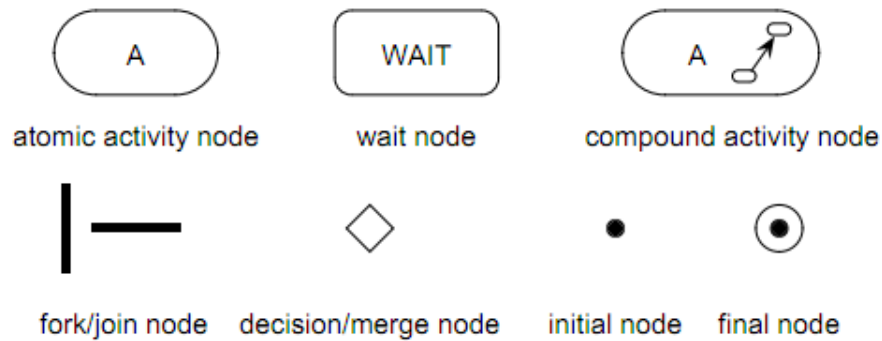


Figura 2.5: Tipos de nós da notação dos diagramas de actividade UML [Esh02]

caracteriza-se pela espera da ocorrência de um evento (e.g. quando o cliente envia alguma informação e espera pela resposta), e é bastante utilizado para sincronização. O nó de actividade composta permite executar várias actividades sequencialmente sendo expressadas no seu interior, e só será terminada após a conclusão das mesmas.

Uma actividade pode dividir-se em vários fios de execução. Para a união/separação desses fios de execução é recomendada a utilização do nó de união/bifurcação. A entrada de um nó de decisão corresponde apenas a um fio de execução, e será seleccionada a próxima actividade baseando-se no resultado de uma condição lógica. O nó inicial e o nó final correspondem, respectivamente, ao início e fim da execução do modelo do processo [Mod11] [Esh02] [DH01].

A figura 2.6 representa um processo de decisão utilizando a notação dos diagramas de actividade em UML.

2.2.3 Business Process Modeling Notation

A Business Process Management Initiative (BPMI) desenvolveu uma notação denominada por Business Process Modeling Notation (BPMN). Adoptada como padrão Object Management Group (OMG) em 2006, tornou-se um complemento de outro padrão OMG relacionado com processos de negócio, denominado Business Process Definition Meta-model (BPDM) [Gro11b].

O objectivo do BPMN é dar suporte à gestão de processos de negócio, através de uma notação bastante intuitiva na representação de processos complexos. A especificação BPMN também fornece um mapeamento entre os gráficos da notação e as construções subjacentes das linguagens de execução [RIRG06].

O conjunto de elementos de modelação BPMN permite um desenvolvimento simples de *Business Process Diagram (BPD)*, que é um diagrama baseado numa técnica bastante similar aos *flowchart* e a diagramas de actividade, através de um vasto conjunto de elementos gráficos. Existem quatro elementos básicos:

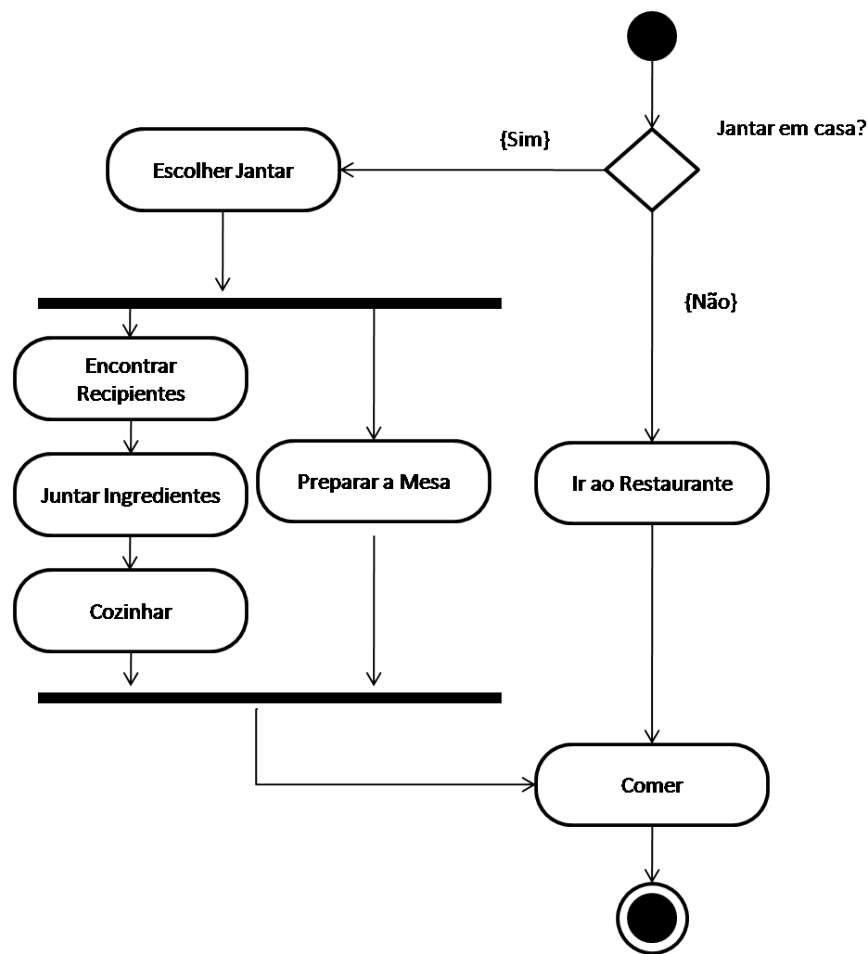


Figura 2.6: Exemplo de diagrama de actividade UML

- **Objectos de fluxo** - constituídos apenas por três elementos (eventos, actividades e *Gateways*), sendo desnecessário o reconhecimento de um vasto número de formas geométricas;
- **Objectos de conexão** - necessários para conectar objectos de fluxo e produzir o esqueleto básico de um processo de negócio;
- **Swimlanes** - fornecem a possibilidade de organizar actividades em categorias visuais separadas, e ilustrar diferentes capacidades ou responsabilidades;
- **Artefactos** - fornecem alguma flexibilidade na expansão da notação básica, e a possibilidade de adicionar contexto apropriado [Gro11a].

A tabela 2.1 explica com maior pormenor os diferentes elementos utilizados na notação BPMN. Na Figura 2.7 estão representados cada um desses elementos da tabela.

Utilizando os elementos identificados na tabela 2.1 é possível criar um diagrama que melhor se adequa às necessidades processuais. Consegue-se constatar que existem várias

Tabela 2.1: Informação detalhada sobre os elementos da notação BPMN [Gro11a]

Elemento	Tipo	Função
Evento	Objecto de Fluxo	Representa algo que acontece durante o progresso do processo de negócio. Existem três tipos de eventos: Início, Intermediário e Final.
Actividade	Objecto de Fluxo	Utilizado como termo genérico para uma tarefa ou acção. Existem dois tipos de actividades: tarefa e sub-processo. O sub-processo é distinguido da tarefa por possuir um pequeno sinal '+' no fundo da forma, e corresponde a um conjunto de tarefas.
<i>Gateways</i>	Objecto de Fluxo	Utilizado para controlar divergências e convergências num fluxo sequencial. Tradicionalmente diz-se que determina decisões.
Fluxo sequencial	Objecto de Conexão	Utilizado para definir a ordem de execução das actividades.
Fluxo de mensagem	Objecto de Conexão	Utilizado para definir o fluxo de mensagens enviadas e recebidas entre dois participantes no processo.
<i>Pool</i>	<i>Swimlane</i>	Representa um participante num processo, e também actua como contentor gráfico separando o conjunto de actividades de outros <i>Pools</i> .
<i>Lane</i>	<i>Swimlane</i>	Representa uma sub-partição de um <i>Pool</i> e é utilizado para organizar e categorizar actividades.
Objectos de dados	Artefacto	Representa os dados necessários ou produzidos pelas actividades.
Grupo	Artefacto	É utilizado para fazer um agrupamento, que pode ser utilizado para documentação ou para efeitos de análise, não tendo qualquer influência no fluxo sequencial.
Anotação	Artefacto	Utilizado para adicionar informação textual ao diagrama BPMN.

semelhanças com as notações *flowchart* e diagramas de actividade UML por tratar-se de uma evolução dos mesmos. A figura 2.8 ilustra um processo de encomenda de um produto utilizando a notação BPMN.

Os objectos gráficos BPMN podem ser mapeados para Business Process Execution Language para Web Services (BPEL4WS v1.1), que é um *standard* para a execução de processos. Assim, o processo modelado através da notação BPMN pode ser exportado para BPEL4WS e executado em ferramentas de automação ou motores de workflow.

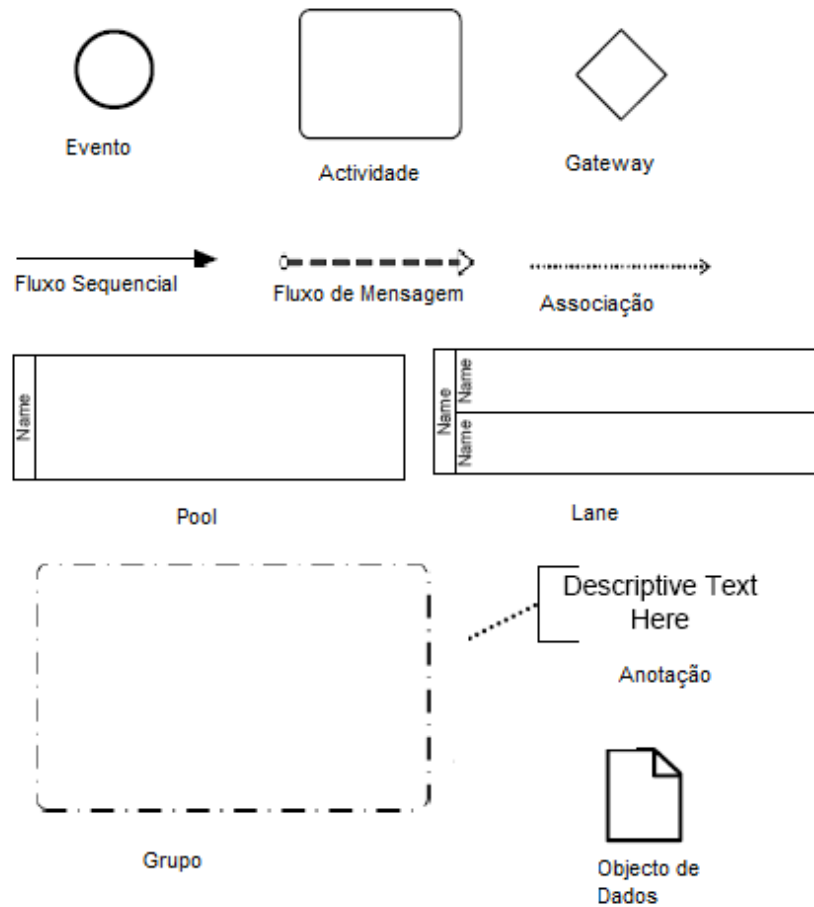


Figura 2.7: Elementos principais de um diagrama BPMN

2.3 Tecnologias

Neste sub-capítulo são apresentadas tecnologias actuais de alguma forma relacionadas com a modelação de processos de negócio. Existem várias tecnologias que foram desenvolvidas para este fim, porém só são descritas algumas delas.

2.3.1 Alfresco - JBoss jBPM

O JBoss jBPM é um motor de workflows lançado pelo JBoss Community que permite modelar, executar e monitorizar processos de negócio em notação BPMN 2.0. JBoss jBPM dá suporte a processos adaptativos e dinâmicos que necessitam de flexibilidade para modelar situações complexas e próximas da vida real, que dificilmente seriam descritas usando processos rígidos. A complexa lógica de negócios pode ser modelada como uma combinação de processos de negócio com regras de negócio e processamento de eventos complexos.

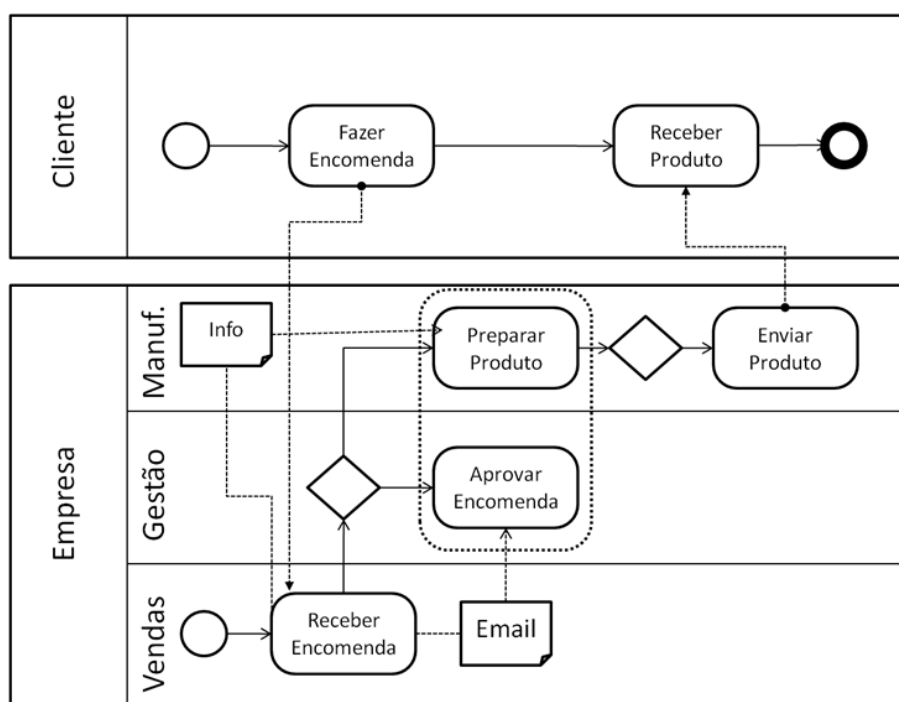


Figura 2.8: Exemplo de diagrama utilizando a notação BPMN

O Alfresco é uma solução *open-source* e multi-plataforma que fornece um serviço de gestão de conteúdos empresariais (Enterprise Content Management - ECM). Agrega as vantagens da utilização de uma ferramenta *open-source* com a estabilidade de uma verdadeira plataforma empresarial [Alf09]. O Alfresco fornece também a possibilidade de implementar workflows avançados para automatizar processos de negócio complexos. Workflows avançados são executados pelo motor incorporado JBoss jBPM.

Esta ferramenta tornou-se rapidamente uma alternativa para a gestão de documentos, arquivos, colaboração e conteúdos Web de várias empresas, que procuram um ECM de fácil instalação, simples de utilizar e custo de manutenção efectivo.

O Alfresco tem duas opções para a implementação de workflows: (i) simples e (ii) avançados. Uma das vantagens da implementação de workflows simples no Alfresco passa pela configuração sem qualquer habilidade técnica dos utilizadores. Um ficheiro que contenha um workflow simples dispõe dos passos "avançar" ou "retroceder". Quando um utilizador configura um workflow simples decide se deseja utilizar uma ou ambas as opções, atribuindo nomes apropriados aos passos (e.g., "aprovar" ou "rejeitar"). Quando o passo é invocado, o conteúdo pode ser copiado ou movido para outra pasta. Assim, em workflows simples, o estado de um documento é determinado pela pasta em que está situado, podendo existir pastas denominadas, e.g., "Aprovados" ou "Em Revisão". A desvantagem da implementação destes workflows reside na impossibilidade de acrescentar detalhes, que podem ser necessários com o aumento da complexidade dos processos de

negócio [Pot10].

Ao contrário dos workflows simples, que são configurados pelo utilizador e limitados a processos em série, os workflows avançados potencializam processos paralelos e fornecem a capacidade de adicionar lógica através de código Javascript ou Java. A implementação de workflows avançados no Alfresco apresenta as seguintes características:

- Integração do motor de workflows JBoss jBPM para a execução de workflows;
- Definições de processos de negócio usando um editor de texto ou o JBoss Graphical Process Designer (GPD);
- Acrescentar lógica utilizando Javascript e Java;
- Adequados a processos de longa duração e podem incluir etapas assíncronas desencadeadas por programas externos;
- Temporizadores podem ser acrescentados utilizando datas absolutas ou relativas;

Existem diversas características e ferramentas que a tecnologia JBoss jBPM disponibiliza por forma a suportar processos de negócio em todo o seu ciclo de vida:

- Editor baseado na ferramenta Eclipse e na *web* para suporte ao desenvolvimento gráfico dos processo de negócio, utilizando as funcionalidades de *drag and drop*;
- Funcionalidades de persistência e transacções baseadas em Java Persistence API (JPA) e Java Transaction API (JTA);
- Serviço de tarefas pessoais WS-HumanTask, para incluir tarefas que necessitam ser realizadas por pessoas;
- Gestão da consola suportando gestão de instâncias de processos, gestão de listas, formulários de tarefas, e relatórios;
- Repositório optimizado de processos para publicação (e outro conhecimento relacionado);
- Histórico para monitorização e análise;
- Integração com Seam, Spring, OSGi, etc [Com11b];

A edição baseada na *web* é possível utilizando o Oryx Designer. Este designer está integrado no Guvnor, um conhecido repositório onde se pode guardar os processos modelados. Este designer pode ser utilizado para ver, modificar ou criar processos utilizando a notação BPMN 2.0 [Com11a].

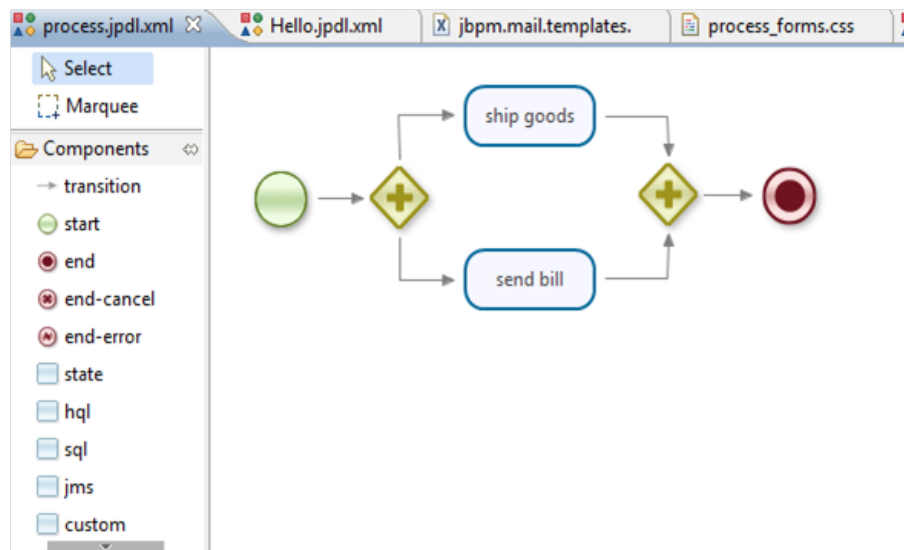


Figura 2.9: Visualização gráfica de workflow utilizando o JBoss jBPM

O JBoss Graphical Process Designer (GPD) é um *plugin* da plataforma Eclipse, que permite modelar graficamente workflows avançados, configurando todos os detalhes necessários para a execução no Alfresco. A figura 2.9 ilustra o ambiente gráfico de modelação do GPD integrado com a plataforma Eclipse [Col11b].

2.3.2 Alfresco - Activiti

Uma ferramenta recentemente criada para a modelação e configuração de workflows para posteriormente serem executados no Alfresco, é a Activiti. Esta ferramenta é um motor de workflows dirigido para programadores Java, e foi criada com o objectivo de se tornar um substituto do JBoss jBPM.

A primeira versão do Activiti contém um conjunto de ferramentas necessárias para o desenvolvimento de um projecto relacionado com workflows, inclui:

- **Activiti Engine** - um simples ficheiro JAR contendo a Processo Virtual Machine e a linguagem de implementação de processos BPMN;
- **Activiti Probe** - uma consola de administração do sistema para controlar e operar o motor Activiti;
- **Activiti Explorer** - uma simples aplicação *end-user* para a gestão de tarefas e a respectiva execução dos workflows;
- **Activiti Modeler** - uma ferramenta de modelação BPMN 2.0 baseada em *web* e Ajax, especialmente desenhada para analistas de negócios [Alf11].

O Activiti Modeler é uma versão personalizada do editor de processos *open-source* e baseados em *web* Signavio. Pode ser utilizado para construir graficamente processos segundo a notação BPMN 2.0. Os ficheiros que são obtido na modelação dos workflows são guardados pelo servidor num sistema de ficheiros para serem mais facilmente acessível.

Existe também uma versão *plugin* do Eclipse, o Activiti Eclipse Designer, que pode ser utilizado para modelar graficamente o workflow pretendido pelo utilizador. Normalmente utiliza-se mais o Activiti Modeler para modelação de alto-nível. O Activiti Eclipse Designer pode ser posteriormente utilizado para acrescentar detalhes técnicos necessários, como tarefas de serviços Java e *execution listeners*. [Act11].

2.3.3 Microsoft Sharepoint 2010

O Microsoft SharePoint 2010 é uma plataforma de colaboração dirigida a empresas, que disponibiliza aos utilizadores um conjunto integrado de recursos avançados, respondendo às necessidades comerciais da empresa.

Para além de possuir workflows pré-definidos para atender diversas necessidades processuais, o Microsoft Sharepoint 2010, dispõe de uma funcionalidade adicional para o desenvolvimento e implementação de workflows de acordo com as necessidades da organização, podendo ser simples ou complexos dependendo da exigência do processo de negócio.

O Microsoft Sharepoint 2010 inclui vários modelos de workflows embutidos (e.g. documento para aprovação), que por vezes são inadequados, necessitando da respectiva reformulação. A implementação de novos workflows, para além dos pré-definidos, pode ser obtida de 3 maneiras distintas:

- Utilização do Microsoft Sharepoint Designer;
- Utilização do Microsoft Office Visio com posterior exportação para o Microsoft Sharepoint Designer;
- Utilização do Visual Studio através de código de programação;

Para uma implementação de workflows sem a utilização de código de programação, utilizando apenas uma lista pré-definida de actividades, será aconselhada a utilização do Microsoft Office Visio com a posterior exportação para o Microsoft Sharepoint Designer. A figura 2.10 ilustra o ambiente gráfico do Microsoft Office Visio.

O Microsoft Office Visio é uma aplicação que permite criar diagramas para ambientes Windows. Pode ser utilizado para criar diagramas de vários tipos, como organogramas, fluxogramas, modelos de dados (utilizando UML ou outra notação gráfica) ou diagramas de redes. Esta aplicação foi desenvolvida pela Visio Corporation, uma empresa independente que foi adquirida em 2000 pela Microsoft. Desde então, o Visio foi incorporado no

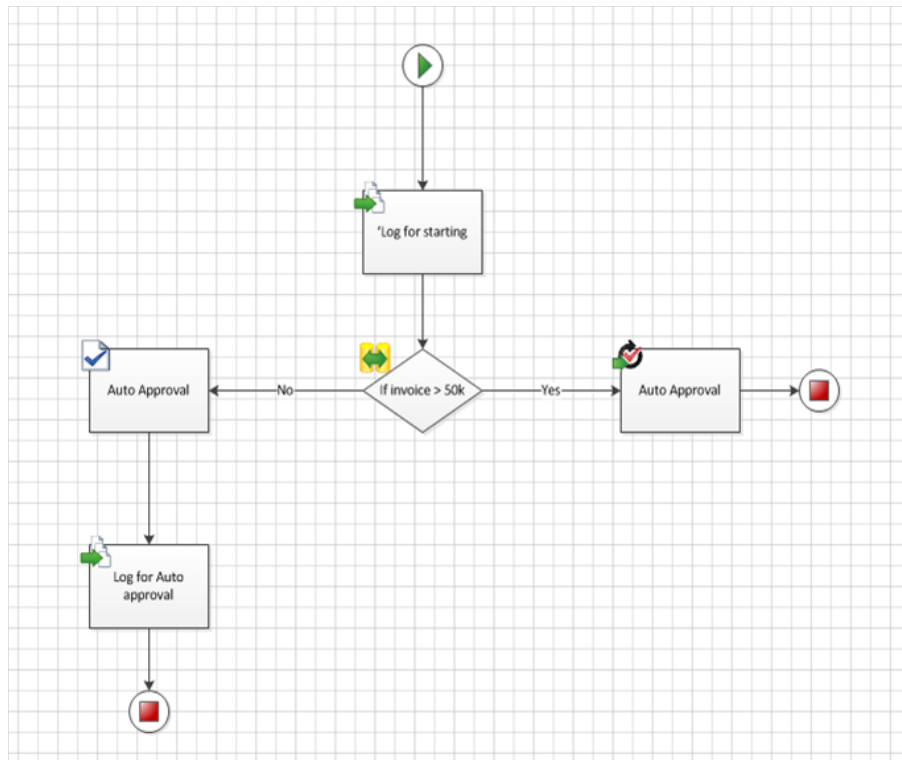


Figura 2.10: Visualização gráfica da modelação no Microsoft Office Visio

pacote de ferramentas do Microsoft Office. As ferramentas avançadas de criação de diagramas do Visio 2010 ajudam a simplificar a sua complexidade, utilizando efeitos visuais dinâmicos [Off11].

O Microsoft Office Visio possibilita a modelação de workflows, através da capacidade de exportação e importação de workflows com o Microsoft SharePoint Designer 2010.

O Visual Studio 2010 poderá ser utilizado em situações cujo código é a forma mais simples de implementar um determinado workflow [20111].

2.3.4 TIBCO Business Studio

O TIBCO Business Studio é um ambiente de modelação de processos de negócio *open-source* baseado na notação BPMN, que permite a analistas de negócios modelarem e simularem processos baseando-se no suporte de dados e modelos da organização. O TIBCO Business Studio disponibiliza dois modos de modelação, *Business Analyst* e o *Solution Designer*.

O *Business Analyst* permite a modelação de processos, dispondo do acesso a vários artefactos, e utilizando apenas um ficheiro adequado a uma modelação abstracta e de alto-nível.

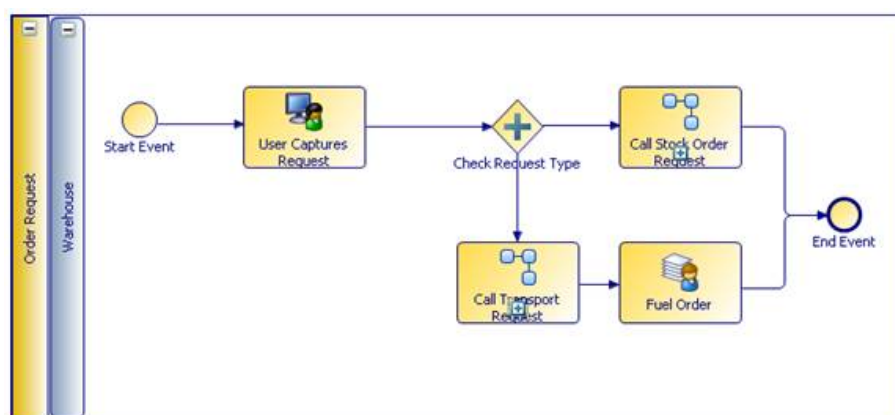


Figura 2.11: Exemplo de Diagrama BPMN utilizado no TIBCO Business Studio

O *Solution Designer* fornece uma visão baseada no projecto que permite ao designer adicionar detalhes de gestão de processos. Este modo é indicado para utilizadores mais familiarizados com ferramentas de desenvolvimento e que necessitam do acesso a todos os arquivos e ficheiros relacionados com o projecto.

Na modelação gráfica são utilizados dois tipos de processos:

- **Processos de Negócio** - correspondem a processos de negócio actuais ou futuros da organização, geralmente envolvendo mais do que uma pessoa, sendo de curta ou longa duração;
- **Processos de Fluxo de Página** - correspondem a processos de curta duração que são projectados uma implementação da interface com o utilizador, sendo apenas executados por uma pessoa;

O editor dos processos disponível na aplicação dispõe de vasto um conjunto de ícones e diagramas BPMN. O TIBCO apresenta uma agradável visualização gráfica, que permite ao utilizador construir o seu processo adaptando-se facilmente a diversos conceitos, como pacotes, processos, eventos ou tarefas. A utilização de BPMN é uma mais-valia na modelação, por permitir a utilizadores familiarizados com essa notação, uma intuitiva modelação gráfica [BPM11]. Na figura 2.11 está representado um exemplo de um diagrama BPMN utilizando o TIBCO Business Studio.

2.3.5 IBM Websphere Business Modeler

A aplicação IBM Websphere Business Modeler permite a analistas de negócios documentar, projectar, analisar e otimizar processos de negócios, com o objectivo oferecer a base para soluções de integração de processos.

Além de permitir a concepção, modelação e implementação de processos de negócios vitais à organização, permite ainda:

- Tomar decisões informadas antes da distribuição, através de recursos avançados de simulação baseada em dados reais e modelados;
- Fornece um conteúdo relacionado com a área industrial para ajudar os utilizadores no desenvolvimento da solução;
- Acelera o processo de optimização, permitindo aos utilizadores visualizar e identificar deficiências em processos;
- Permite a especialistas no assunto partilhar modelos de processos de negócio e colaborar na sua produção, usando o navegador *web* denominado por WebSphere Business Compass [IBM11].

Uma das funcionalidades mais interessantes consiste na importação de modelações desenvolvidas noutras ferramentas e em diversos formatos (e.g. modelos Microsoft Visio, folha de cálculos Microsoft Excel). Assim, um utilizador poderá utilizar outros modelos que possui, como ponto de partida na modelação do novo workflow.

O IBM WebSphere Business Modeler disponibiliza um vasto conjunto de modos de modelação de processos, que permitem o seu desenvolvimento em diferentes níveis de detalhe. Quatro dos modos de modelação facilitam a exportação de modelos para diferentes tecnologias. A tabela 2.2 apresenta esses modos de modelação.

O modelo gráfico de um workflow padrão utilizado pelo WebSphere é baseado num *layout* de forma livre, ou seja, não impõe restrições relativamente à localização dos elementos no diagrama. Outra alternativa é o *swimlane layout*, que identifica facilmente as trocas desnecessárias entre elementos, resolvendo alguns problemas relacionados com redundância.

O Websphere também permite a criação de diagramas de processos que partilham a mesma convenção e simbologia padronizada pelo Business Process Modeling Notation (BPMN) [IBM11]. A figura 2.12 ilustra o estilo BPMN no Websphere.

2.3.6 BizAgi

A ferramenta *freeware* BizAgi Process Modeler permite modelar, documentar e publicar processos de negócio. Na criação dos modelos de workflow, o Process Modeler da BizAgi suporta integralmente a notação BPMN. Esta ferramenta permite exportar os dados da modelação para outros tipos de formato, tais como: PNG, PDF, Microsoft Visio, Word e XPD. A partir da versão 1.5.1 é também possível fazer a publicação do modelo na Web, exportar para ferramentas Wiki ou exportar para o Microsoft Sharepoint.

A empresa BizAgi disponibilizou também uma ferramenta para executar os processos que foram modelados no Process Modeler, o BPM Suite. Para o motor de workflows do BizAgi funcionar completamente são necessários os seguintes módulos:

Tabela 2.2: Funções dos vários modos de modelação do IBM Websphere Business Modeler [IBM11]

Modo de Modelação	Função
Básico	Para o desenvolvimento de modelos de processos de negócio de alto-nível, utilizando a criação e exibição de fluxos sequenciais.
Avançado	Para especificar e desenvolver detalhes adicionais, como regras e lógica de negócio.
WebSphere Business Integration Server Foundation	Para exportação de modelos de processos em Business Process Execution Language (BPEL), Web Services Description Language (WSDL) e XML Schema Definition (XSD).
WebSphere MQ Workflow	Para a geração de <i>output</i> em FlowMark Definition Language (FDL).
WebSphere Process Server	Para a exportação de modelos Service Component Architecture (SCA) e ficheiros BPEL, WSDL, e XSD.
FileNet Business Process Manager	Para a exportação de modelos de processos XML Process Definition Language (XPDL).
WebSphere Business Services Fabric	Para a exportação de modelos de processos Service Component Architecture (SCA), e formatos como BPEL, WSDL, XSD, e arquivos CAD.

- **BizAgi Process Modeler** - Módulo que permite a construção de diagramas BPMN e associação de documentação;
- **BizAgi Studio** - Módulo de construção de informação;
- **BizAgi BPM Server** - Módulo de execução e controlo.

O conceito base do BizAgi é a geração automática de uma aplicação *web*, através da modelação, configuração e publicação de um diagrama de processo de negócio sem quaisquer requisitos de programação. Para alcançar esse objectivo, o BizAgi traduz um ciclo de vida completo de um processo de negócio, através das etapas: Model, Automate, Execute, e Improve. Cada uma dessas fases é gerida por diferentes componentes, que permitem, através de um ambiente gráfico e dinâmico, a implementação de uma solução baseada em processos.

O primeiro passo para a criação de uma solução baseada em BizAgi é a identificação dos processos. Esta funcionalidade é adquirida com o módulo Process Modeler, e permite a construção de diagramas e a documentação dos processo de negócio, apresentando-os graficamente através da notação bastante conhecida BPMN.

Com o BizAgi, a fase de desenho e documentação do processo corresponde à utilização do Process Modeler, depois é utilizado o BizAgi Studio para inserir toda a informação

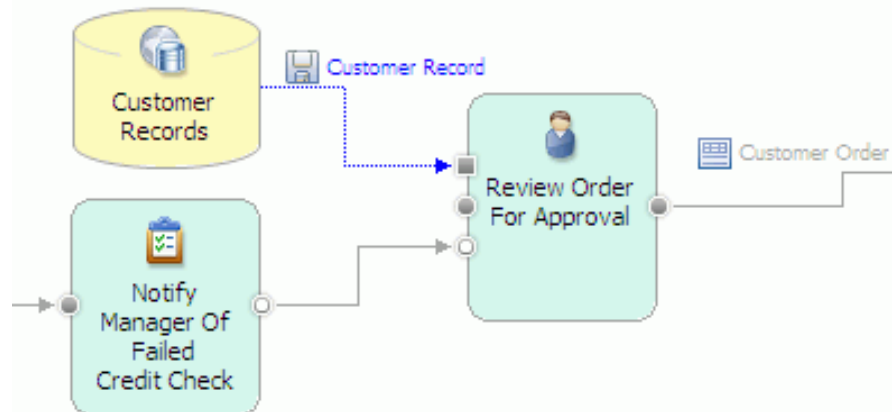


Figura 2.12: Modelo de workflow utilizando o estilo BPMN no IBM Websphere Business Modeler [IBM11]

necessária à execução do processo (e.g. tempo de execução, custos, interfaces com o utilizador, regras de negócios). Toda a informação do modelo é colocada numa base de dados e posteriormente utilizada em tempo de execução pelo BizAgi BPM Server. O BizAgi BPM Server executa directamente BPMN e disponibiliza um portal para a interacção com utilizadores [Biz11].

2.3.7 Windows Workflow 4.0

O Windows Workflow 4.0 (muitas vezes denominado por WF4) é uma tecnologia Microsoft incluída na plataforma .NET 4. Apresenta um novo paradigma na construção de aplicações baseadas em workflows. O WF4 contém uma enorme biblioteca de actividades pré-definidas que permitem a construção de outras actividades/workflows mais elaboradas, através da plataforma Workflow Designer incluída no Visual Studio 2010 [Col10].

Tipicamente no WF4, os utilizadores necessitam de seleccionar um estilo de workflow. Esses estilos, que diferem principalmente na definição do controlo do fluxo, são constituídos pelas actividades *Sequence* e pelas actividades *Flowchart*.

A figura 2.13 ilustra o exemplo de um modelo de workflow sequencial (*Sequence*). As actividades *Sequence* correspondem a um estilo em que as actividades constituintes executam uma após a outra, ou seja, não é permitido retroceder para actividades anteriores, o que exige uma execução direccionada sempre para a próxima actividade.

Os *Flowchart* são alternativas às actividades *Sequence*, na medida em que têm suporte a *loops* (ou seja, permitem retroceder para actividade anterior), utilizando para esse efeito nós de decisão (*FlowDecision*) ou nós de comutação (*FlowSwitch*). O fluxo de controlo é definido através de um grafo direccionado, criado para ligar as actividades entre si. A

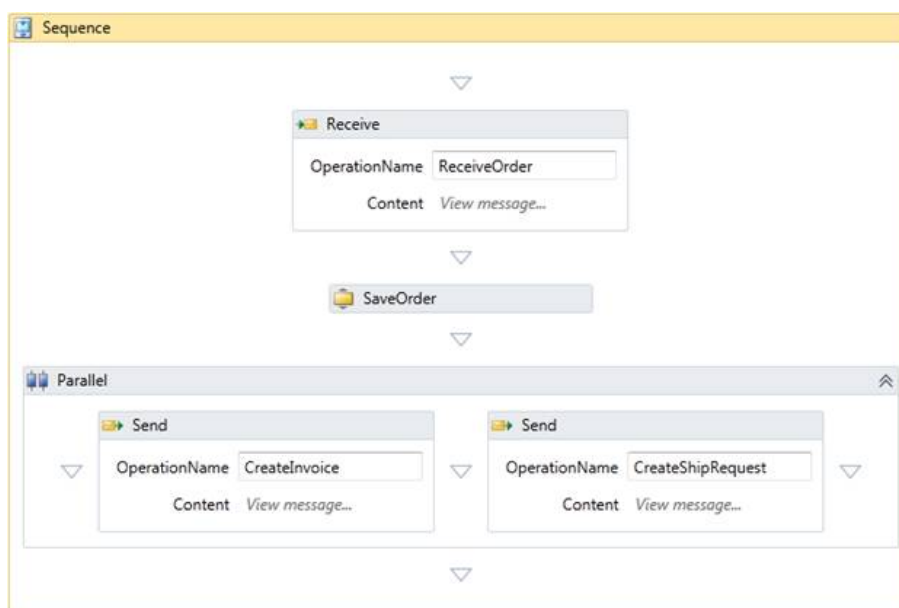


Figura 2.13: Modelo Sequencial do Windows Workflow 4.0 [Mil11]

figura 2.14 representa um modelo *Flowchart* do WF4, evidenciando as diferenças entre modelo sequencial da figura 2.13.

A WF incluiu nas suas anteriores versões um outro tipo de actividades, designadas por *State Machines*, que permitia ao utilizador criar explicitamente diagramas baseados em máquina de estados. Porém, na versão 4 da plataforma .NET esse estilo não foi implementado, devido à necessidade de uma adaptação à nova definição de actividade. Recorrendo aos nós de decisão e comutação, as actividades *Flowchart* permitem ainda assim obter variadas situações que anteriormente eram conseguidas com as *State Machines* [Cha07].

Para a modelação de workflows, além de poderem ser utilizadas actividades personalizadas, poderá ser necessário o recurso às diversas actividades pré-definidas pela plataforma .NET 4:

- **Assign** - atribui um valor a uma variável;
- **DoWhile** - executa uma actividade e verifica uma condição. A actividade deixará de executada se a condição verificar-se verdadeira;
- **ForEach** - executa uma actividade por cada objecto de uma colecção;
- **If** - cria um salto numa execução se uma determinada condição verificar-se;
- **Parallel** - executa múltiplas actividades em paralelo;
- **Persist** - lança um pedido explícito ao WF Runtime para persistir o workflow;
- **Pick** - permite esperar por um conjunto de eventos;

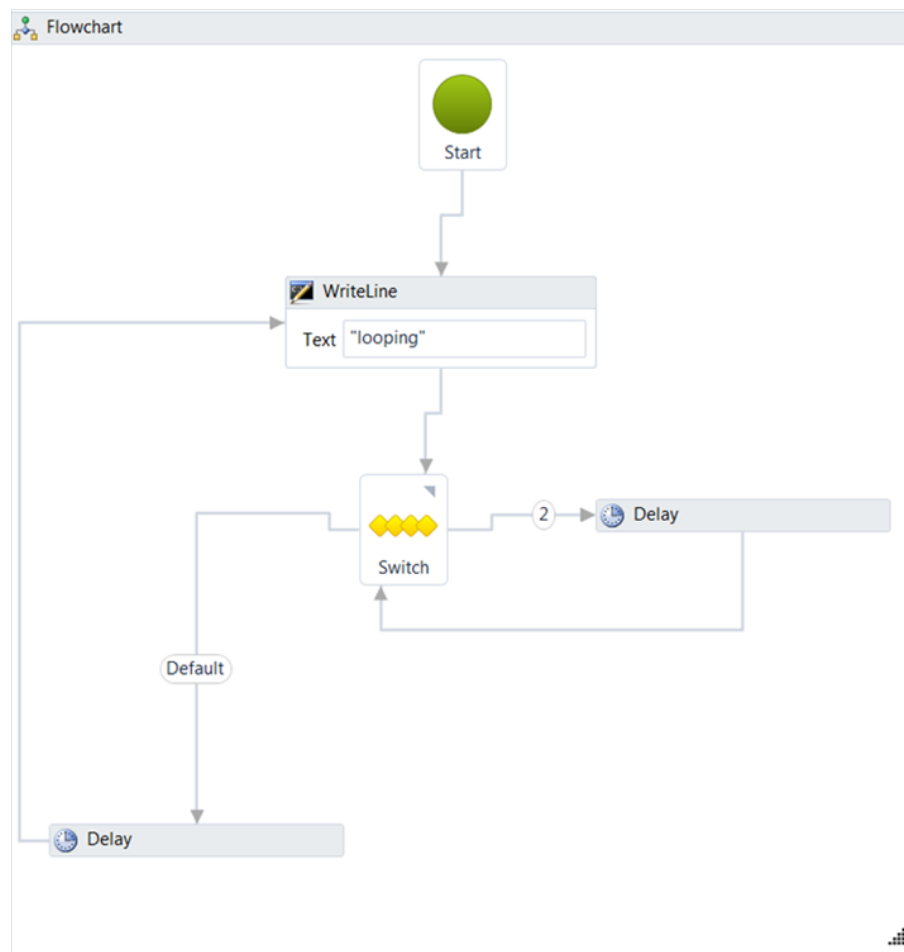


Figura 2.14: Modelo Flowchart do Windows Workflow 4.0 [Mil11]

- **ReceiveMessage** - recebe uma mensagem através de Windows Communication Foundation (WCF);
- **SendMessage** - envia uma mensagem através de WCF;
- **Switch** - disponibiliza uma forma de efectuar vários saltos condicionais utilizando uma variável;
- **Throw** - lança uma excepção;
- **Try/Catch** - permite criar um bloco try/catch para o tratamento de excepções;
- **While** - executa uma determinada actividade enquanto uma certa condição for verdadeira [Cha07].

Uma das características mais vantajosas no WF4 é a capacidade de *re-hosting*. Esta característica permite a utilização do WF4 fora do ambiente do Visual Studio através de

um vasto conjunto de API's, fornecendo ao utilizador a oportunidade de implementar um ambiente próprio e especializado para a modelação de workflows.

Além da capacidade de *re-hosting*, há várias vantagens que o WF4 fornece, tais como:

- Abstracção de alto-nível e representação visual para os processos de negócio das organizações, o que torna fácil de perceber e de projectar a utilizadores peritos no domínio de negócio ou a programadores;
- Facilidade de mudar o fluxo e as regras associadas a processos de negócio, muitas das vezes sem a necessidade de recompilar;
- Possibilidade de criar actividades que podem ser reutilizadas no contexto de outros workflows;
- Execução robusta de workflows, através do Windows Workflow Runtime (WWR).

O WF4 contém um ambiente gráfico (*Workflow Designer*) que é executado no Visual Studio 2010. Esse ambiente gráfico é interpretado através de código eXtensible Application Markup Language (XAML) associado a cada workflow, que inclui a definição do seu estado e do controlo de fluxo [Cha07].

Em relação a anteriores versões do WWF, existem várias melhorias que foram acrescentadas no WF4. Além de possuir uma cuidadosa melhoria no ambiente gráfico, foi implementado com um largo aumento de desempenho e escalabilidade em relação à versão anterior.

As vantagens em relação à anterior versão 3.0 são:

- **WF Runtime** - O WF Runtime é um *scheduler* assíncrono que conduz a execução das actividades num workflow. Fornece um ambiente de execução de actividades com largo desempenho. O ambiente possui contractos bem-definidos para execução, continuação, finalização, cancelamento e excepções. Em comparação com o WF3, o WF4 Runtime possui um *scheduler* mais eficiente. Aproveita o mesmo *I/O thread pool* que é usado para WCF. A fila interna de items de trabalho do *scheduler* é optimizado para o uso de padrões mais comuns. O WF4 Runtime também gere os estados de execução de uma maneira simples e com o mínimo de sincronização e lógica de rotinas de eventos, o que a sua anterior versão depende do registo de eventos e invocação para realizar sincronizações complexas para transições de estados;
- **Armazenamento e Fluxo de Dados** - No WF3, os dados associados a uma actividade são modelados através de propriedades de dependência implementados pelo tipo *DependencyProperty*. Este padrão de propriedades de dependência é bastante flexível na medida em que permite o suporte a *Data Binding* e outras suportes de UI. Mas por outro lado, o padrão requer que as propriedades sejam definidas como

campos estáticos na definição do workflow. Sempre que o WF Runtime faz um *get* ou *set* aos valores da propriedade, envolve uma forte busca de lógica, o que diminuiu bastante o desempenho do sistema. Já a nova versão WF4 utiliza propriedades simples para melhorar o acesso e tratamento de dados. Permite separar os dados armazenados numa actividade dos dados que estão fluindo transversalmente às fronteiras da actividade. Para esta finalidade são utilizados dois conceitos: variáveis e argumentos. Por utilizar um claro alcance hierárquico para variáveis e argumentos de *input* e *output*, a complexidade de uso de dados para as actividades é drasticamente reduzida e o tempo de vida dos dados também é automaticamente bem-definida. Por inspecionar simplesmente uma actividade, podem-se determinar quais os dados que são esperados receber e quais os dados serão produzidos como resultado final da execução da respectiva actividade. Outra diferença que existe nas duas versões é que as actividades da versão WF3 são inicializadas quando o workflow é criado, enquanto que na versão WF4 as actividades são inicializadas apenas quando são executadas. Esta funcionalidade permite um simples ciclo de vida da actividade sem desempenhar operações de inicialização e encerramento, o que cria um aumento de eficiência;

- **Fluxo de Controlo** - Quando uma actividade necessita ser re-executada, na versão WF3 era criada uma nova classe *ActivityExecutionContext* e a actividade era clonada, o que tornava o desempenho do controlo de fluxo iterativos demasiadamente lentos do que a execução sequencial das mesmas actividades. Em WF4 este tratamento funciona de maneira diferente, porque cada actividade é adicionada a fila do *scheduler* e executada sempre que for iterada;
- **Programação Assíncrona** - WF4 permite o suporte a programação assíncrona através actividades derivarem da classe *AsyncCodeActivity*. O ambiente de execução percebe as actividades assíncronas e coloca automaticamente numa zona de não-persistência. As actividades personalizadas podem derivar destes tipos e aumentar o desempenho de trabalho assíncrono, dado que não é necessário agarrar um *thread* do *scheduler* e bloquear outras actividades que poderão também estar prontas a correr em paralelo;
- **Envio/Recepção de Mensagens** - a versão WF3 tinha várias limitações relativamente ao suporte de troca de mensagens de eventos externos ou invocações a web services. Na plataforma .NET 3.5, os workflows podiam ser implementados como clientes ou serviços WCF, utilizando as actividades *SendActivity* e *ReceiveActivity*. Em WF4, o conceito de programação através dea troca de mensagens foi reforçada como uma integração total da lógica de mensagens WCF em WF;

- **Programação Declarativa** - WF4 permite uma simples e fácil programação declarativa para modelar processos de negócio e serviços, podendo fazer composição de actividades apenas declarativamente. Na framework .NET 4, a programação declarativa baseada em XAML foi unificada numa única *assembly* (System.Xaml.dll) para suportar WPF e WF. Este estilo de programação tornava-se difícil de fazer com o formato XOML sem envolver lógica por *code-behind*.
- **Workflow Designer** - O suporte a programação inteiramente declarativa impõe altos requisitos para um desempenho de design para a modelação de grandes workflows. O Workflow Designer para grandes workflows oferece maior escalabilidade do que o WF3. Este efeito é conseguido através do suporte a virtualização da UI, o que proporciona um carregamento de um grande workflow de 1000 actividades em apenas alguns segundos, o que era quase impossível para o WF3 carregar um workflow com algumas centenas de actividades [MD11]

Com as melhorias de desempenho anteriormente referidas, consegue-se constatar que existem diversas vantagens do WF4 em relação ao WF3. Apesar da nova versão não envolver a definição de *State Machines*, existem várias vantagens relacionadas com o desempenho.

Uma outra vantagem do WF4, é a possibilidade de construir *Custom Activities*, ou seja, actividades que possuem um funcionamento próprio. O WF4 possui uma rica biblioteca de actividades para a interacção com serviços, objectos e colecções, mas não disponibiliza actividades para interagirem com subsistemas como base de dados, servidores de email, ou objectos e sistemas do domínio do programador. Quando um workflow está a ser desenvolvido pode ser composto por actividades mais pequenas fornecidas pela plataforma.

Apesar de derivarem directamente da classe *Activity*, existe uma hierarquia de classes de actividades, que podem ser escolhidas para construir a *Custom Activities*. São elas:

- **Activity** - utilizada para modelar actividades normalmente definidas por XAML;
- **CodeActivity** - utilizada quando é necessário escrever código para efectuar a sua execução;
- **AsyncCodeActivity** - utilizada quando a actividade executa o trabalho assíncronamente;
- **NativeActivity** - utilizada quando é necessária a criação de *bookmarks* na actividade [Mil11].

2.4 Conclusão

Numa primeira parte deste capítulo foram apresentados conceitos teóricos relacionados com a modelação de processos de negócio.

O segundo sub-capítulo apresenta algumas notações de modelação de processos de negócios, nomeadamente: *flowchart*, diagramas de actividade, e BPMN. Existem poucas diferenças entre essas três notações. A grande diferença reside no modo de aplicação e no contexto em que são utilizadas. Consegue-se constatar que as notações BPMN e diagramas de actividade são uma evolução da notação *flowchart*, sendo esta a notação mais simplificada (contendo apenas quatro tipos de símbolos: Terminador, Processo, Decisão e Conector) para permitir a percepção de vários *stakeholders*.

Relativamente aos diagramas de actividade UML e a BPMN, os autores Eloranta, Kallio e Terh afirmam que as diferenças do poder de representação destas notações são ligeiramente diminutas. Nos seus resultados obtidos numa análise à diferença entre diagramas de actividade UML e BPMN, através de estrutura de padrões de workflows, conclui-se que, é necessário proceder a uma escolha da notação a ser utilizada, não baseada na forma de representação, mas sim na experiência já existente com a notação [LE06]. Um estudo comparativo sugere até, que o nível de dificuldade para a compreensão do processo de negócio, em ambas as notações, é o mesmo [DP06].

No terceiro sub-capítulo foram apresentadas as principais características das tecnologias que permitem a modelação e configuração de processos de negócio. Através de uma análise empírica e comparativa das tecnologias evidenciadas neste sub-capítulo, foi possível construir um quadro-resumo, representado na figura 2.15, das principais características necessárias num ambiente de modelação de processos de negócio. Na análise empírica foram avaliadas as seguintes características:

- **Passagem de parâmetros** - facilidade na configuração da troca de dados e da correcta definição de parâmetros entre actividades;
- **Configuração de workflows** - facilidade na configuração de propriedades relacionadas com a modelação do workflow;
- **Reutilização de workflows** - possibilidade do modelo do workflow ser reutilizado no contexto de outro modelo de workflow, e o modo como é alcançado esse mesmo mecanismo;
- **Apresentação gráfica de actividades** - visualização e modificação aspecto gráfico das actividades;
- **Usabilidade** - facilidade em alterar o modelo gráfico do workflow através da inserção e disposição das actividades no modelo e das alteração das respectivas conexões;

Revisão Bibliográfica

	JBoss jPBM	Activiti	Microsoft Sharepoint Designer 2010	TIBCO Business Studio	IBM WebSphere Business Modeler	BizAgi Process Modeler	Windows Workflow 4.0
Passagem de Parâmetros	●	●	●	●	●	●	●
Configuração de workflows	●	●	●	●	●	●	●
Reutilização de workflows	●	●	●	●	●	●	●
Apresentação gráfica de actividades	●	●	●	●	●	●	●
Usabilidade	●	●	●	●	●	●	●
Re-Hosting	●	●	●	●	●	●	●

● Mau ● Razoável ● Bom

Figura 2.15: Quadro-Resumo das características das tecnologias

- **Re-hosting** - possibilidade em transferir o ambiente gráfico e as respectivas funcionalidades para uma outra aplicação;

Neste quadro-resumo é possível concluir que, numa primeira impressão com a ferramenta, não é totalmente intuitivo o modo de definição da passagem de parâmetros entre actividades e da respectiva configuração de propriedades do workflow e das actividades. Esta característica limitativa evidencia-se com a utilização de extensos formulários e com o impacto da utilização de diversos conceitos que não estão totalmente uniformizados entre as várias tecnologias. Também é possível concluir que o modo de reutilização de workflows nas tecnologias relacionadas com o Alfresco, não são totalmente clarividentes em relação às outras tecnologias. Ao nível da apresentação gráfica existe uma clara divergência entre as respectivas tecnologias, realçando positivamente as apresentações gráficas das tecnologias TIBCO Business Studio e da IBM WebSphere Business Modeler. Na usabilidade da modelação gráfica existe claramente a possibilidade de um melhoramento, evidenciando-se positivamente a BizAgi Process Modeler, por permitir uma modelação mais rápida e eficaz.

Uma característica comum em quase todas as tecnologias é a impossibilidade de transferir o ambiente gráfico e respectivas funcionalidades para uma outra aplicação (*re-hosting*), o que realça uma característica bastante limitativa em aplicações deste género. A tecnologia WF4 é a única, entre as referidas, que permite o *re-hosting*, criando um ambiente gráfico de modelação capaz de ser modificado de acordo com as necessidades funcionais. Assim, esta característica da tecnologia WF4 tornou-se um factor decisivo

Revisão Bibliográfica

para a selecção de uma plataforma base na implementação do ambiente de modelação e configuração de processos.

Capítulo 3

Especificação

3.1 Definição do Problema

A solução da Siemens Sector Healthcare tem como objectivo fundamental o suporte das tarefas administrativas que sustentam o fluxo de processos administrativo/financeiros associados a um paciente. Cada vez mais os sistemas necessitam de uma separação de tarefas das equipas multidisciplinares, fornecendo um aumento da flexibilidade dos processos organizacionais. A arquitectura da solução da Siemens Sector Healthcare reflecte uma visão modular que possibilita o nivelamento de acordo com as competências e a organização das equipas envolvidas no projecto (*Designer: "thinks about the look and behavior"; Developer: "thinks about code and functionality"; Consultant: "thinks about the workflow"*).

Existe a necessidade de adaptar a essa solução um componente de gestão de workflows, que permitirá modelar processos de negócio e configurar determinados parâmetros, para uma serem executados convenientemente num ambiente de execução, também integrado na solução global. Este componente permitirá uma orientação para trabalho colaborativo, ou seja, permitirá a separação de tarefas entre utilizadores. A separação de tarefas e competências é determinante para a organização para uma concentração de cada membro da equipa no âmbito do seu trabalho, acrescentando benefícios de tempo. Neste sentido, um componente de gestão de workflows integrado na solução global, constituirá uma mais valia para para a organização, permitindo a separação das tarefas de modelação e execução do processo de negócio.

Presentemente, e como foi referido no capítulo 2, existem diversas aplicações que permitem modelar processos de negócio utilizando várias notações e formas de configuração, mas que fornecem um conjunto de funcionalidades que, devido à sua complexidade ou âmbito, não são adequadas à solução que necessita de ser implementada.

Um problema evidenciado na maior parte das aplicações existentes é o facto de não permitirem o seu *re-hosting*. Esta funcionalidade corresponde a uma diminuição do es-

forço e tempo de desenvolvimento do componente de gestão de workflows. O WF4 permite o *re-hosting*, no entanto, as funcionalidades desta tecnologia estão demasiado orientadas à implementação de workflows de baixo nível.

Para a execução dos passos do modelo do processo de negócio, o ambiente de execução da solução global utiliza um mecanismo de *wizard*. O *wizard* é um padrão muito utilizado na interface gráfica, que utiliza um esquema passo-a-passo, e permite, de forma simples, realizar tarefas complexas numa aplicação. Nesta perspectiva, é indispensável uma implementação que permite uma navegação do processo de negócio, dependendo da opção do utilizador (e.g. "Anterior", "Seguinte").

Concluindo, existem dois problemas principais no desenvolvimento do ambiente de modelação e configuração de processos: i) o tempo de desenvolvimento e a complexidade, que seria menor se fosse reaproveitado um ambiente de modelação já existente, ii) a necessidade de tornar a modelação dos processos de negócio mais intuitiva, dirigida aos intervenientes *Developer* e *Consultant*, e adequada ao âmbito da solução global.

3.2 Visão de Alto-Nível

Do ponto de vista da gestão de workflows, a solução global da Siemens Sector Healthcare apresenta-se dividida em duas áreas:

- **Configurador** - área onde está disponível o componente da modelação e configuração dos workflows;
- **Execução** - área onde reside a gestão da execução e persistência dos workflows;

A arquitectura do Configurador é baseada numa arquitectura modular representada na figura 3.1. O componente de gestão de workflows é um componente *loose coupled* e independentemente evolutivo, que comunica com a *Shell* através de uma camada de serviços da aplicação. A *Shell* é uma componente responsável por agrupar todos os módulos numa visualização gráfica dirigida ao utilizador.

O componente de gestão de workflows utiliza o padrão de desenho Model-View-ViewModel (MVVM). A sua arquitectura está representada na figura 3.2. A vantagem da utilização do padrão MVVM é o facto de permitir à camada de apresentação uma total separação da camada de dados e da lógica de negócio.

A camada *Model* refere-se ao modelo de um objecto ou uma camada de acesso a dados. A camada *View* representa todos os elementos visíveis. A camada do *ViewModel* é uma abstracção de dados da camada *View*, e representa o local onde a *View* obtém toda a informação necessária para apresentar na interface gráfica. A camada *View* acede ao conteúdo da *ViewModel* se necessitar de alguma informação, através de *data binding*.

Especificação

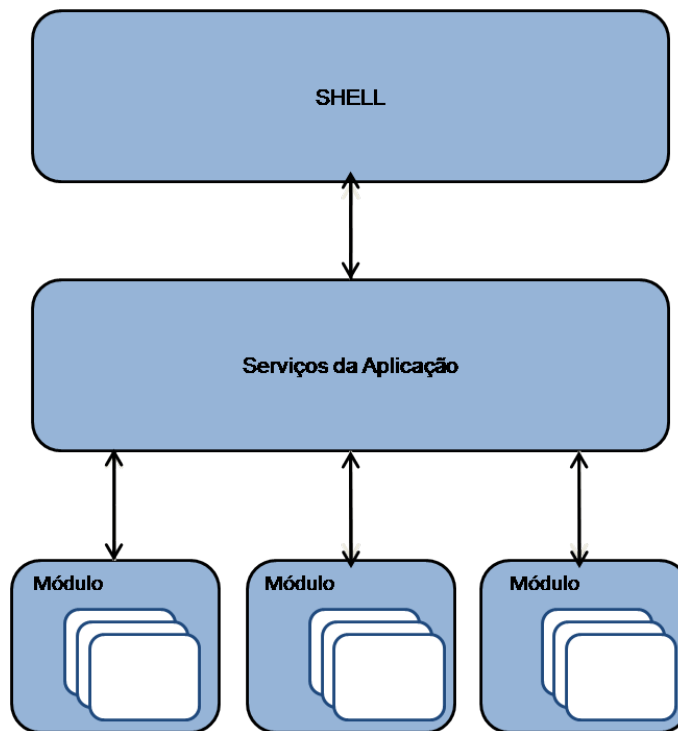


Figura 3.1: Diagrama da arquitectura do Configurador

Os objectos da camada do *Model* e do *ViewModel* implementam a interface *INotifyPropertyChange*, que permitem notificar a camada superior da alteração de alguma propriedade. O *Business Layer* é uma camada que contém serviços que são invocados pelo *ViewModel*, mudando as propriedades da camada *Model*. Esta camada pode conter funções, operadores e o acesso à *Data Layer*, que corresponde à camada responsável pelo acesso à base de dados.

3.3 Requisitos

Nesta secção são descritos os requisitos do componente de gestão de workflows através de *user stories* e diagramas de casos de uso.

Os actores que correspondem aos utilizadores finais do sistema são:

- **Administrador da Siemens** - colaborador da Siemens SA com privilégios máximos de acesso ao configurador;
- **Utilizador da Siemens** - colaborador da Siemens SA com alguns privilégios de acesso ao configurador;
- **Utilizador** - utilizador que corresponde ao cliente final;

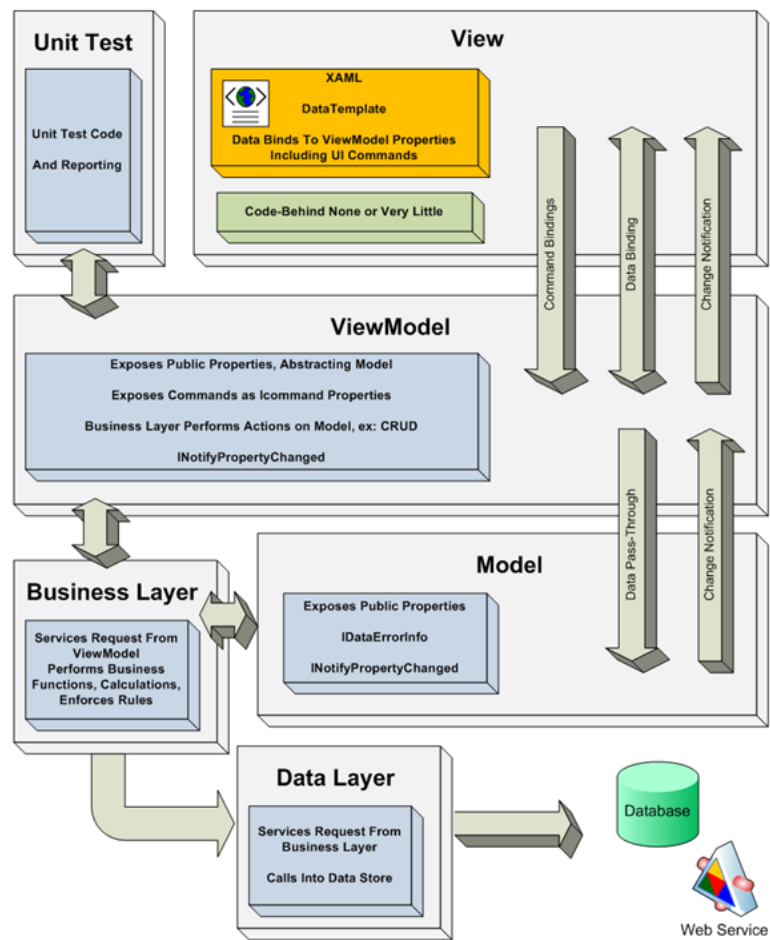


Figura 3.2: Diagrama da arquitetura do componente de gestão de workflows

3.3.1 User Stories

As *user stories* descrevem as funcionalidades desejadas numa solução na perspectiva de um utilizador, contendo informação suficiente para que seja possível produzir uma estimativa do trabalho de implementação. Neste subcapítulo são descritas as *user stories* do componente de gestão de workflows, presentes na tabela 3.1.

Especificação

Tabela 3.1: User Stories

ID	Requisito	User Story
US1	Visualização do Modelo de Processo de Negócio	- Como utilizador, posso visualizar o modelo gráfico de um processo de negócio, baseado em notação <i>Flowchart</i> , e incluindo uma representação gráfica das actividades/workflows no modelo;
US2	Criação do Modelo de Processo de Negócio	- Como utilizador, posso inicializar a modelação de um processo de negócio, preenchendo os seguintes campos: <ul style="list-style-type: none"> • Name - nome do modelo; • Description - descrição do modelo; • Runtime Environment - ambiente de execução do modelo (pode ser <i>UI</i>, para ser visível em ambiente de execução, ou <i>Batch</i>, utilizado como um conjunto de tarefas sequenciais); • Notes - notas informativas agregadas ao modelo;
US3	Actualização do Modelo de Processo de Negócio	- Como utilizador, poderei modificar um modelo de processo de negócio anteriormente criado, acedendo a um painel que contém uma lista de modelos disponíveis para actualização;
US4	Guardar Modelo de Processo de Negócio	- Como utilizador, poderei guardar o estado actual do modelo de processo de negócio;
Continua na próxima página		

Tabela 3.1 – continuação da página anterior

ID	Requisito	User Story
US5	Publicação do Modelo de Processo de Negócio	<p>- Como utilizador, posso publicar um modelo de processo de negócio para este ser executado posteriormente num ambiente de execução, seleccionando a opção de publicação e a categoria em que o modelo será inserido. Essas categorias poderão ser:</p> <ul style="list-style-type: none"> • Framework - correspondem a actividades disponibilizadas pela <i>framework</i> do WF4; • System - correspondem a workflows disponibilizados pelos administradores da Siemens SA, que não podem ser abertos para leitura nem escrita por outros utilizadores; • Base - correspondem a workflows disponibilizados pelos utilizadores da Siemens SA, que não podem ser abertos para leitura nem escrita por utilizadores, mas que podem ser copiados se forem assinalados para tal; • Custom - correspondem a workflows modelados pelos utilizadores, que podem ser abertos para leitura e escrita; <p>- Como administrador da Siemens, na publicação posso categorizar os modelos de processos de negócio em <i>System</i>, <i>Base</i> e <i>Custom</i>;</p>
Continua na próxima página		

Tabela 3.1 – continuação da página anterior

ID	Requisito	User Story
US5	Publicação do Modelo de Processo de Negócio	<p>- Como utilizador da Siemens, na publicação posso categorizar os modelos de processos de negócio em <i>Base</i> e <i>Custom</i>. Se for categorizado como <i>Base</i> é permitida a cópia do modelo a qualquer utilizador;</p> <p>- Como utilizador, na publicação posso categorizar os modelos de processos de negócio como <i>Custom</i>;</p>
US6	Zoom do Modelo de Processo de Negócio	<p>- Como utilizador, posso fazer <i>zoom-in</i> e <i>zoom-out</i>, para obtenção uma melhor visualização do modelo do processo de negócio;</p>
US7	Configuração de Propriedades do Modelo de Processo de Negócio	<p>- Como utilizador, posso ter acesso a um painel de configuração de propriedades do modelo do processo de negócio, para a configuração das seguintes propriedades:</p> <ul style="list-style-type: none"> • Name - nome do modelo; • Description - descrição do modelo; • Runtime Environment - ambiente de execução do modelo (pode ser <i>UI</i>, para ser visível em ambiente de execução, ou <i>Batch</i>, utilizado como um conjunto de tarefas sequenciais); • Notes - notas informativas agregadas ao modelo; <p>- Como utilizador, se atribuir o valor <i>UI</i> à propriedade do modelo de processo de negócio <i>Runtime Environment</i> posso ter acesso a um painel de propriedades de configuração, com impacto no ambiente de execução:</p>

Continua na próxima página

Tabela 3.1 – continuação da página anterior

ID	Requisito	User Story
US7	Configuração de Propriedades do Modelo de Processo de Negócio	<ul style="list-style-type: none"> ● Image - imagem utilizada para identificar o modelo no ambiente de execução; ● Name - nome utilizado para identificar o modelo no ambiente de execução; ● Description - descrição do modelo que será mostrado no ambiente de execução; ● Help - informação de ajuda na execução do modelo; ● Tooltip - descrição utilizada na selecção da execução do modelo; <p>- Como utilizador, posso atribuir uma área de configuração a um modelo de processo de negócio, que acrescenta uma configuração adicional ao workflow no momento da sua reutilização;</p> <p>- Como utilizador, posso criar argumentos de entrada e saída do modelo, através de uma opção disponível numa área própria, incluída no painel de configuração;</p> <p>- Como utilizador, posso alterar os atributos dos argumentos de entrada do modelo, nomeadamente a identificação e tipo de argumento; Nos argumentos de saída poderei atribuir apenas a identificação;</p> <p>- Como utilizador, posso atribuir a cada argumento de saída do modelo, um outro argumento de saída de uma actividade/workflow presente no modelo, ou um argumento de entrada do respectivo workflow, através de uma opção que dá acesso a uma organização com os argumentos disponíveis;</p>
Continua na próxima página		

Tabela 3.1 – continuação da página anterior

ID	Requisito	User Story
US8	Visualização da Toolbox	- Como utilizador, posso visualizar uma caixa de ferramentas (<i>Toolbox</i>) onde serão apresentadas as actividades/workflows que podem ser incluídos no modelo do processo de negócio, organizados nas categorias <i>Framework</i> , <i>System</i> , <i>Custom</i> e <i>Base</i> ;
US9	Adicionar Actividades/Workflows ao Modelo do Processo de Negócio	- Como utilizador, posso adicionar actividades/workflows no modelo do processo de negócio acedendo à caixa de ferramentas;
US10	Ligação entre Actividades/Workflows	- Como utilizador, posso fazer ligações entre actividades/workflows presentes no modelo, para pertencerem ao respectivo fluxo de controlo. Cada actividade/workflow poderá ser ligado apenas a uma outra actividade/workflow.
US11	Retirar Actividades/Workflows do Modelo de Processo de Negócio	- Como utilizador, posso remover uma actividade/workflow do modelo do processo de negócio;
US12	Modificação do Aspecto Gráfico das Actividades/Workflows	- Como utilizador, posso modificar visualmente a apresentação gráfica das actividades/workflows, nomeadamente o ícone e descrição;
US13	Acesso e Configuração das Propriedades dos Workflows	<p>- Como utilizador, posso ter acesso a um painel de configuração de propriedades dos workflows reutilizados, onde posso fazer a leitura das seguintes propriedades:</p> <ul style="list-style-type: none"> • Name - nome do workflow; • Description - descrição do workflow;

Continua na próxima página

Tabela 3.1 – continuação da página anterior

ID	Requisito	User Story
US13	Acesso e Configuração das Propriedades dos Workflows	<ul style="list-style-type: none"> ● Runtime Environment - ambiente de execução do workflow; ● Version - versão publicada do workflow; ● Unique Code - código de identificação do workflow; ● Author Properties - área que contém a informação de autoria do workflow, nomeadamente a data da criação, data da última modificação, e respectivos autores; <p>- Como utilizador, posso ter acesso a uma área de configuração de parâmetros de um workflow, anteriormente associado na modelação, e com impacto nos argumentos de saída do respectivo workflow;</p> <p>- Como utilizador, posso atribuir a cada actividade/workflow no modelo, um componente com impacto na execução do mesmo. Cada componente disponibiliza um conjunto de propriedades, incluindo uma interface gráfica, respectiva validação e a definição de argumentos de saída, que poderão ser atribuídos aos argumentos de entrada da respectiva actividade/workflow;</p> <p>- Como utilizador, posso atribuir a cada argumento de entrada de cada actividade/workflow, um argumento de saída de uma actividade/workflow presente no modelo e anterior no respectivo fluxo de controlo, ou um argumento de entrada do respectivo modelo, através do acesso a uma organização dos argumentos disponíveis;</p>

3.3.2 Diagramas de Casos de Uso

Nesta secção são apresentados os diagramas de casos de uso resultantes das *User Stories* anteriormente referidas. As figuras 3.3 e 3.4 ilustram esses diagramas.

3.3.3 Requisitos Não-Funcionais

Nesta secção são descritos os requisitos não-funcionais, que dizem respeito ao funcionamento geral do componente de gestão de workflows.

- **Usabilidade** - dado que o sistema pode ser utilizado por utilizadores com o mais variado nível de conhecimento, é importante que seja de utilização intuitiva e o seu aspecto agradável;
- **Fiabilidade** - o componente deverá ser capaz de permanecer funcional vinte e quatro horas por dia, sete dias por semana, em condições normais;
- **Modularidade** - o componente deverá ser implementado de uma forma o mais modular possível para facilitar eventuais alterações de componentes individuais, sem grande impacto no funcionamento geral;
- **Desempenho** - o componente deverá funcionar com taxas de resposta rápidas, de acordo com as tecnologias envolvidas.

3.4 Objectivos

Partindo dos requisitos descritos na secção anterior, o principal objectivo desta dissertação é descrição de uma implementação de um componente de gestão de workflows com a capacidade de:

- Criar e modelar processos de negócio;
- Modificar e publicar processos de negócio;
- Disponibilizar uma biblioteca com blocos de actividades pré-definidas.

A implementação do componente de gestão de workflows necessita da disponibilizar um ambiente integrado de desenvolvimento e modelação de processos facilitado pela gestão da interligação entre as diferentes actividades/workflows e pelo acesso a actividades/workflows pré-definidas(os), funções, operadores, dados e variáveis de contexto. O componente deverá disponibilizar a possibilidade de encapsular as actividades em blocos que caracterizarão um comportamento específico e que poderão possuir uma interface gráfica de configuração individual de forma a flexibilizar a utilização. São necessárias funcionalidades que permitem uma correcta e eficaz modelação dos processos de negócio, orientadas ao âmbito da solução global.

Especificação

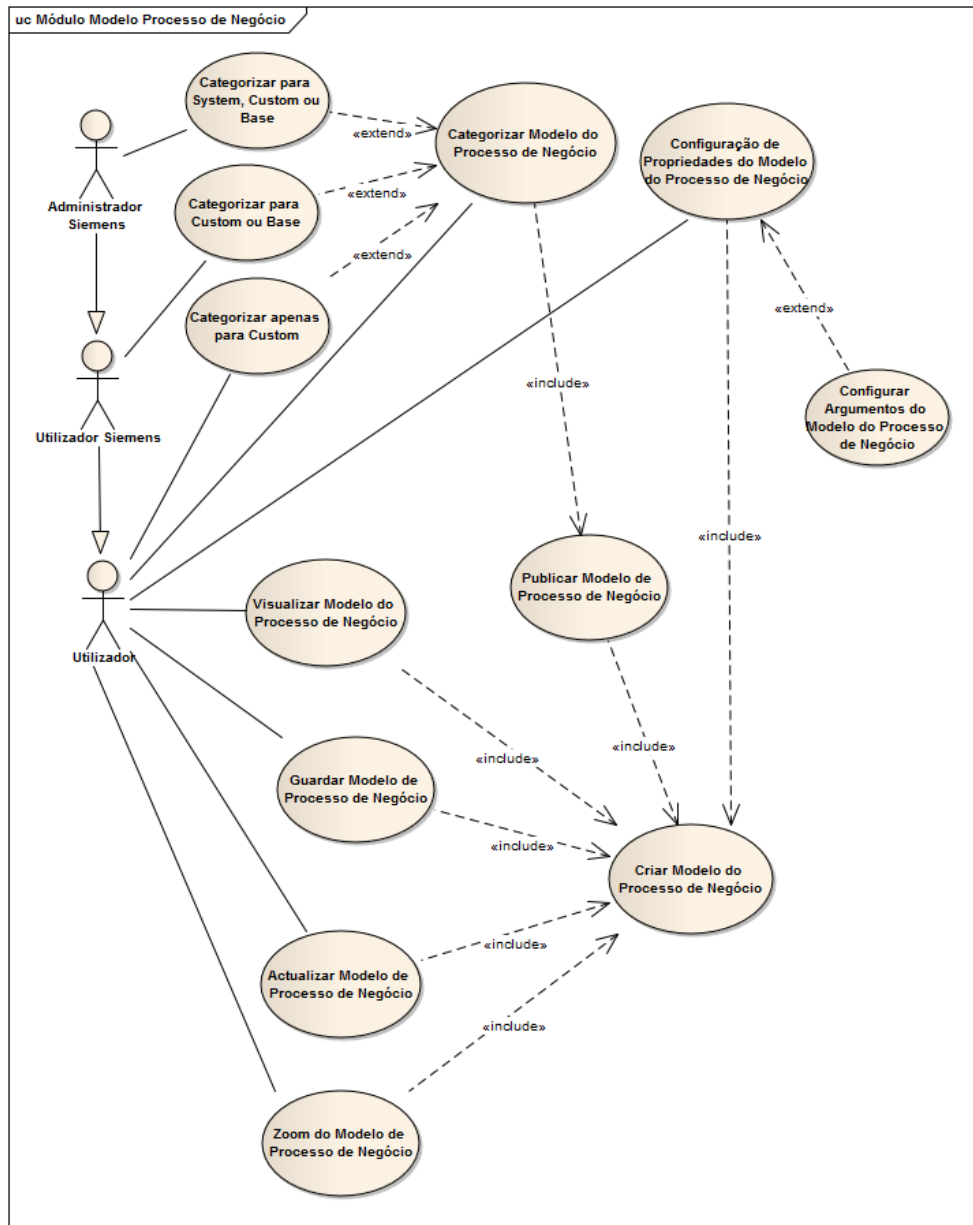


Figura 3.3: Diagrama de Casos de Uso

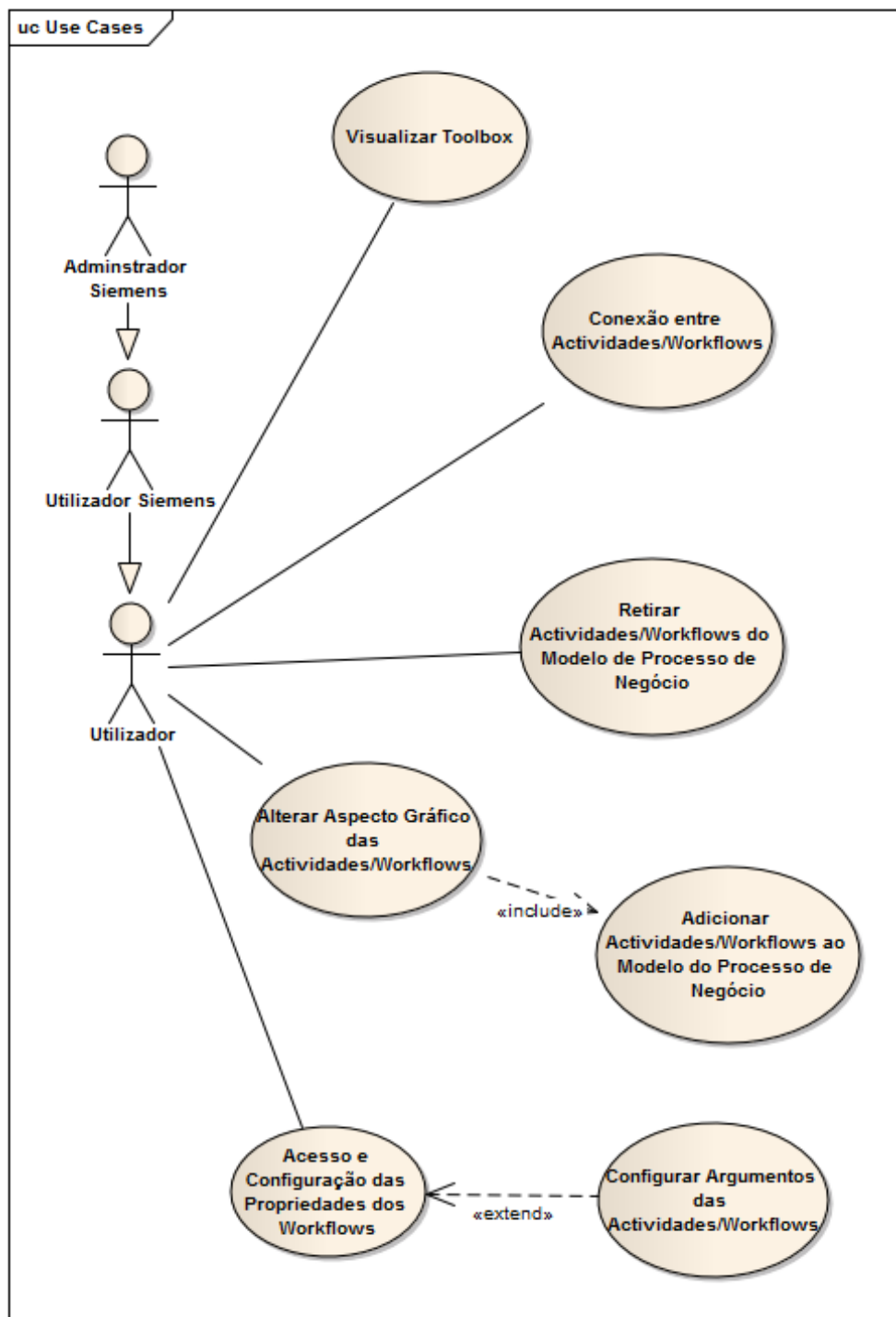


Figura 3.4: Diagrama de Casos de Uso (continuação)

Especificação

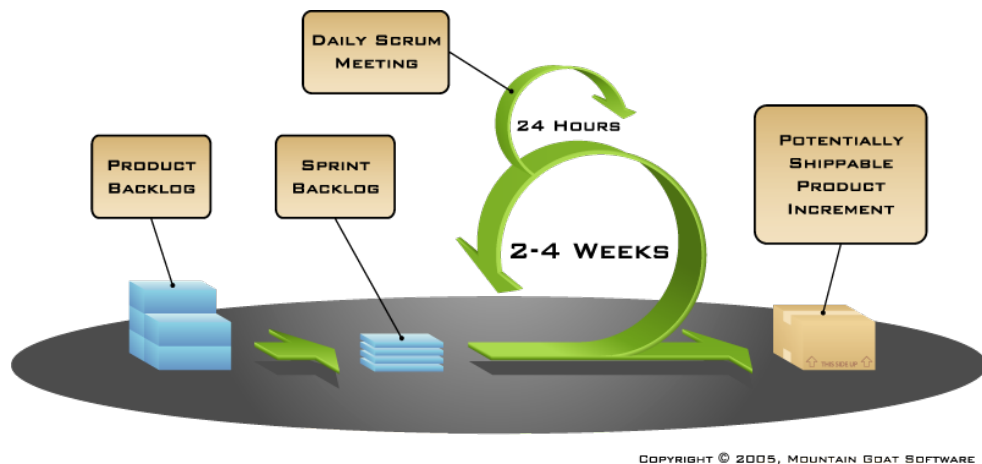


Figura 3.5: Processo de desenvolvimento iterativo e incremental Scrum [Sof11]

3.5 Metodologia

O componente de gestão de workflows foi desenvolvido utilizando uma metodologia ágil de gestão de projectos denominado por Scrum.

O Scrum é um processo de desenvolvimento de software iterativo e incremental, que permite manter a equipa focalizada na entrega de maior valor de negócio, no menor tempo possível. A figura 3.5 descreve o processo de desenvolvimento definido pelo Scrum.

No Scrum, os projectos são divididos em ciclos (normalmente de entre duas a quatro semanas) denominados de *sprints*. Cada *sprint* representa um limite de tempo dentro do qual um conjunto de tarefas devem ser executadas. As funcionalidades que devem ser implementadas no projecto são mantidas numa lista conhecida como *Product Backlog*. No início de cada *Sprint*, faz-se um *Sprint Planning Meeting*, ou seja, uma reunião de planeamento, para decidir quais as tarefas do *Product Backlog* que devem ser executadas dentro do intervalo de tempo do *Sprint*. Essas tarefas são transferidas do *Product Backlog* para o *Sprint Backlog*.

O Scrum possui três papéis fundamentais:

- **Product Owner** - é responsável por comunicar a visão do produto à equipa de desenvolvimento. Representa os interesses do cliente através dos requisitos e a respectiva priorização, ficando depois responsável por aceitar ou rejeitar o resultado do trabalho (gestão do *Product Backlog*);
- **Scrum Master** - é responsável por efectuar uma ligação entre o *Product Owner* e a equipa de desenvolvimento. O *Scrum Master* tenta remover impedimentos que surgem à equipa na concretização dos objectivos de cada *sprint*; O *Scrum Master* pode e deve ser um elemento da equipa;

- **Team Member** - na metodologia do Scrum, um *Team Member* é responsável por concretizar as tarefas de desenvolvimento, ou seja, determina como será alcançada cada tarefa em cada *sprint* [Coll1a].

Todos os dias, a equipa faz uma breve reunião, denominado de *Daily Scrum*. O objectivo é de partilhar com a equipa o estado actual do trabalho desenvolvido no dia anterior, e quais as actividades que cada um fará no próprio dia, identificando e debatendo impedimentos.

Para o desenvolvimento do componente de gestão de workflows, foram utilizados *sprints* com a duração de quatro semanas, com reuniões diárias e o respectivo *Sprint Review*. Esta metodologia foi bastante importante no desenvolvimento do componente da gestão de workflows, porque permitiu contrastar e debater diariamente ideias com os membros da equipa de desenvolvimento.

3.6 Estratégia

Pela possibilidade do *re-hosting*, é utilizado o WF4 como plataforma base para o desenvolvimento do componente de gestão de workflows. O objectivo da utilização do WF4 como plataforma base é para reaproveitar o aspecto gráfico do ambiente de modelação e respectivo motor de workflows para uma posterior execução, resultando numa diminuição do tempo e esforço de desenvolvimento. Neste sentido, a estratégia de desenvolvimento corresponde a uma adaptação do ambiente de modelação do WF4 às necessidades da solução, simplificando os seus métodos de configuração e passagem de parâmetros entre actividades/workflows.

A notação BPMN, além de ser bastante completa e intuitiva na representação de processos complexos e por fornecer um mapeamento entre os gráficos da notação e as construções subjacentes das linguagens de execução. Porém, devido à sua complexidade, não é conveniente para descrever o modelo do componente de gestão de workflows. Para esta finalidade é melhor uma notação básica de *flowchart*, que fornece ao utilizador um modelo gráfico fácil e simples para a criação/modificação do workflow, dispensando os diversos elementos da notação BPMN.

No WF4 a passagem de parâmetros entre actividades/workflows não é muito intuitiva, uma vez que, para a passagem de dados entre actividades, são necessárias referências a variáveis ou argumentos pré-definidos. A este nível, é necessário criar um método de atribuição de argumentos entre actividades diferente e mais intuitivo do método disponibilizado pelo WF4.

O mecanismo de *wizard* é um objectivo fundamental na definição da modelação, sendo este o formato em que o modelo do processo de negócio será executado. Um *wizard* (ou Assistente) é um padrão bastante utilizado na interface gráfica, que utiliza um esquema

passo-a-passo, e permite, de forma simples, realizar tarefas complexas numa aplicação. Assim, é indispensável implementar um mecanismo para que o modelo do processo de negócio seja deslocado facilmente para um passo anterior e seguinte, dependendo de um *input* fornecido pelo utilizador (e.g. "Anterior", "Seguinte").

Para que os passos, que representam actividades/workflows, sejam apresentados num ambiente de execução, num formato de *wizard*, é necessário criar uma forma de atribuição da apresentação gráfica a cada uma das actividades do modelo. Essa atribuição engloba um conjunto pré-definido de componentes, que contém, além da apresentação gráfica, a respectiva validação de campos, e a definição dos argumentos que são disponibilizados pelo componente.

Seguindo a estratégia definida neste capítulo, a fase de desenvolvimento divide-se em quatro partes fundamentais:

- *re-hosting* do ambiente de modelação do WF4;
- implementação de um fluxo de dados mais simples e intuitivo em relação ao disponibilizado pelo Windows Workflow 4.0;
- desenvolvimento de funcionalidades de configuração do modelo do processo de negócio e das respectivas actividades;
- adaptação do modelo do processo de negócio para este ser executado no motor de workflows do WF4;
- desenho de interface gráfica agradável e intuitiva;

3.7 Conclusão

Neste capítulo foi apresentada a definição do problema, constatando-se a necessidade de criar um solução (componente) de modelação de processo de negócio, utilizando como tecnologia base o WF4.

Os requisitos dessa solução foram propostos através de *user stories* e diagramas de casos de uso.

Foram também descritos os objectivos principais, metodologias e estratégias utilizadas no decorrer da fase de implementação.

Capítulo 4

Implementação

Neste capítulo estão detalhados os pormenores da implementação do componente de gestão de workflows.

Cada passo do modelo do processo de negócio corresponde a uma actividade ou workflow, dependendo se o passo é atómico ou se corresponde a um processo de negócio, anteriormente modelado por um utilizador. Neste próximo capítulo, para simplificar a identificação dos passos do modelo do processo de negócio, apenas serão referidos como actividades.

4.1 Tecnologia Base

Como foi mencionado anteriormente, o WF4 permite fazer o *re-hosting*. Esta funcionalidade permite desenvolver uma aplicação com as mesmas funcionalidades do WF4, e adaptá-las às necessidades da implementação. Existem três módulos que o WF4 dispõe:

- **Workflow Designer** - é o módulo principal que representa o modelo gráfico do workflow. Este módulo expõe a actual superfície de desenho do modelo através da propriedade *View*, e interliga as propriedades do modelo utilizando a vista *PropertyInspectorView*. O aspecto gráfico da vista do *Workflow Designer* é facilmente moldável às necessidades da implementação;
- **Toolbox** - é o módulo que representa o objecto *ToolboxControl*. Esse objecto possui um conjunto de objectos *ToolboxItemWrapper*, que definem o tipo de actividade disponível na *Toolbox*, possibilitando a deslocação da mesma para o modelo do workflow;
- **Property Grid** - este módulo corresponde ao painel de propriedades que é apresentado no Visual Studio 2010. Este painel de propriedades permite definir a atribuição de variáveis, expressões ou argumentos a cada uma das actividades presentes no modelo.

Implementação

Na implementação do componente de gestão de workflows, foram utilizados os módulos *Workflow Designer* e *Toolbox*. O módulo *Property Grid* não foi incluído na solução.

O objecto *WorkflowDesigner* inclui a definição do modelo do workflow e dos seus constituintes. Para alterar a apresentação gráfica do *Workflow Designer* e as funcionalidade do WF4 é necessário modificar as propriedades do objecto *WorkflowDesigner*. As propriedades que a constituem são:

- **Name** - identificação do objecto;
- **Context** - acesso a um objecto *EditingContext* que contém uma colecção de serviços, partilhados entre todos os elementos contidos no modelo, e utilizados para a interacção entre a aplicação e o *designer*;
- **ContextMenu** - acesso às definições das opções do menu de contexto, ou seja, o menu que é acessível através do clique direito do rato no modelo do workflow;
- **DebugManagerView** - fornece um serviço utilizado para *debug*;
- **PropertyGridFontAndColorData** - modifica o tipo de letra e a cor da *Property Grid*;
- **PropertyInspectorView** - retorna um objecto *UIElement* que permite ao utilizador visualizar e editar propriedades do workflow;
- **Text** - acesso a um código XAML em formato de *string* que representa o modelo do workflow;
- **View** - retorna um objecto *UIElement* que representa a vista gráfica do modelo do workflow.

O ambiente de modelação está separado em duas vistas: i) uma vista correspondente à *Toolbox*, e uma vista gráfica do modelo que corresponde ao *Workflow Designer*. A *Toolbox* pode ser expandida ou minimizada para dar mais espaço à modelação. Na criação de um novo workflow, o *Workflow Designer* é inicializado com um modelo *Flowchart*, contendo apenas um nó de inicialização. A barra disponibilizada pelo *Workflow Designer* contém a definição dos argumentos, variáveis, opções de zoom e o mini-map. Porém, essa barra foi retirada da vista principal, apesar das variáveis e argumentos pré-definidos continuarem instanciados.

A *Toolbox* é instanciada com um objecto *ToolboxControl*, que corresponde a uma colecção de tipos de actividades e workflows pré-definidos. Os tipos de actividades e workflows que estão definidos na *Toolbox* podem ser arrastados (*drag-and-drop*) para o *Workflow Designer*, o que instanciará um objecto do tipo da actividade/workflow definido na *Toolbox*, e carregado (visualmente e internamente) para o modelo.

Os workflows presentes na *Toolbox*, correspondem a uma das quatro categorias:

- **Framework** - correspondem a actividades disponibilizadas pela *framework* do WF4;
- **System** - correspondem a actividades/workflows disponibilizados pelos administradores da Siemens, que não podem ser abertos por outros utilizadores para leitura nem escrita;
- **Base** - correspondem a actividades/workflows disponibilizados pelos utilizadores da Siemens, que não podem ser abertos por utilizadores para leitura nem escrita, mas que pode ser feita uma cópia se forem configurados para tal;
- **Custom** - correspondem a workflows modelados pelos utilizadores, que podem ser abertos para leitura e escrita.

4.2 Notação Base

Neste capítulo são descritas as funcionalidades implementadas que tiveram impacto na notação do modelo do processo de negócio.

4.2.1 Fluxo de Dados

Para melhorar e facilitar a atribuição de parâmetros entre actividades, foi utilizada uma definição do fluxo de dados diferente da oferecida pelo WF4. A estratégia foi criar uma própria estrutura de fluxo de dados para o modelo, e posterior tradução desse fluxo de dados para ser interpretado pelo motor de workflows do WF4.

O WF4 segue uma estratégia em que a atribuição dos valores dos argumento de entrada de cada actividade é obtida através do acesso a variáveis globais ou argumentos definidos no momento da modelação, e acedendo à barra inferior do *Workflow Designer*. Para tornar o fluxo de dados mais intuitivo e simples de definir, utilizou-se uma estratégia baseada no padrão *pipes and filters*, utilizando um mapeamento e atribuição de argumentos a outros definidos anteriormente, através da propriedade *ModelPath*.

O *ModelPath* é uma propriedade do objecto *ModelItem*. Cada actividade inserida no modelo é representado como um objecto *ModelItem*. Todos os *ModelItem* presentes no modelo podem ser acedidos através da propriedade *Context* do objecto *WorkflowDesigner*. Cada propriedade *ModelPath* indica a posição da actividade (*ModelItem*) relativamente ao modelo do workflow, tornando possível, criar um filtro de argumentos disponíveis e antecedentes em cada actividade, para a atribuição do valor aos seus argumentos.

O padrão *pipes and filters* é definido quando existe uma tarefa complexa que pode ser sub-dividida num número de pequenas tarefas, das quais podem ser definidas através de operações ou serviços independentes entre si. Cada uma dessas tarefas (filtros) transforma um conjunto de dados de entrada (*input*) num conjunto de dados de saída (*output*).

Implementação

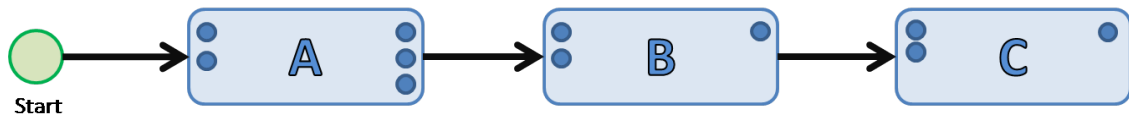


Figura 4.1: Fluxo de controlo de um workflow

Esta estrutura equivale a um workflow constituído por diversas actividades que trocam argumentos de dados entre si.

Num padrão *pipes and filters*, a troca de argumentos entre actividades não deve ser realizada com base num componente monolítico, porque seria demasiado complexo e constituiria um obstáculo em termos de modificação e reutilização de cada componente. Além disso, os dados de entrada das respectivas actividades devem ser devidamente fornecidos. Para atingir esse objetivo, deve ser possível compor de forma flexível a passagem de dados entre actividades, de acordo com as necessidades do utilizador final. [AZ04].

Para melhor entender esta definição do fluxo de dados, na figura 4.1 está ilustrado um exemplo de um workflow com um fluxo de controlo bem-definido. A actividade A é a primeira a ser inicializada após o nó de inicialização (*Start*). A actividade B e C são posteriormente executadas por esta ordem. Para facilitar a atribuição de argumentos em cada actividade, é utilizado um fluxo de dados representado no esquema da figura 4.2. Esta solução permitirá, acedendo ao painel de propriedades de cada actividade, atribuir a cada um dos argumentos de *input* um argumento de *output* de actividades anteriores ao seu *ModelPath*.

A actividade B tem acesso aos argumentos de *output* da actividade A, por ser consecutiva à actividade B relativamente ao *ModelPath*. Neste sentido, a actividade B poderá atribuir aos seus argumentos de *input* X e Y os argumentos de *output* W e V da actividade A, respectivamente. Consequentemente a actividade C poderá atribuir aos seus argumentos de *output* quaisquer argumentos da actividade A e B, atribuindo ao argumento de *input* X e Y, os argumentos de *output* Z da actividade A e o argumento de *output* Z da actividade B, respectivamente.

Para melhor entender este mecanismo de atribuição de argumentos, um exemplo simples de um workflow, baseado na figura 4.2, é demonstrado se a actividade A realizar uma soma, uma multiplicação e uma subtracção dos seus dois valores de *input*, a actividade B efectuar apenas a soma dos seus valores de *input*, e a actividade C executar uma multiplicação dos dois valores de entrada, a equação do workflow era a seguinte:

$$Z_A(X_A, Y_A) = X_A + Y_A \quad (4.1)$$

$$W_A(X_A, Y_A) = X_A * Y_A \quad (4.2)$$

$$V_A(X_A, Y_A) = X_A - Y_A \quad (4.3)$$

Implementação

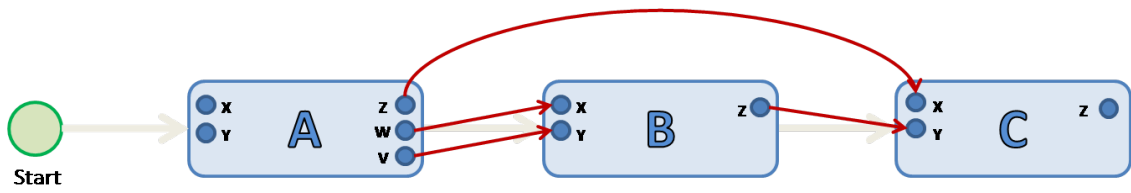


Figura 4.2: Fluxo de dados de um workflow

$$Z_B(X_B, Y_B) = X_B + Y_B \quad (4.4)$$

$$Z_C(X_C, Y_C) = X_C * Y_C \quad (4.5)$$

$$(4.6)$$

Para $X_A = 2, X_B = 1$, temos que:

$$Z_A(X_A, Y_A) = 2 + 1 = 3 \quad (4.7)$$

$$W_A(X_A, Y_A) = 2 * 1 = 2 \quad (4.8)$$

$$V_A(X_A, Y_A) = 2 - 1 = 1 \quad (4.9)$$

$$Z_B(X_B, Y_B) = 2 + 1 = 3 \quad (4.10)$$

$$Z_C(X_C, Y_C) = 3 * 3 = 9 \quad (4.11)$$

$$(4.12)$$

Com esta definição das operações, o argumento de *output* Z da actividade C produz o valor 9 no final da execução do workflow.

Para melhor representar as variáveis globais do modelo do workflow, foram definidas variáveis que são apresentadas como argumentos de *input* do respectivo modelo. Deste modo, é necessário também proceder à atribuição dos *outputs* do modelo do workflow. Além da atribuição de um identificador ao respectivo argumento de *output*, deverá também ser atribuído um valor que corresponde a qualquer argumento de *output* das actividades presentes no modelo. A figura 4.3 é uma extensão do modelo do workflow da figura 4.2, em que são utilizados os argumentos *input* do modelo para definir as variáveis a serem passadas após o início da execução, e poderão ser atribuídos os argumentos de *output* do próprio workflow. Cada argumento pode ser atribuído mais do que uma vez, como é o caso do argumento Z_B , que é atribuído ao argumento Y_C e a um dos *outputs* do workflow.

Para a atribuição de argumentos entre actividades utiliza-se um mecanismo designado *Object Browser*, que organiza devidamente os argumentos disponíveis para a atribuição.

Quando um workflow é reutilizado no modelo de um processo de negócio, os argumentos de *input* e de *output* são os argumentos globais que foram definidos no momento

Implementação

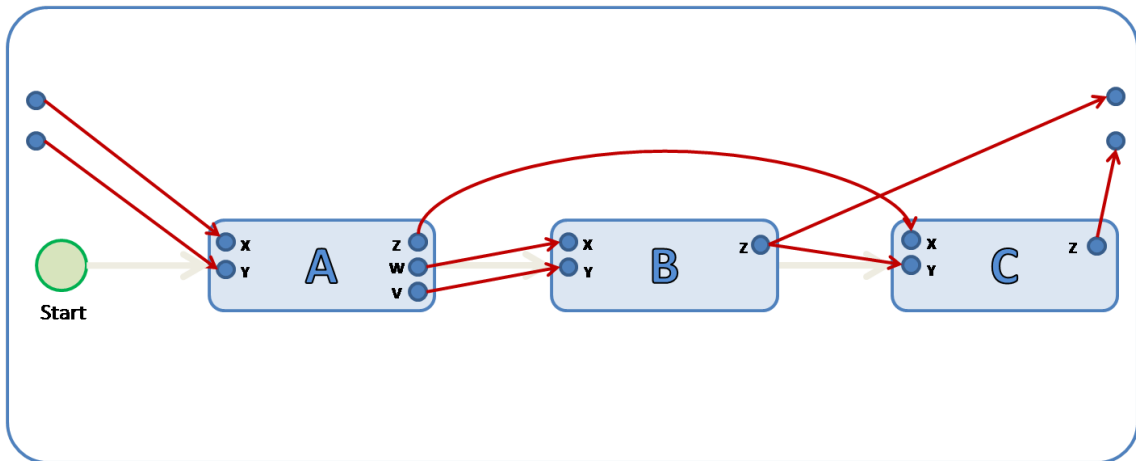


Figura 4.3: Fluxo de dados de um workflow utilizando os argumentos globais

da modelação.

As propriedades de cada workflow/actividade que definem os seus argumentos são o *InputArguments* e o *OutputArguments*, que correspondem respectivamente, aos argumentos de *input* e aos argumentos de *output*. Estas propriedades são do tipo *string*, mas representam uma estrutura em XML. Os formatos XML dessas propriedades são:

```
<InputArguments>
  <Argument Name='[Name]' Type='[Type]'> [Value]</Argument>
  ...
</InputArguments>

<OutputArguments>
  <Argument Name='[Name]' Type='[Type]'> [Value]</Argument>
  ...
</OutputArguments>
```

O número de nós *Argument* depende do número de argumentos de *input/output*. Para cada argumento são definidos o nome e o tipo através dos atributos *Name* e *Type*, respectivamente, e ainda o valor da atribuição a algum outro argumento, que corresponde ao valor do nó *Argument*. O valor da atribuição é identificado através da forma '*Actividade:Argumento*'. Por exemplo, se atribuirmos a um argumento de *input* 'X' de actividade B um argumento de *output* 'Z' de uma actividade com identificador 'A', então, se esse argumento for do tipo *string*, a propriedade *InputArguments* da actividade B será:

Implementação



Figura 4.4: Representação gráfica de uma actividade

```
<InputArguments>  
  <Argument Name='X' Type='string'>A:Z</Argument>  
</InputArguments>
```

Deste modo, identifica-se claramente a atribuição de cada um dos argumentos das actividades. Se uma atribuição corresponder a um argumento global do modelo, então não é atribuído o identificador 'Actividade' em *Value*.

4.2.2 Actividades

Após ser arrastada uma actividade para o ambiente gráfico do *Workflow Designer* utilizando a descrição da mesma indicada na *Toolbox*, será apresentado um rectângulo que representa a actividade no modelo. Um exemplo da sua representação gráfica está ilustrada na figura 4.4.

Cada passo do modelo do processo de negócio corresponde a uma actividade ou workflow, dependendo se o passo é atómico ou se corresponde a um processo de negócio anteriormente modelado por um utilizador. Deste modo, um workflow quando é reutilizado num modelo do processo de negócio é representado de uma forma diferente da actividade. A sua representação gráfica está ilustrada na figura 4.5.

Na apresentação gráfica da actividade ou do workflow está presente uma pequena descrição da mesma, que pode ser modificada localmente no modelo. Esta descrição pretende dar apenas uma breve ideia da funcionalidade do passo. Para além da descrição, também é possível modificar a imagem que representa a actividade ou workflow. O utilizador, ao seleccionar a mudança de ícone, tem acesso a um conjunto de imagens pré-definidas, ou pode acrescentar novas, conforme a necessidade de representar a actividade ou workflow. Esta funcionalidade de mudar a imagem da actividade pretende, juntamente com a descrição, realçar a função da actividade ou workflow.

Para que actividades ou workflows sejam incluídos no *ModelPath* do modelo, ou seja, executados juntamente com a execução do respectivo modelo, necessitarão de serem ligados a um outro nó do modelo (actividade/workflow ou nó de inicialização). Essa ligação

Implementação

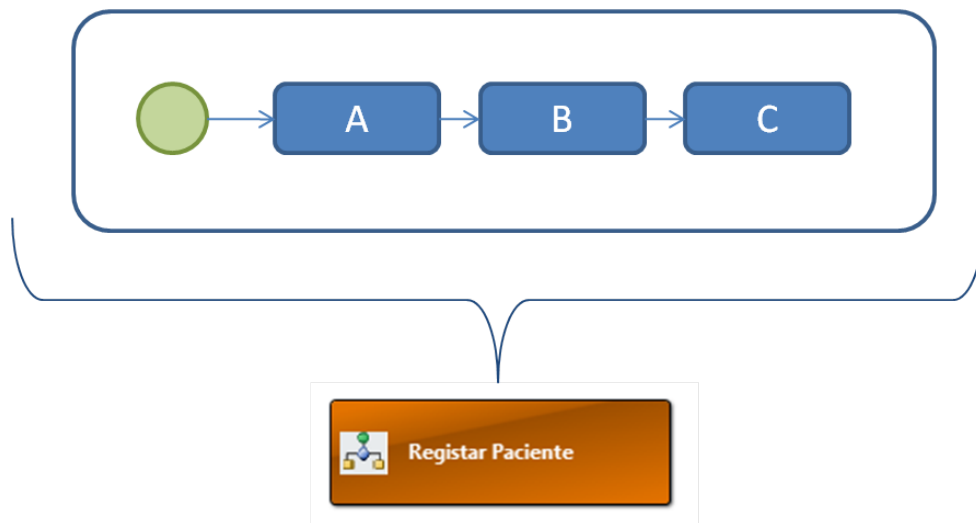


Figura 4.5: Representação gráfica de um workflow anteriormente modelado e reutilizado no modelo de outro workflow

obtêm-se arrastando as setas localizadas nos lados do rectângulo da actividade/workflow.

4.3 Ambiente Gráfico

Neste capítulo são descritas as funcionalidades que foram implementadas com impacto no ambiente gráfico do componente de gestão de workflows.

4.3.1 Painel de Configuração

O processo de negócio que está a ser modelado e as respectivas actividades/workflows têm acesso a um painel de configuração, onde poderão ser configuradas as respectivas propriedades.

As propriedades do workflow e das actividades estão divididas em seis áreas:

- **Header** - informação do modelo do workflow (apenas para escrita) e da actividade (apenas para leitura);
- **Frontend** - propriedades necessárias para a apresentação do modelo do workflow em tempo de execução (apenas disponível nas propriedades do modelo);
- **Designer Runtime Behavior** - utilizado para carregar e configurar uma área de configuração de propriedades do workflow, com impacto na reutilização do mesmo;
- **Input Arguments** - área de configuração de argumentos de *input*;
- **Output Arguments** - área de configuração de argumentos de *output*;

Implementação

- **Business Component** - utilizada para atribuir, ao nível da actividade/workflow, um componente com impacto na execução do mesmo. Cada componente disponibiliza um conjunto de propriedades, incluindo uma interface gráfica, respectiva validação e definição de argumentos que poderão ser atribuídos a argumentos de *input* da respectiva actividade/workflow;

A informação disponível na área do *Header* identifica as propriedades principais de qualquer actividade/workflow. Quando um processo de negócio está a ser modelado, a área do *Header* apresenta quatro campos para preenchimento:

- **Name** - nome do modelo;
- **Description** - descrição do modelo;
- **Runtime Environment** - ambiente de execução do modelo (pode ser *UI*, para ser visível em ambiente de execução, ou *Batch*, utilizado como um conjunto de tarefas sequenciais);
- **Notes** - notas informativas agregadas ao modelo;

Quando um modelo do workflow é publicado e reutilizado no modelo de outro workflow, a área correspondente ao *Header* apenas é utilizada para a leitura das propriedades, algumas delas apenas configuráveis no momento da modelação. Além da descrição das propriedades anteriormente referidas do *Header*, contém também:

- **Version** - versão publicada do workflow;
- **Unique Code** - código de identificação do workflow;
- **Author Properties** - área que contém a informação do autor do workflow, nomeadamente o nome do autor e data da criação e da última modificação;

Quando o *Runtime Environment* seleccionado corresponder a *UI*, será visível no painel de configuração uma área reservada à informação que será mostrada no ambiente de execução. Essa área designada por *Frontend* contém os campos:

- **Image** - imagem que será utilizada para identificar o workflow no ambiente de execução;
- **Name** - nome que será utilizado para identificar o workflow no ambiente de execução;
- **Description** - descrição que será disponibilizada no ambiente de execução;
- **Help** - informação que ajuda a execução do workflow;
- **Tooltip** - informação que acrescenta informação adicional ao utilizador na execução do workflow;

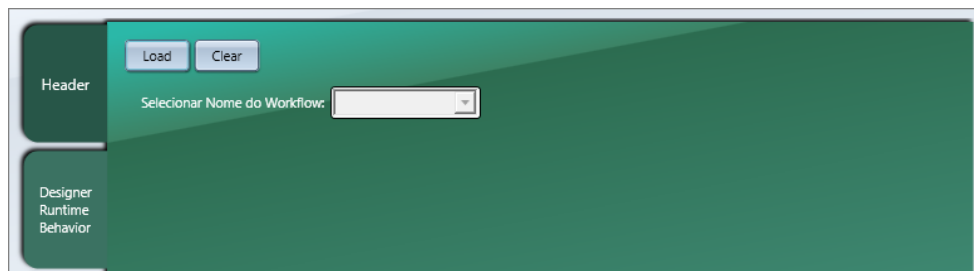


Figura 4.6: Painel de configuração na área do DRB no momento da modelação

4.3.2 Designer Runtime Behavior

O comportamento do Designer Runtime Behavior (DRB) permite configurar modelos de workflows que mais tarde são reutilizados num outro modelo de workflow. Quando um workflow é reutilizado, é possível que determinados *outputs* desse workflow possam ser definidos com a selecção de uma opção ou configuração de uma propriedade. Assim, a definição das propriedades configuráveis do DRB no momento da reutilização, necessita de ser atribuída no momento da modelação.

Quando um processo de negócio está a ser modelado, o utilizador poderá carregar um código XAML utilizando a opção "Load", disponível na área Designer Runtime Behavior do painel de configuração. Quando esse código é carregado, a interface da área do DRB modifica, transpondo todo o conteúdo do código XAML carregado para a propriedade *ConfigApp*, pertencente às propriedades do modelo do workflow.

Após a publicação do modelo do workflow, a sua definição é colocada na *Toolbox*, incluindo todas as suas propriedades. Assim, quando esse mesmo workflow é arrastado para um outro modelo, o painel do DRB é automaticamente apresentada para a selecção de todas as opções de configuração disponíveis.

Para melhor compreender este conceito, um exemplo dos dois painéis DRB, no momento da modelação e no momento da reutilização, estão ilustrados na Figura 4.6 e 4.7, respectivamente. Neste exemplo, é carregado um código XAML para a propriedade *ConfigApp* do workflow modelado, que apresenta uma *ComboBox* com a funcionalidade de seleccionar um nome de um workflow já publicado. Este código XAML é atribuído entre vários pré-definidos. O código XAML da propriedade *ConfigApp* anteriormente atribuída, será apresentada no painel de configuração no momento em que este workflow é colocado no ambiente gráfico do modelo.

Como vimos anteriormente, quando o código XAML é carregado, a propriedade *ConfigApp* é modificada, e conseqüentemente, os argumentos de *input* e *output* do próprio modelo. Para a implementação desta funcionalidade, existe um mapeamento entre o código XAML e os argumentos. Admitindo que o código XAML carregado da Figura 4.6, está representado com a seguinte estrutura:

Implementação

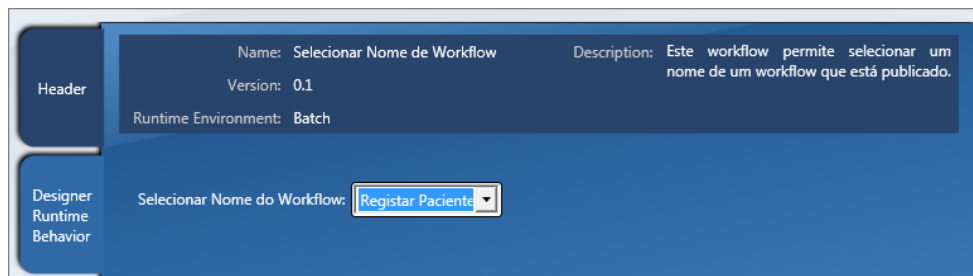


Figura 4.7: Painel de configuração na área do DRB no momento da reutilização

<Grid>

<StackPanel Margin="10"Orientation="Horizontal">

<Label Foreground="White"> Selecionar Nome do Workflow: </Label>

<ComboBox

VerticalAlignment="Top"

HorizontalAlignment="Left"

Width="120"

ItemsSource="Binding WorkflowNames"

SelectedItem="{Binding **ConfigProperties[Property1]**, Mode=TwoWay}"

SelectedValuePath="Value"/>

</StackPanel>

</Grid>

Através de um *parser* da propriedade *ConfigApp*, obtemos uma referência ao objecto *ConfigProperties*, que corresponde ao mapeamento entre o nome da propriedade (neste caso *Property1*) e o objecto seleccionado pela *ComboBox*. Utilizando esta referência a *Property1*, é possível criar um argumento de *input* e outro de *output* que corresponde à propriedade carregada, através da seguinte definição:

<InputArgument>

<Argument Name="Property1"Type="System.String">

ConfigBinding:Property1

</Argument>

</InputArgument>

<OutputArgument>

<Argument Name="Property1"Type="System.String">

ConfigBinding:Property1

</Argument>

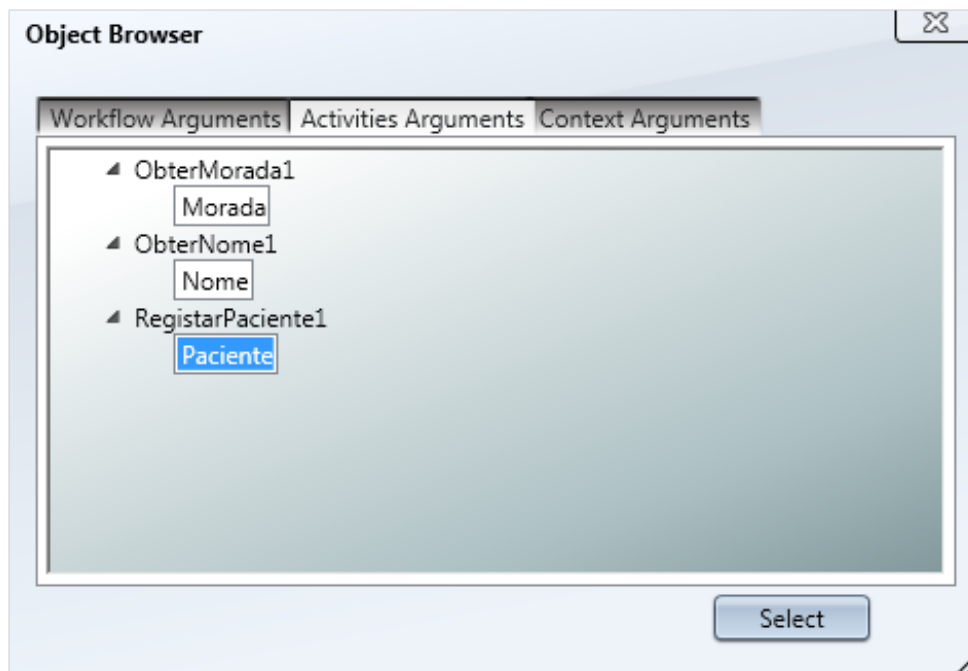


Figura 4.8: Organização em árvore dos argumentos das actividades/workflows do Object Browser

</OutputArgument>

Desta forma, será criado um argumento de *input* e outro de *output* com o mesmo nome e tipo da propriedade referenciada no código XAML. O valor do nó *Argument* corresponde a uma associação entre a *string* "ConfigBinding" e o nome da propriedade carregada. Assim, no momento da reutilização do workflow, existe uma associação entre os argumentos e o painel de configuração do DRB. O mapeamento entre os argumentos e o painel de configuração foi necessário para o utilizador disponibilizar a propriedade de configuração para atribuição de argumentos de outras actividades (justificando a criação do argumento de *output*), como para a respectiva validação do valor (justificando a criação do argumento de *input*).

4.3.3 Object Browser

Para a atribuição e mapeamento de argumentos, foi desenvolvida uma organização dos argumentos disponíveis para atribuição. Esta organização, denominada por *Object Browser*, permite aceder de forma rápida e simples ao conjunto de argumentos que a actividade/workflow tem acesso. Em cada definição de argumento, na área de Input/Output Arguments do painel de configuração, está disponível uma opção para o acesso a uma janela com o aspecto da figura 4.8. Nesta janela estão disponíveis quatro tipos de argumentos:

- **Workflow Arguments** - argumentos de *input* do respectivo modelo;
- **Activities Arguments** - argumentos das actividades presentemente ligadas ao modelo;
- **Context Arguments** - argumentos/variáveis de contexto (e.g. objectos de sessão, valores por omissão);
- **Business Component Arguments** - argumentos obtidos através das associações de componentes de negócio;

Os argumentos das actividades estão organizados em árvore, a cada nó de primeiro nível corresponde a uma actividade presente no modelo, e as folhas correspondem aos argumentos disponíveis. Seleccionando o argumento desejado, é feita uma atribuição de argumentos. Um exemplo de organização em árvore dos argumentos das actividades está ilustrado na figura 4.8. Neste exemplo, o argumento da actividade (ou workflow) que está a ser atribuído, apenas têm acesso aos argumentos das três actividades: RegistrarPaciente1, ObterNome1, e ObterMorada1. Esses argumentos são, respectivamente: Paciente, Nome e Morada.

4.3.4 Atribuição de Business Components

Cada actividade corresponde a um passo do workflow, que será executado sequencialmente utilizando um mecanismo de *wizard*. Para tal implementação será necessário associar a cada actividade uma *User Interface* (UI). Esta UI será visualizada no passo correspondente à actividade em execução.

Para tal funcionalidade, está definida no painel de configuração de cada actividade, na área *Business Components*, uma opção para escolher um componente que permitirá associar a UI mais adequada para esse passo. Esta selecção do componente é com recurso a uma lista de componentes pré-definidos que contêm os aspectos visuais necessários para um passo do workflow, nomeadamente a respectiva validação de campos e os argumentos de saída disponibilizados (que podem ser mais tarde atribuídos à actividade correspondente através do *Object Browser*).

4.3.5 Navegação

Foi necessária a implementação de um mecanismo no modelo para permitir a navegação entre os passos (actividades).

No momento da execução, o motor de workflows da WF4, executa sequencialmente o modelo *flowchart* e avança sempre para a próxima actividade. A única forma de voltar a uma actividade anteriormente executada, é através da inserção de um nó de decisão

(*FlowDecision*) ou um nó comutador (*FlowSwitch*) no modelo. Estes nós necessitam de uma condição para avaliar o próximo passo.

Para simplificar o processo de modelação dos workflow, mantendo a filosofia de *wizards*, mas retirando a necessidade de inserir os nós de decisão e comutador, foi criada uma solução baseada na construção de um *flowchart* lógico representativo do *flowchart* físico (visível). Esse *flowchart* lógico contém um nó *FlowSwitch* que utiliza uma variável pré-definida (denominada "NextStep") utilizada para invocar a próxima actividade a ser executada.

Como foi referenciado no capítulo 4.2.1, existe um *ModelPath* para cada *ModelItem* (actividade/workflow) presente no modelo. Esse *ModelPath* permite saber a posição da actividade relativamente ao modelo. Dependendo do valor do *ModelPath*, cada actividade do modelo, será copiada para um *flowchart* lógico, e ligado ao *FlowSwitch* que redirecionará o fluxo de controlo para a respectiva actividade, caso o valor da variável *NextStep* for igual ao seu *ModelPath*. Para explicar melhor esta funcionalidade, nas figuras 4.9 e 4.10 estão ilustrados ambos os *flowchart*, o lógico e o físico.

O *flowchart* físico do exemplo da figura 4.9 contém apenas conectores unidireccionais. E indica, neste exemplo, que o workflow executa sequencialmente as seguintes actividades: i) Obter Nome do Paciente, ii) Obter Morada do Paciente, iii) Registrar Paciente, iv) Visualizar Informação do Paciente. Uma forma de tornar esta sequência bidireccional (ou seja, possibilitar o avanço e recuo entre actividades), é a criação do *flowchart* lógico da figura 4.10. Este *flowchart* lógico é inicialmente composto por um *FlowSwitch*. Todas as actividades inseridas no *flowchart* físico são copiadas para o modelo da figura 4.10, ligando-os ao *FlowSwitch* com o respectivo caso que corresponde ao *ModelPath* (neste exemplo, a actividade "Obter Nome do Paciente" corresponde ao caso 1 do *FlowSwitch*, a "Obter Morada do Paciente" ao caso 2, a "Registrar Paciente" ao caso 3, e a "Visualizar Informação do Paciente" ao caso 4).

Assim, no momento em que o modelo do processo de negócio é publicado, apenas o *flowchart* lógico é gravado e executado, sendo o modelo de *flowchart* físico apenas necessário para a respectiva visualização gráfica e modelação. Com este mecanismo, cada actividade poderá indicar o seu próximo e anterior passo, dependendo ou não de uma condição. Bastará à actividade modificar o valor da variável *NextStep*, definida no conjunto de variáveis do modelo do *Workflow Designer*.

4.4 Integração com o Workflow Engine

Para a execução conveniente do fluxo de dados utilizando o *Workflow Engine* do WF4, é utilizado um objecto designado por *Message*. Este objecto actua como uma interface entre actividades, transportando o fluxo de dados do modelo do workflow.

Implementação

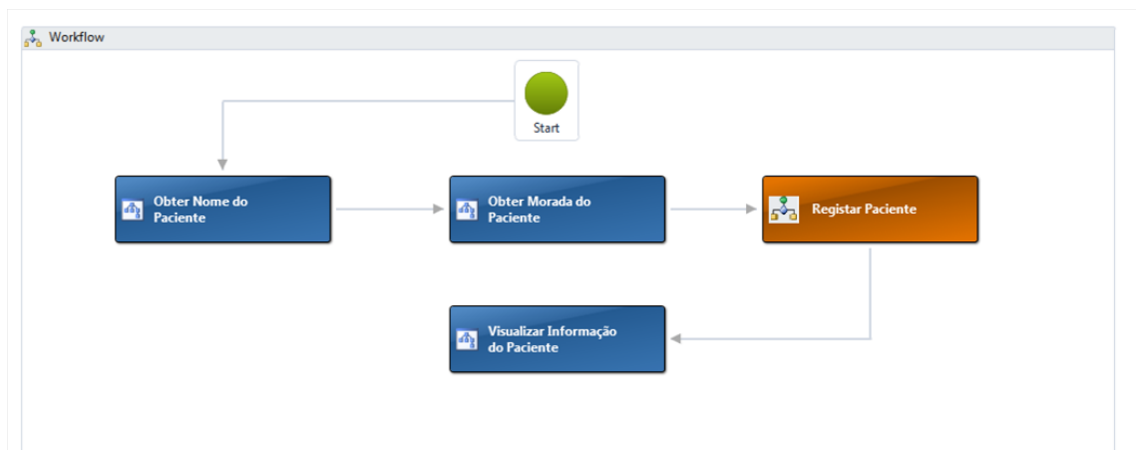


Figura 4.9: Representação do Flowchart visível no ambiente de modelação

Na execução do modelo de workflow poderão inicialmente ser atribuídos *inputs* que inicializam o conteúdo do fluxo de dados. O único *input* do modelo é o objecto *Message* anteriormente referido. Qualquer argumento presente na execução de actividades é incluído no objecto *Message*, e apenas esse objecto é utilizado como fluxo de dados do WF4.

Para a inclusão de um argumento de *output* no objecto *Message* é utilizado um identificador, ou seja, o nome da actividade em que foi definido juntamente com o nome do argumento. Imagine que uma actividade A atribui o número 5 a um seu argumento de *output* denominado "Number", então no objecto *Message* é incluído o identificador "A:Number" mapeado com o valor 5.

Uma atribuição de um argumento de *input* a uma actividade equivale, em tempo de execução, o acesso ao valor do argumento atribuído localizado no objecto *Message*. Todo este mecanismo de obtenção de um valor do argumento é alcançado através da definição das propriedades em formato XAML *InputArguments* e *OutputArguments*, anteriormente referidos no capítulo 4.2.1. Assim, no exemplo anteriormente referido, quando um utilizador atribui a um argumento de *input* o argumento de *output* "Number" da actividade A, no momento de execução, esta funcionalidade traduz-se num acesso a um objecto disponível na colecção do objecto *Message*, identificado com "A:Number" (que corresponde neste exemplo ao valor 5).

O objecto *Message* funciona como uma interface entre actividades, que conduz o fluxo de dados ao longo da execução do workflow. Este objecto é progressivamente alimentado pelas actividades do modelo. O esquema da figura 4.11 ilustra esse mecanismo de acesso ao objecto *Message*. Neste exemplo, a actividade "Registrar Paciente" adiciona um valor no objecto *Message* que equivale ao registo de um paciente. Mais tarde, a actividade "Visualizar Informação do Paciente" acede ao registo do Paciente através do objecto *Message*.

Um outro aspecto bastante importante, é a respectiva execução do workflow, que é

Implementação

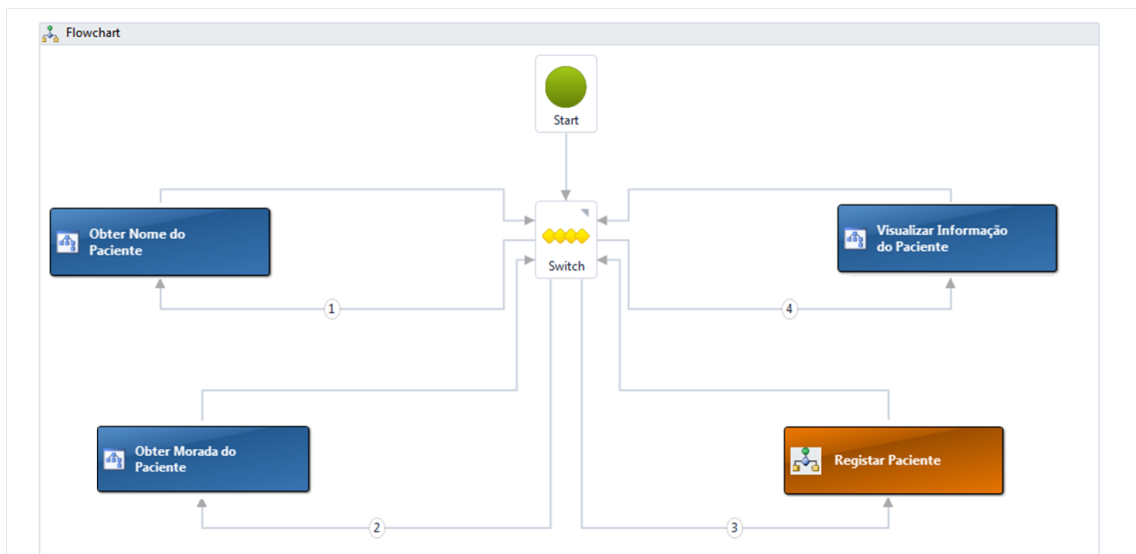


Figura 4.10: Representação do Flowchart lógico representativo do ambiente de modelação

controlada através de um componente denominado *Workflow Controller*, que tem como objectivo a condução do fluxo de controlo do modelo.

A figura 4.12 ilustra um diagrama representativo do controlo que é necessário para a execução do workflow, utilizando o *Workflow Controller* e realçando o tratamento em cada actividade.

A figura 4.12 ilustra o conjunto de tarefas que o *Workflow Controller* necessita realizar para a adequada execução das actividades do modelo do workflow. Este conjunto de tarefas permitirá a correcta implementação do mecanismo de *wizards*.

O workflow é inicializado através do objecto *WorkflowApplication*. Este objecto disponibiliza um conjunto de funcionalidades para a gestão de instâncias de workflows (e.g. métodos para criação e carregamento de instâncias, pausa, continuação, terminação e notificação do ciclo de vida dos eventos).

Enquanto o fluxo de execução não terminar, ou seja, sempre que a variável *NextStep* seja diferente de qualquer valor de caso do *FlowSwitch*, o *Workflow Controller* executa a respectiva actividade, dependendo do valor da variável *NextStep*.

A função *Execution*, derivada da classe *NativeActivity*, pressupõe o seu acesso quando a actividade em questão é executada. No interior da função *Execution* são criados três *Bookmarks*. Um *Bookmark* é um mecanismo que permite a espera passiva de uma actividade por um *input* da execução do workflow, ou seja, representa um ponto onde a execução pode ser suspensa e resumida utilizando o motor de workflows. Na execução de um *Bookmark* podem ser transmitidos dados entre o *Workflow Controller* e a respectiva actividade em execução. Após a finalização da função que corresponde ao *Bookmark*, a execução do workflow continua.

Implementação

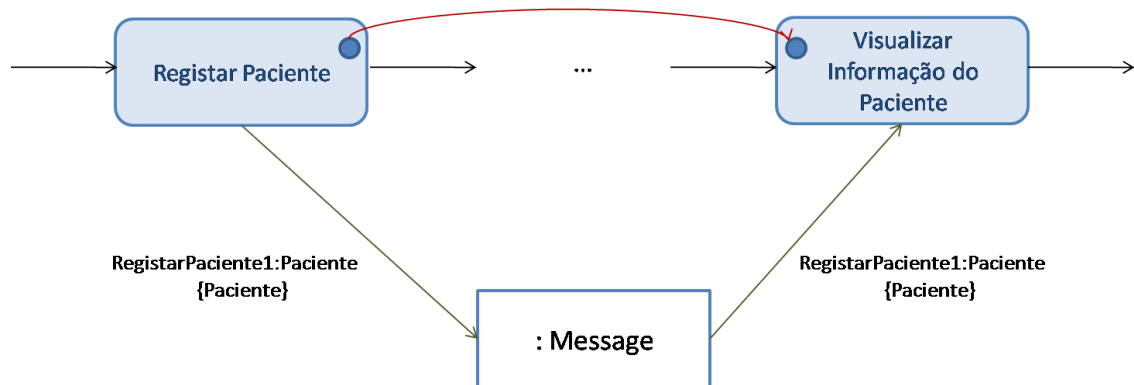


Figura 4.11: Acesso ao objecto *Message* para integração com o fluxo de dados do *Workflow Engine*

Enquanto os *Bookmarks* não forem processados pelo *Workflow Controller* o fluxo de execução suspende. Assim, o *Workflow Controller* processa os três *Bookmarks* que a actividade em execução instanciou. Esses *Bookmarks* têm a seguinte função:

- **Bookmark de Envio de BC (Business Component)** - envio da informação do *Business Component*, atribuído na fase da modelação, ao *Workflow Controller*;
- **Bookmark de Execução** - validação dos dados recebidos (dependendo dos argumentos de *input* e *output* definidos) e execução da lógica da actividade;
- **Bookmark do Próximo Passo** - tem como função calcular o próximo passo do workflow, dependendo da opção do utilizador (de avançar ou recuar o passo de execução) ou através de uma decisão;

Estes *Bookmarks* são executados sequencialmente e na actividade respectiva. É de salientar que, utilizando este mecanismo, o workflow garante sempre a execução de apenas uma actividade, construindo assim o mecanismo de *wizard*. O *Workflow Controller* necessitará de fornecer *inputs* para o conhecimento da actividade sobre o próximo passo a ser executado.

4.5 Conclusão

Neste capítulo foram descritas detalhadamente as implementações do componente de gestão de workflows.

No primeiro sub-capítulo foram abordados aspectos fundamentais relacionados com a notação do modelo de workflow, nomeadamente a definição do fluxo de dados do modelo e do aspecto gráfico das actividades.

Implementação

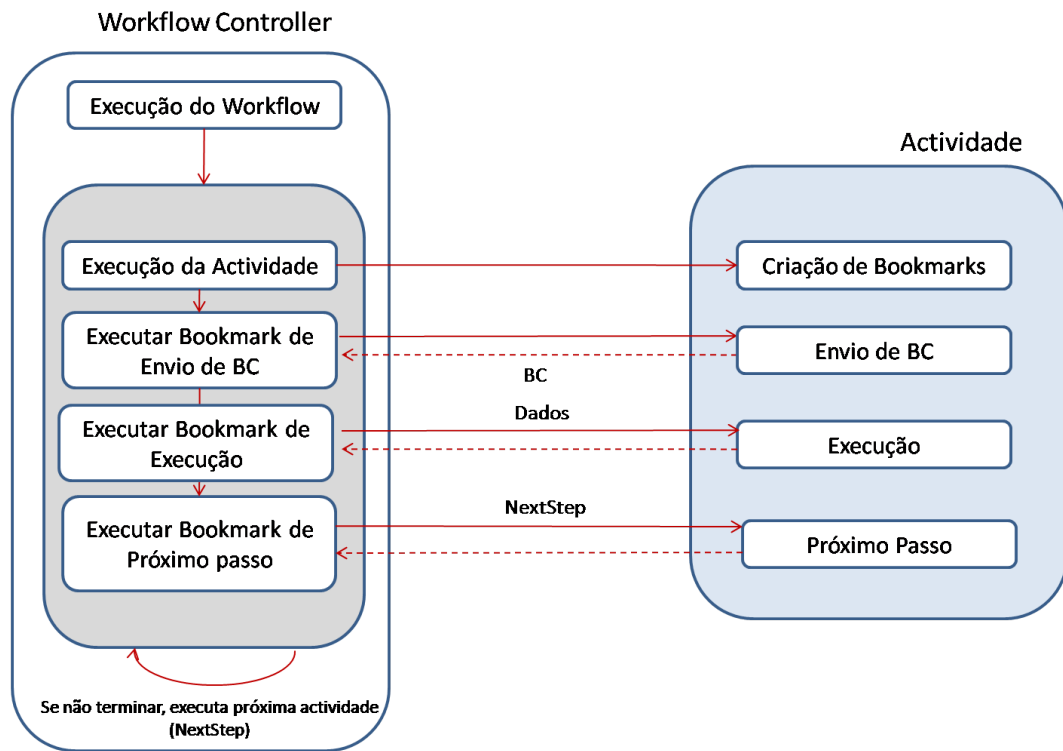


Figura 4.12: Fluxo de execução do workflow

De seguida foi elaborada uma descrição sobre aspectos gráficos da restante implementação, e por último apresentada uma descrição sobre o trabalho da integração com o motor do Windows Workflow 4.0.

Capítulo 5

Conclusões e Trabalho Futuro

Neste capítulo é descrita uma conclusão dos resultados obtidos e a consequente satisfação dos mesmos, no sentido de apresentar uma análise crítica ao trabalho desenvolvido. Também são referidos pontos de melhoria que correspondem a trabalho futuro.

5.1 Satisfação dos Objectivos

A revisão bibliográfica descrita no capítulo 2 permitiu adquirir conhecimentos na área da gestão de modelação de workflows, sendo essenciais no posterior desenvolvimento da solução. Este conhecimento foi decisivo para melhoria do desenho do componente de gestão de workflows, salientando-se a importância deste estudo na apresentação gráfica do modelo e na definição da troca de parâmetros entre actividades/workflows.

Este documento descreveu toda a especificação e a implementação de uma parte substancial do componente de gestão de workflows, incluindo a apresentação gráfica do modelo, a respectiva configuração de parâmetros e a atribuição de argumentos entre actividades/workflows. Porém, os requisitos que estão definidos no capítulo 3 não foram totalmente satisfeitos.

A funcionalidade que permitiria a publicação de workflows, referenciado pelo *user story* US5, não foi implementada devido à falta de tempo de desenvolvimento, o que permitiria uma posterior execução dos workflows em ambiente de execução e a transferência dos mesmos para a barra de ferramentas do WF4. Porém, foram feitos testes que permitiram concluir a correcta execução e reutilização do modelo do workflow. A atribuição de argumentos utilizando as variáveis de contexto e as variáveis dos *Business Components* também não foram implementadas devido ao reduzido tempo de desenvolvimento. Relativamente a estas funcionalidades perspectiva-se a sua implementação numa fase futura.

O resultado desta dissertação permitiu também concluir a possibilidade de modificação do ambiente de modelação do WF4, para se adaptar às necessidades de cada solução.

Durante a fase de desenvolvimento foi ainda adquirido um vasto conhecimento, não só ao nível da experiência de trabalho em ambiente empresarial, mas também a nível da utilização de novas tecnologias e padrões de desenvolvimento (como C#, WF4, WPF e MVVM).

5.2 Discussão

A primeira fase do desenvolvimento desta dissertação resultou num estudo detalhado de identificação da melhor forma de modelar workflows. Apesar da existência de uma quantidade significativa de notações e aplicações orientadas à gestão de workflows, foram descritas apenas algumas delas, sendo considerado suficiente para a aquisição de uma visão geral de conceitos relacionados com o tema.

A intuitividade e facilidade de atribuição de parâmetros entre actividades é um assunto que necessita de maior investigação, por se tratar de uma questão com diferentes pontos-de-vista. Ou seja, o facto de ser intuitivo para alguns utilizadores, pode não ser intuitivo para outros, e vice-versa. Contudo, relativamente à usabilidade primitiva do WF4, conseguiu-se uma adaptação da plataforma às necessidades da solução, utilizando as ideias da equipa de desenvolvimento para a melhoria em função da intuitividade e facilidade de utilização.

Outra questão pertinente tem a ver com o facto do WF4 não disponibilizar soluções baseadas em máquinas de estados, o que não se verifica na antiga versão, já que nesta, é possível redireccionar o fluxo de controlo de cada actividade para várias. A antiga versão facilita a navegação entre workflows, através de um esquema em *wizard*. No entanto, a solução obtida apresenta uma forma de contornar esse problema, através da implementação dos *flowchart* físico e lógico, descrita no capítulo 4.3.5.

Algumas modificações do aspecto gráfico do ambiente de modelação do WF4 não foram totalmente conseguidas (e.g. mudar o *background* do modelo *flowchart*), o que tornou bastante moroso determinar os limites de personalização do WF4.

5.3 Trabalho Futuro

Como a solução continua em fase de desenvolvimento, existem melhorias e incrementos de funcionalidades antes da solução ser disponibilizada para o cliente final. Os pontos de melhoria e trabalho futuro são:

- implementação de funcionalidade de publicação de workflows, para posteriormente serem executados num ambiente de execução da solução geral;
- implementação de um mecanismo que permita a reutilização dos workflows publicados;

- implementação de uma forma de atribuição de variáveis de contexto, uniformizando-as em relação às outras variáveis (argumentos);
- implementação de uma forma de atribuição de variáveis resultantes dos *Business Components*, uniformizando-as em relação às outras variáveis (argumentos);
- integração de nós de decisão na modelação;
- integração com a sessão do utilizador para controlo de acessos;
- investigação de aspectos relacionados com persistência e *tracking* de workflows, para serem utilizados pelo *Workflow Controller*;
- melhorias ao nível da interface gráfica, de modo a permitir uma utilização ainda mais simplificada e intuitiva;

5.4 Conclusão Final

Concluindo, esta dissertação permitiu descrever a implementação de um componente através da reutilização e adaptação de um ambiente de modelação já existente ao âmbito da solução global, e executável no respectivo motor de workflows, em formato *wizard*. Esta adaptação do WF4 tornou possível a simplificação e a modelação de processos de negócio, retirando aspectos que eram desnecessários para a respectiva execução do workflow e para a solução global.

Este componente de gestão de workflows inclui uma nova definição do fluxo de dados do processo de negócio diferente daquela que é disponibilizada pelo WF4. Inclui ainda aspectos que permitem a execução de workflows num formato *wizard*, utilizando *inputs* do utilizador sobre o próximo passo a ser executado.

Juntamente com ambiente de modelação, foram imprescindíveis as implementações do *Designer Runtime Behavior* e da atribuição de *Business Components*. Estas funcionalidades visam oferecer aos utilizadores um conjunto de possibilidades de configuração do modelo do workflow. A implementação do *Object Browser* permitiu ainda facilitar a atribuição de argumentos entre actividades. Este conjunto de funcionalidades, aliada à definição de um fluxo de dados baseada em *pipes and filters*, e à remoção de aspectos de modelação dispensáveis à solução global, ofereceu uma nova forma de modelação e de configuração de processos de negócio.

Referências

- [20111] Microsoft Sharepoint 2010. Microsoft sharepoint 2010, 2011. Disponível em <http://sharepoint.microsoft.com> acessado a última vez em Junho de 2011.
- [Aal97] W.M.P. Van Der Aalst. The application of petri nets to workflow management, Abril 1997. Department of Mathematics and Computing Science.
- [Act11] Website Activiti. Activiti 5.5 user guide, 2011. Disponível em <http://www.activiti.org/userguide/index.html> acessado a última vez em Junho de 2011.
- [AL11] Fernando Correia e Vanessa Silva Angela Lopes. Portal da codificação clínica e dos gdh, 2011. Disponível em <http://posrtalcodgdh.min-saude/index.php/SONHO> acessado a última vez em Junho de 2011.
- [Alf09] Website Alfresco. Alfresco website, 2009. Disponível em <http://www.alfresco.com/> acessado a última vez em Junho de 2011.
- [Alf11] Website Alfresco. Alfresco launches activiti bpmn 2.0 business process engine, 2011. Disponível em http://www.alfresco.com/media/releases/2010/05/activiti_bpm/ acessado a última vez em Junho de 2011.
- [Ang09] Chen Ang. *A Multi-Dimensional Compositional Approach for Business Process Semantic Engineering*. PhD thesis, Université de Genève, 2009.
- [AtH03] W.M.P. Van Der Aalst e A.H.M. ter Hofstede. Business process management: A survey, workflow management systems, 2003.
- [AZ04] Paris Avgeriou e Uwe Zdun. Architectural patterns revisited a pattern language. Technical report, Fraunhofer IPSI, Vienna University of Economics and BA Darmstadt, Germany Vienna, Austria, 2004.
- [Biz11] Website Bizagi, 2011. Disponível em <http://www.bizagi.com/> acessado a última vez em Junho de 2011.
- [BPM11] TIBCO BPM. Tibco bpm website, 2011. Disponível em <http://www.tibco.com/> acessado a última vez em Junho de 2011.
- [Cha07] David Chappel. *The Workflow Way - Understanding Windows Workflow Foundation*. Chappell and Associates, 2007.

REFERÊNCIAS

- [Col10] Mark Collins. *Beginning WF Windows Workflow in .NET 4.0*. Apress, 2010.
- [Col11a] Collabnet. Scrum methodology learn the scrum methodology, 2011. Disponível em <http://scrummethodology.com/> acessado a última vez em Junho de 2011.
- [Col11b] Alfresco Share Collaboration. Getting started with alfresco share collaboration, 2011. Disponível em http://www.alfresco.com/help/32r/enterprise/sharetutorial/Getting_Started_with_Alfresco_Share_Collaboration_for_Enterprise_Edition_3_2_r.pdf acessado a última vez em Maio de 2011.
- [Com11a] JBoss Community. jbpm user guide, 2011. Disponível em <http://docs.jboss.org/jbpm/v5.0/userguide/> acessado a última vez em Junho de 2011.
- [Com11b] JBoss Community. jbpm website, 2011. Disponível em <http://www.jboss.org/jbpm> acessado a última vez em Junho de 2011.
- [DH01] M. Dumas e A. HofstedeJ. Uml activity diagrams as a workflow specification language. Technical report, In Proceedings of the UML2001 Conference, 2001.
- [DP06] Ana Atayde Eduardo Borges Rodolfo Resende e Clarindo Pádua Daniela Peixoto, Vitor Batista. A comparison of bpmn and uml 2.0 activity diagrams. Technical report, Departamento de Ciência da Computação Universidade Federal de Minas Gerais, 2006.
- [eAMB05] Aurora Amélia Castro Teixeira e Ana Margarida Brochado. Quando o sonho se torna realidade... avaliação estatística do impacto das tecnologias de informação nos serviços de consulta externa hospitalar, 2005.
- [Esh02] Hendrik Eshuim. Sementics and verification of uml activity diagrams for workflow modeling, 2002.
- [FCYA10] Hugo Sereno Ferreira, Filipe Figueiredo Correia, Joseph Yoder e Ademar Aguiar. Core patterns of object-oriented meta-architectures. Technical report, Proceedings of the 17th Conference on Pattern Languages of Programs, 2010.
- [Fer11] Hugo Sereno Ferreira. *Adaptive Object-Modeling: Patterns, Tools and Applications*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, 2011.
- [Flo11a] FlowBreeze. What is a flowchart?, 2011. Disponível em <http://www.breezetre.com/articles/what-is-a-flow-chart.htm> acessado a última vez em Junho de 2011.
- [Flo11b] RFFlow 5 Professional Flowcharting. What do the different flowchart shapes mean?, 2011. Disponível em http://www.rff.com/flowchart_shapes.htm acessado a última vez em Junho de 2011.

REFERÊNCIAS

- [GHS95] D. Georgakopoulos, M. Hornick e A. Sheth. An overview of workflow management from process modeling to workflow automation infrastructure. Technical report, Kluwer Academic Publishers, Boston, 1995.
- [GJT07] R. Garud, S. Jain e P. Tuertscher. Incomplete by design and designing for incompleteness, Fevereiro 2007.
- [Gro11a] Object Management Group. Business process model and notation (bpmn) specification version 1.0, 2011. Disponível em http://www.omg.org/bpmn/Documents/BPMN_1-1_Specification.pdf acessado a última vez em Junho de 2011.
- [Gro11b] Object Management Group. Object management group/business process management initiative, 2011. Disponível em <http://www.bpmn.org/> acessado a última vez em Junho de 2011.
- [Hol95] David Hollingsworth. Workflow management coalition: The workflow reference model, Janeiro 1995.
- [IBM11] IBM. Ibm software websphere, 2011. Disponível em <http://www-01.ibm.com/software/websphere/> acessado a última vez em Junho de 2011.
- [LE06] Eero Kallio e Ilkka Terh Lauri Eloranta. A notation evaluation of bpmn and uml activity diagram. Technical report, Helsinki University of Technology, 2006.
- [MD11] Dustin Metzgar e Wenlong Dong. Windows workflow foundation 4 performance, 2011. Disponível em <http://msdn.microsoft.com/en-us/library/gg281645.aspx> acessado a última vez em Junho de 2011.
- [Mil11] Matt Milner. A developers introduction to windows workflow foundation (wf) in .net 4, 2011. Disponível em <http://msdn.microsoft.com/en-us/library/ee342461.aspx> acessado a última vez em Junho de 2011.
- [Mod11] Agile Modeling. Agile modeling - uml 2 activity diagrams, 2011. Disponível em <http://www.agilemodeling.com/artifacts/activityDiagram.htm> acessado a última vez em Junho de 2011.
- [Off11] Microsoft Office. Funcionalidades e vantagens do visio 2010, 2011. Disponível em <http://office.microsoft.com/pt-pt/visio/funcionalidades-e-vantagens-do-visio-2010-HA101631752.aspx> acessado a última vez em Junho de 2011.
- [Pot10] Jeff Potts. Alfresco developer guide, Janeiro 2010.
- [RIRG06] J. Recker, M. Indulska, M. Rosemann e P. Green. How good is bpmn really? insights from theory and practice, 2006.
- [Rog00] Simon Rogerson. Electronic patient records. Technical report, ETHicol IMIS Journal, 2000.

REFERÊNCIAS

- [SA11] Siemens SA. Siemens sa website, 2011. Disponível em <http://www.siemens.com> acessido a última vez em Junho de 2011.
- [Sof11] Mountain Goat Software. Introduction to scrum, an agile process, 2011. Disponível em <http://www.mountaingoatsoftware.com/topics/scrum> acessido a última vez em Junho de 2011.
- [SZ05] M. Slot e P. V. Zoolingen. Workflow management systems. Technical report, Division of Mathematics and Computer Science, Vrije Universiteit, The Nerherlands, 2005.
- [Tec11] IBM Data Processing Techniques. Flowcharting techniques, 2011. Disponível em <http://www.fh-jena.de/~kleine/history/software/IBM-FlowchartingTechniques-GC20-8152-1.pdf> acessido a última vez em Junho de 2011.
- [Tip11] Tipskey. Advanced flowchart, 2011. Disponível em http://www.tipskey.com/manufacturing/advanced_flowchart.htm acessido a última vez em Junho de 2011.
- [Von07] Ivo Vondrak. Business process modeling. Technical report, Dept. of Computer Science Faculty of Electrical Engeneering and Computer Science Technical University of Ostrava, Czech Republic, 2007.