



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Departamento de Engenharia de Electrónica e
Telecomunicações e de Computadores

Sistemas de Informação

Gestão Integrada de Sistemas de Execução de Processos WS-BPEL

Hugo Alexandre Tavares Marçal
(Licenciado)

Trabalho de projecto realizado para obtenção do grau de Mestre
em Engenharia Informática e de Computadores

Orientador:

Professor Coordenador Licenciado António Luís Freixo Guedes Osório, ISEL

Júri:

Presidente: Professor Coordenador Doutor Manuel Martins Barata, ISEL

Vogal: Professor Adjunto Mestre Luís Manuel da Costa Assunção, ISEL

Vogal: Professor Coordenador Licenciado António Luís Freixo Guedes
Osório, ISEL

Novembro de 2010

Resumo

O trabalho de projecto consiste numa proposta de solução para gestão integrada de sistemas de execução de processos de negócio.

Os processos de negócio definem um quadro de realização de tarefas numa organização. Perante a constante evolução dos parâmetros do negócio, as organizações necessitam de adoptar metodologias que permitam ter mais controlo sobre a definição e execução dos seus processos, contribuindo para a sua automatização. Paralelamente a esta evolução, surgem iniciativas de linguagens para definição de processos e sistemas de execução provenientes de diversos fornecedores. Todos eles apresentam funcionalidades que permitem o acesso à informação de gestão dos processos, mas, na sua generalidade, através de aplicações ou interfaces programáticas proprietárias. Esta abordagem constitui um problema de dependência das aplicações de gestão de processos relativamente à especificidade da gestão em diversos sistemas de execução de uma organização.

O desenvolvimento deste projecto, com a elaboração de um protótipo demonstrador, visa a validação de um modelo uniforme para representação e acesso à informação de gestão dos processos de negócio. A estratégia elaborada segue uma abordagem de gestão integrada que permite unificar a gestão de processos em diversos sistemas de execução, usufruindo de outros modelos de referência para sustentar a proposta apresentada.

Palavras-chave:

- Gestão de processos de negócio;
- Processo WS-BPEL;
- Sistema de execução WS-BPEL;
- Gestão integrada de processos WS-BPEL.

Abstract

The project is a proposal for an integrated management solution for systems running business processes.

The business processes define a set of performing tasks in an organization. Given the constant evolution of business parameters, organizations need to adopt methodologies that allow more control over the definition and implementation of processes, contributing to its automation. Alongside with this evolution, there are initiatives for processes definition languages and execution systems from various suppliers. They all have features that allow access to business process management information, but, in general, through proprietary application programming interfaces. This approach causes a problem of dependency between management applications and the specific form of access process management information in different execution engines inside of an organization.

The development of this project, alongside with the development of a prototype demonstrator, aims to validate a uniform model of representation and access to the business processes management information. The strategy drawn following an integrated management approach allows unifying process management in various execution systems, taking advantage of other role models to support the proposal presented.

Keywords:

- *Business Process Management;*
- *WS-BPEL Process;*
- *WS-BPEL Execution System;*
- *Integrated management of WS-BPEL processes.*

Agradecimentos

Quero registar o agradecimento aos demais intervenientes neste projecto que, de forma directa ou indirecta, representaram para mim um símbolo de força e ânimo para conseguir atingir este objectivo.

Ao orientador do projecto, por toda a disponibilidade demonstrada e por todo o apoio prestado.

À minha família, pelo apoio constante e esforço empregue para que pudesse concluir o curso.

À Carla, minha namorada, por toda a compreensão nos momentos mais difíceis e pelo empenho e coragem transmitidos durante toda a realização do projecto.

Ao Rui Freitas, Nuno Maurício e Alexandre Cavaco, meus colegas de curso do ISEL, que sempre se revelaram incansáveis amigos e virtuosos encorajadores, transmitindo sempre mensagens de apoio para que todo o projecto fosse realizado da melhor forma possível.

Aos restantes colegas de curso, com quem me cruzei ao longo destes anos, que contribuíram também, sem dúvida, para a minha formação académica e pessoal, sendo peças fundamentais na tentativa de alcançar o grande objectivo de terminar o mestrado.

A todos, o meu muito obrigado.

Glossário de termos

WS-BPEL – *Web Services Business Process Execution Language*

Linguagem, baseada em XML, para definição e execução de processos de negócio, especificando interacções entre *Web Services*.

XML – *Extensible Markup Language*

Formato para criação de documentos organizados de forma hierárquica.

XSD – *XML Schema*

Formato que descreve a estrutura de um documento XML.

XPath – *XML Path Language*

Linguagem de navegação entre elementos e atributos de um documento XML.

Web Service

Serviço que permite a comunicação uniforme entre aplicações, através de normas como XML, WSDL e SOAP sobre internet.

Sistema

Conjunto de funcionalidades que permitem concretizar determinados objectivos para um utilizador (pessoa ou sistema).

Aplicação

Definição de sistema, com inclusão de interface (gráfica) para utilizador.

Serviço

Componente desenvolvido para concretizar funcionalidades específicas no âmbito de um sistema e promover a capacidade de reutilização em aplicações.

Recurso

Entidade lógica abstracta, identificável, constituída por propriedades e utilizada para manutenção de estado.

Tabela de conteúdos

1. Introdução	1
1.1. Motivação.....	1
1.2. Hipóteses consideradas	2
1.3. Metodologia e objectivos propostos	4
1.4. Estratégia adoptada.....	5
1.5. Organização do documento	7
2. Enquadramento	9
2.1. A integração numa estrutura organizacional.....	9
2.2. Gestão de processos de negócio	11
2.2.1. Definição de processos em WS-BPEL	14
2.2.2. Outras iniciativas de referência	19
2.3. Execução de processos WS-BPEL	23
2.4. Monitorização de processos WS-BPEL	24
2.5. Tecnologias de gestão baseadas em recursos.....	25
2.6. Síntese.....	30
3. Arquitectura proposta	31
3.1. Hipóteses consideradas	32
3.2. Objectivos propostos	37
3.3. Análise de requisitos.....	38
3.4. Arquitectura da solução.....	39
3.5. Estratégia de solução	41
3.5.1. Representação de processos WS-BPEL através de recursos.....	41
3.5.2. Actualização das propriedades dos recursos	46
3.5.3. Interface de acesso aos recursos.....	47
3.5.4. Arquitectura detalhada.....	49
3.6. Síntese.....	51
4. Implementação	53
4.1. Requisitos de implementação	53
4.1.1. Apache ODE - Sistema de execução de processos WS-BPEL	54
4.2. Protótipo demonstrador	57
4.2.1. Sistema Gerador de Recursos (SGR)	57
4.2.2. Sistema de Actualização de Recursos (SAR)	68
4.2.3. Serviço de Gestão (SG).....	75
4.2.4. Aplicação de Gestão Integrada (AGI).....	87
4.3. Considerações sobre demonstrador	91
4.4. Síntese.....	93
5. Demonstração de resultados	95

5.1. Definição de processo WS-BPEL.....	95
5.2. Representação de processo em recurso	98
5.3. Gestão/monitorização da execução do processo	99
6. Conclusão.....	101
6.1. Trabalho futuro	104
7. Referências.....	105
8. Apêndices.....	111
8.1. Apêndice A – Ficheiro de definição de um recurso	111
8.2. Apêndice B – Ficheiro de configuração do Sistema Gerador de Recursos	114
8.3. Apêndice C – Exemplo de ficheiro recurso (original).....	115
8.4. Apêndice D – Exemplo de ficheiro recurso (após execução).....	117
8.5. Apêndice E – Ficheiro de configuração do Sistema de Actualização de Recursos	119
8.6. Apêndice F – Ficheiro WS-BPEL do processo de negócio <i>Loan Request</i>	120
8.7. Apêndice G – Casos de utilização do Sistema Gerador de Recursos	125
8.8. Apêndice H – Casos de utilização da interface de gestão.....	128

Lista de figuras

Figura 1 – Motivação do projecto	2
Figura 2 – Hipóteses de solução	3
Figura 3 - Resumo da solução proposta para gestão integrada	6
Figura 4 – Conceito de <i>Bus Aberto</i> de Serviços	10
Figura 5 – Definição de processo	11
Figura 6 – Ciclo de vida da gestão de um processo de negócio	13
Figura 7 – Orquestração de serviços	15
Figura 8 – Funcionalidade do elemento <i>partnerLink</i> no WS-BPEL [13]	16
Figura 9 – Processo WS-BPEL e interacções com Web Services (<i>partner links</i>) [5] ...	16
Figura 10 – Uniformização da automatização de processos com WS-BPEL [14]	17
Figura 11 - WS-BPEL no stack WS-* [4]	17
Figura 12 – Arquitectura de um sistema de <i>workflow</i> [15]	20
Figura 13 – <i>Workflow</i> na perspectiva do <i>Windows Workflow Foundation</i> [21]	21
Figura 14 – Arquitectura JMX [25]	26
Figura 15 – Conceito de recurso segundo norma WS-Resource [10]	28
Figura 16 - Arquitectura proposta em "A Management Framework for WS-BPEL" [1].	31
Figura 17 - Hipótese de solução com uniformização dos sistemas de execução	33
Figura 18 – Hipótese de solução com integração na aplicação de gestão	34
Figura 19 – Arquitectura da solução proposta	39
Figura 20 – Diagrama de contexto da solução	41
Figura 21 – Geração de recurso para representar um processo WS-BPEL	43
Figura 22 - Processo <i>HelloWorld</i> e representação em recurso [1]	45
Figura 23 – Diagrama de contexto para actualização de recuso	46
Figura 24 – Manutenção de estado em recursos [9]	47
Figura 25 – Relação entre <i>Web Service</i> e Recurso	48
Figura 26 – Arquitectura detalhada da solução	49
Figura 27 – Arquitectura do orquestrador Apache ODE [30]	55
Figura 28 – Sistema Gerador de Recursos na arquitectura proposta	57
Figura 29 – Interface gráfica do Sistema Gerador de Recursos	58
Figura 30 – Diagrama de casos de utilização do Sistema Gerador de Recursos	58
Figura 31 – Diagrama de comunicação do sistema SGR	59
Figura 32 – Diagrama de sequência do caso de utilização <i>Iniciar geração</i>	59
Figura 33 – Diagrama de sequência do caso de utilização <i>Parar geração</i>	60
Figura 34 – Diagrama de sequência do caso de utilização <i>Gerar recurso</i>	61
Figura 35 – Diagrama de sequência do caso de utilização <i>Terminar sistema</i>	61

Figura 36 – Diagrama de classes do Sistema Gerador de Recursos	62
Figura 37 – Diagrama de contexto da geração de um recurso.....	63
Figura 38 – Modelo de dados dos elementos XML do <i>Schema</i> do recurso.....	64
Figura 39 – Definição do elemento XML <i>bpelVariable</i>	65
Figura 40 – Definição do elemento XML <i>bpelInstanceInfo</i>	66
Figura 41 – Diagrama de contexto da geração dinâmica de recursos.....	67
Figura 42 – Sistema de Actualização de Recursos na arquitectura proposta.....	68
Figura 43 – Diagrama de contexto do Sistema de Actualização de Recursos.....	68
Figura 44 – Diagrama de estrutura composta do SAR.....	70
Figura 45 – Interface <i>BpelEventListener</i> do sistema Apache ODE [31].....	71
Figura 46 – Diagrama de sequência da criação/actualização de recursos no SAR.....	71
Figura 47 – Diagrama de classes do sistema SAR	72
Figura 48 – Sistema de Gestão na arquitectura proposta	75
Figura 49 – Diagrama de casos de utilização da interface de gestão.....	76
Figura 50 – Interface programática do Serviço de Gestão	77
Figura 51 – Modelo de análise do caso de utilização <i>Listar sistemas de execução</i>	78
Figura 52 – Modelo de análise do caso de utilização <i>Listar processos</i>	78
Figura 53 – Modelo de análise do caso de utilização <i>Listar instâncias</i>	78
Figura 54 – Modelo de análise do caso <i>Obter documento de propriedades</i>	79
Figura 55 – Modelo de análise do caso de utilização <i>Obter propriedade</i>	79
Figura 56 – Modelo de análise dos casos <i>Prosseguir/Suspender/Terminar instância</i> .	79
Figura 57 – Diagrama de seq. do Serviço de Gestão (interacção com Apache ODE) .	80
Figura 58 – Diagrama de seq. do Serviço de Gestão (interacção com recursos).....	81
Figura 59 – Diagrama de estrutura composta do Serviço de Gestão	82
Figura 60 – Diagrama de classes do Serviço de Gestão.....	82
Figura 61 – Classe <i>ApacheOdeManagement</i>	83
Figura 62 – Definição das classes <i>ProcessInfo</i> e <i>InstanceInfo</i>	85
Figura 63 – Aplicação de Gestão na arquitectura proposta.....	87
Figura 64 – Diagrama de casos de utilização da aplicação de gestão	88
Figura 65 – Interface da aplicação de gestão/monitorização.....	89
Figura 66 – Diagrama de estrutura composta da Aplicação de Gestão.....	89
Figura 67 – Processo WS-BPEL <i>Loan Request</i>	96
Figura 68 – Monitorização do processo WS-BPEL <i>Loan Request</i>	99
Figura 69 – Diagrama de casos de utilização do Sistema Gerador de Recursos	125
Figura 70 – Diagrama de casos de utilização da interface de gestão.....	128

Lista de tabelas

Tabela 1 - Convenção de formatação do relatório	7
Tabela 2 – Soluções de sistemas de execução WS-BPEL	23
Tabela 3 - Comparação das abordagens propostas	36
Tabela 4 – Descrição dos actores do sistema.....	41
Tabela 5 - Propriedades de um recurso e valores possíveis	44
Tabela 6 - Valores das propriedades dos recursos [1]	64
Tabela 7 – Associação dos elementos WS-BPEL a propriedades de recurso.....	65
Tabela 8 - Eventos de execução suportados pelo Apache ODE [29]	69
Tabela 9 – Descrição do caso de utilização <i>Iniciar geração</i> (SGR).....	126
Tabela 10 – Descrição do caso de utilização <i>Parar geração</i>	126
Tabela 11 – Descrição do caso de utilização <i>Gerar recurso</i>	127
Tabela 12 – Descrição do caso de utilização <i>Terminar sistema</i>	127
Tabela 13 – Descrição do caso de utilização <i>Listar sistemas de execução</i>	129
Tabela 14 – Descrição do caso de utilização <i>Listar processos</i>	129
Tabela 15 – Descrição do caso de utilização <i>Listar instâncias</i>	130
Tabela 16 – Descrição do caso de utilização <i>Obter documento de propriedades</i>	131
Tabela 17 – Descrição do caso de utilização <i>Obter propriedade</i>	131
Tabela 18 – Descrição do caso de utilização <i>Suspender instância</i>	132
Tabela 19 – Descrição do caso de utilização <i>Prosseguir instância</i>	133
Tabela 20 – Descrição do caso de utilização <i>Terminar instância</i>	133
Tabela 21 – Descrição do caso de utilização <i>Seleccionar processo</i>	134
Tabela 22 – Descrição do caso de utilização <i>Seleccionar instância</i>	134

Lista de listagens

Listagem 1 – Configuração e adição de receptor de eventos no Apache ODE	70
Listagem 2 – Exemplo de retorno do método <i>GetInstanceInfo</i> do <i>Web Service InstanceManagement</i> (Apache ODE)	73
Listagem 3 – Expressões XPath para obter propriedades do recurso a actualizar	74
Listagem 4 – Assinatura do método <i>ListOrchestrators</i>	83
Listagem 5 – Assinatura do método <i>ListProcesses</i>	83
Listagem 6 – Exemplo de resposta do método <i>ListProcesses</i>	84
Listagem 7 – Assinatura do método <i>ListInstances</i>	84
Listagem 8 - Exemplo de resposta do método <i>ListInstances</i>	85
Listagem 9 – Assinatura do método <i>GetResourcePropertyDocument</i>	86
Listagem 10 – Assinatura do método <i>GetResourceProperty</i>	86
Listagem 11 – Assinatura dos métodos <i>Suspend</i> , <i>Resume</i> e <i>Terminate Instance</i>	86
Listagem 12 – Configuração para inserir o processo <i>Loan Request</i> no Apache ODE .	97
Listagem 13 - Propriedade <i>bpelAssign</i> original	98
Listagem 14 - Propriedade <i>bpelAssign</i> após actualização	98
Listagem 15 - Criação de instância do processo <i>Loan Request</i>	99
Listagem 16 – Ficheiro de definição de um recurso (<i>BpelProcessRepresentation</i>)...	113
Listagem 17 – Ficheiro de configuração do Sistema Gerador de Recursos	114
Listagem 18 – Ficheiro recurso do processo WS-BPEL <i>LoanRequest</i>	116
Listagem 19 – Ficheiro recurso do processo WS-BPEL <i>Loan Request</i> (após actualização)	118
Listagem 20 – Ficheiro de configuração do Sistema de Actualização de Recursos ..	119
Listagem 21 – Ficheiro WS-BPEL de definição do processo <i>Loan Request</i>	124

1. Introdução

Este capítulo tem por objectivo enquadrar o projecto, apresentando a sua motivação, hipóteses equacionadas, metodologia adoptada, resumo da solução proposta, conhecimentos necessários e organização do documento para a compreensão do trabalho realizado.

1.1. Motivação

Os processos de negócio definem um quadro de realização de tarefas de uma organização. Num contexto de constante evolução, as organizações necessitam de adoptar novos modelos de gestão de processos que lhes permitam ter mais controlo sobre a sua definição e execução [2], [16], [51].

A evolução tecnológica dos sistemas de gestão de processos de negócio permite às organizações utilizarem cada vez mais este tipo de sistemas, para definir e executar processos como estratégia para a sua automatização. Por sua vez, o mercado dos sistemas de integração cresceu de forma significativa com o envolvimento de soluções proprietárias e *open source*, provenientes de diversos fornecedores [16], [45].

A OASIS (*Organization for the Advancement of Structured Information Standards*) [8], através da especificação WS-BPEL [3], surge como referência na definição de um processo de negócio através da composição de serviços (*Web Services*). Esta definição adequa-se a um modelo de execução composto por um orquestrador (coordenador) que gere a execução do processo. Existem diversos sistemas de execução que seguem este modelo e são compatíveis com a especificação WS-BPEL, constituindo-se como soluções de gestão de processos que podem ser adoptadas numa organização [1], [5].

No entanto, constata-se que os sistemas de execução disponibilizam o acesso à informação de gestão/monitorização dos processos através de aplicações ou interfaces programáticas proprietárias, apesar dos esforços das comunidades e iniciativas de referência, não apenas da OASIS, mas também de outras como o WfMC (*Workflow Management Coalition*) [15] e o OMG (*Object Management Group*) [18], em desenvolver normas que sustentem a automatização de processos e o conceito de BPM (*Business Process Management*) [1]. Esta abordagem, adoptada pelos diversos fornecedores, estabelece um quadro de dependência das aplicações de gestão de processos relativamente aos sistemas de execução.

A inexistência de um modelo uniforme, para representação e acesso à informação de gestão dos processos, compromete a utilização de um sistema capaz de unificar a gestão/monitorização de processos em diversos sistemas de execução [1]. A Figura 1 ilustra o cenário descrito.

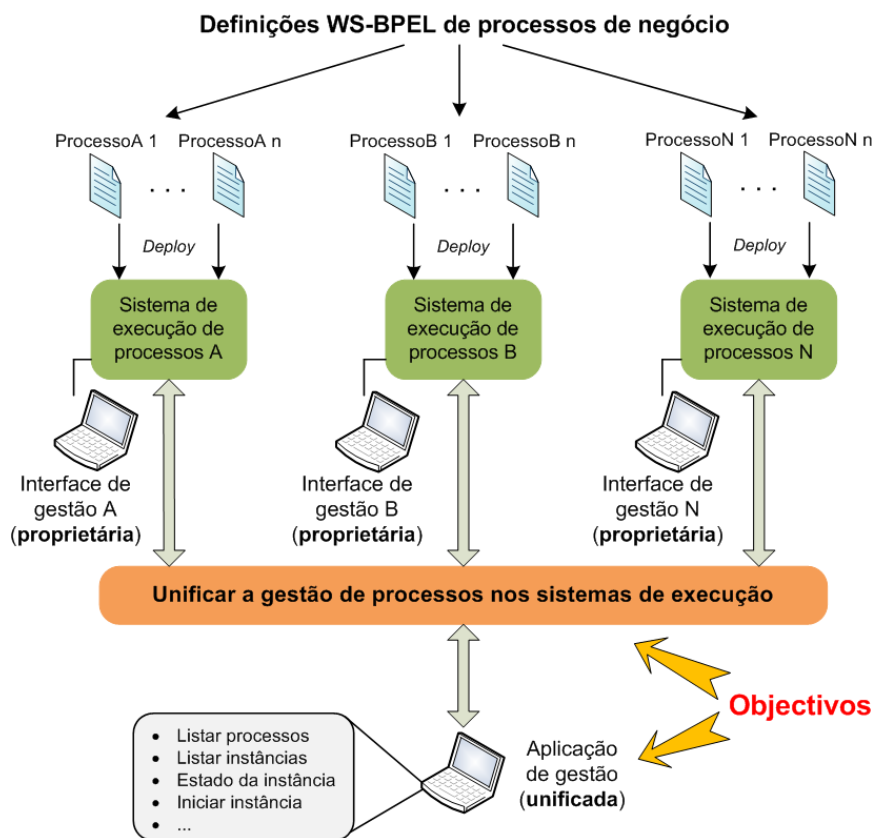


Figura 1 – Motivação do projecto

Assim, a motivação deste projecto baseia-se na problemática da dependência identificada nos actuais sistemas de execução de processos e o seu enquadramento numa organização. É necessária uma solução que unifique a gestão de processos, em diversos sistemas de execução, concretizada numa aplicação de gestão integrada.

1.2. Hipóteses consideradas

Perante a motivação enunciada, a solução a adoptar deve promover a redução de sistemas isolados e a adopção de lógica cooperativa e aberta de integração de sistemas. Por lógica “cooperativa” entende-se a cooperação de entidades (pessoas /sistemas) através da partilha de informação ou tarefas, no sentido de realizar um objectivo¹. Por lógica “aberta” entende-se a adopção de normas não proprietárias.

¹ S. Butenko, R. Murphey, P. M. Pardalos, Cooperative Control: Models, Applications and Algorithms, ISBN 978-1402010828, 31-01-2003

Foram identificadas duas hipóteses para a gestão integrada, sendo que uma delas foi a seleccionada para a concretização do projecto. As duas abordagens consideram a implementação de adaptadores que permitem efectuar a integração entre os sistemas de execução WS-BPEL e a aplicação de gestão de processos.

Uma das abordagens contempla a criação de adaptadores nos sistemas de execução, com o objectivo de realizar a transformação dos seus dados internos para um formato neutro de integração com a aplicação de gestão. Esta solução considera um modelo que minimiza a dependência entre os sistemas envolvidos (fraco acoplamento), promovendo flexibilidade, escalabilidade e adaptabilidade, relativamente à incorporação de novos sistemas de execução e gestão [2], [47].

A abordagem alternativa considera a realização dos adaptadores na aplicação de gestão integrada, sendo esta a responsável por conter a lógica de integração com cada sistema de execução, permitindo manter estes últimos inalterados. Esta solução apresenta um modelo de maior acoplamento (dependência) entre sistemas de execução e aplicação de gestão, sendo menos flexível, adaptável e escalável do que a solução anterior.

A Figura 2 agrega as hipóteses consideradas, ilustrando, para cada uma, a localização dos referidos adaptadores que contêm a lógica de integração entre sistemas.

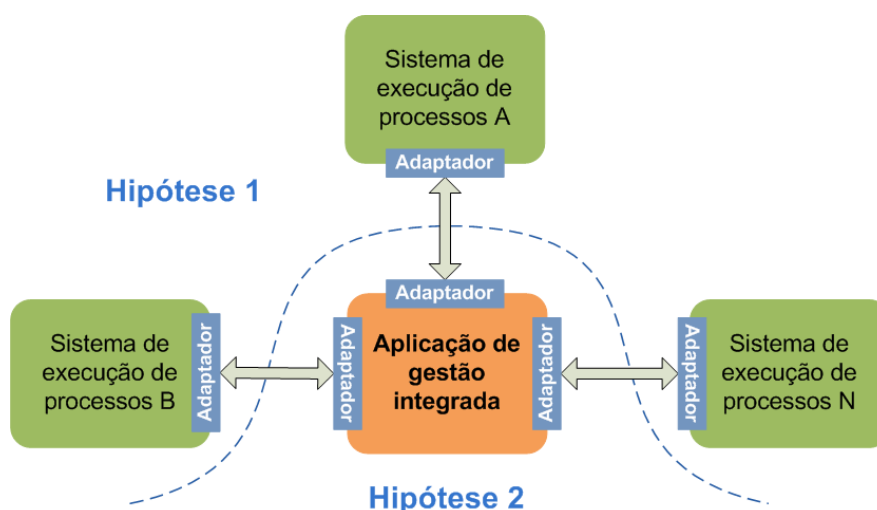


Figura 2 – Hipóteses de solução

Apesar das desvantagens enunciadas para a abordagem alternativa, esta foi a solução considerada para a implementação da solução de gestão integrada de sistemas de execução de processos WS-BPEL, face à primeira hipótese, devido ao actual estado da arte das tecnologias, metodologias e sistemas envolvidos na execução e gestão de processos de negócio. É uma solução conservadora, com maior aceitabilidade e

enquadramento na cultura tradicional de integração praticada nas organizações, o que reduz o investimento (tempo e custos) na alteração dos sistemas de execução já implementados e sobre os quais os gestores do negócio confiam. Aliado a estas vantagens, é uma solução exequível numa metodologia de integração de sistemas, constituída por sucessivas camadas de integração, desenvolvidas à medida das necessidades [2].

1.3. Metodologia e objectivos propostos

A elaboração deste projecto engloba um estudo sobre a automatização dos processos de negócio, quando considerada a sua definição, execução e gestão/monitorização.

Os objectivos propostos incluem o estudo de um sistema de execução *open source*, para execução de processos, com a respectiva interface programática de gestão/monitorização proprietária.

A criação de um modelo uniforme para gerar representações dos processos existentes nos sistemas de execução, baseadas na sua definição e com suporte a informação de gestão/monitorização, considera-se também objecto de estudo e desenvolvimento. Cada processo existente no sistema de execução tem a sua representação, que visa ser genérica e independente do modelo de dados interno do seu executor. Por cada instância de um processo é criada uma cópia da sua representação para ser actualizada dinamicamente, durante a execução, em função das alterações ocorridas no estado dos elementos constituintes do processo [1].

O objectivo da representação de cada processo, segundo o modelo proposto, é disponibilizar a sua informação de gestão/monitorização de cada processo a uma aplicação de gestão integrada. Num cenário ideal, a referida representação uniforme possibilita a gestão integrada, resolvendo a dependência relativamente às interfaces de gestão específicas de cada sistema de execução [1].

Propõe-se a validação dos conceitos e da estratégia de solução através do desenvolvimento de um conjunto de sistemas, que suportem os objectivos anteriormente descritos, de entre os quais se inclui uma aplicação de gestão integrada. Estes sistemas compõem o protótipo demonstrador da solução, permitindo efectuar a gestão/monitorização de processos no sistema de execução estudado.

1.4. Estratégia adoptada

As actuais linguagens de definição de processos de negócio permitem delinear o conjunto de tarefas necessário para executar o processo. No entanto, por si só, a definição de um processo WS-BPEL não contempla a informação de gestão/monitorização das suas instâncias. Esta informação adicional necessita de ser armazenada para posteriormente ser actualizada, na sequência da execução das instâncias.

A solução adoptada apresenta um modelo de gestão integrado através da representação da definição de processos na forma de recursos. Cada recurso é uma entidade lógica abstracta que representa uma instância de um processo em execução e é actualizada em função dos eventos de execução suportados pelo sistema onde é executada. Um recurso é constituído por propriedades, definidas segundo um modelo de dados que permitem armazenar a informação de gestão/monitorização da instância que representa [1], [6].

A actualização das propriedades dos recursos é efectuada através da utilização de uma interface programática de definição dos eventos do sistema de execução, disponibilizada pelo seu fabricante. A estratégia baseia-se num padrão de notificação dos eventos de execução, ocorridos sobre a instância, mais concretamente, sobre alterações no estado dos seus elementos constituintes. As alterações são propagadas para as propriedades do recurso associado, mantendo-o actualizado ao longo da execução do respectivo processo [1].

A informação presente nos recursos é disponibilizada à aplicação de gestão/monitorização através de uma interface programática implementada por um serviço de gestão que permite o acesso aos recursos e às suas propriedades [1]. A aplicação de gestão utiliza a respectiva interface do serviço mencionado para aceder à informação dos recursos, representativos das instâncias dos processos em execução nos diversos sistemas de execução.

A Figura 3 ilustra a solução descrita.

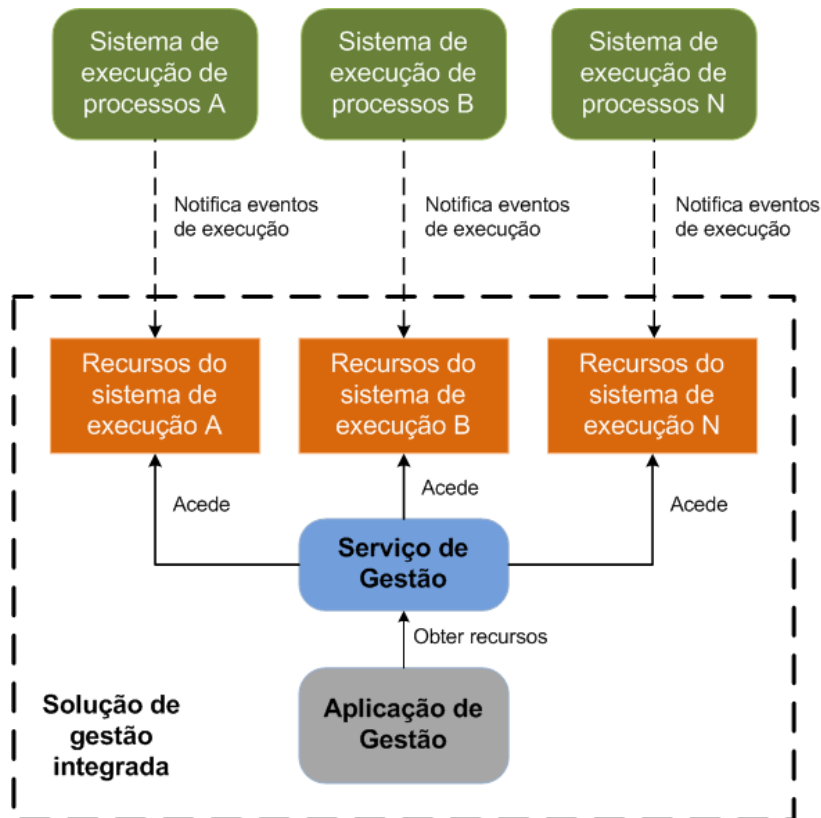


Figura 3 - Resumo da solução proposta para gestão integrada

Conhecimentos necessários

Os requisitos mínimos necessários para a compreensão deste documento assentam em noções de arquitectura orientada a serviços, programação orientada a objectos, desenvolvimento de processos de negócio na linguagem WS-BPEL, experiência em XML, XSD, XPath e *Web Services*.

1.5. Organização do documento

O documento encontra-se estruturado de forma a introduzir o tema da dissertação, o seu enquadramento na actualidade, a estratégia de solução adoptada e a sua implementação, conclusões e trabalho futuro. O documento está estruturado segundo a seguinte descrição.

O segundo capítulo, designado por **Enquadramento**, apresenta o estado actual relativamente a metodologias e tecnologias de automatização, definição, execução e monitorização de processos de negócio.

O terceiro capítulo, designado por **Arquitectura proposta**, apresenta a discussão das hipóteses de solução consideradas, bem como os objectivos, requisitos e a arquitectura da solução para a hipótese colocada.

O quarto capítulo, designado por **Implementação**, apresenta a estratégia de desenvolvimento adoptada para a arquitectura proposta e os aspectos de implementação do protótipo demonstrador com os sistemas constituintes da solução.

O quinto capítulo, designado por **Demonstração**, apresenta a validação da solução proposta com uma demonstração de resultados obtidos através de testes sobre o protótipo demonstrador.

O sexto capítulo, designado por **Conclusão**, apresenta as conclusões resultantes do desenvolvimento do projecto, bem como a sua análise crítica e algumas considerações para melhoramentos e opções de trabalho futuro.

Convenções de formatação

A tabela seguinte apresenta as formatações utilizadas para a escrita deste documento.

Convenção	Utilização
Arial	Texto, tabelas e legendas
<i>Arial</i>	Estrangeirismos
Arial	Nome dos capítulos
Courier-New	Listagem de código e ficheiros

Tabela 1 - Convenção de formatação do relatório

2. Enquadramento

Este capítulo descreve o enquadramento do projecto, nomeadamente o estado actual de tecnologias, metodologias e sistemas para automatização de processos de negócio através da sua definição, execução e gestão/monitorização.

2.1. A integração numa estrutura organizacional

Uma organização possui diversos sistemas de informação de suporte ao seu negócio. De entre estes podem considerar-se sistemas de ERP (*Enterprise Resource Planning*), CRM (*Customer Relationship Management*) ou SCM (*Supply Chain Management*), desenvolvidos numa lógica proprietária por diferentes fornecedores. Como forma de realização dos processos de negócio são gradualmente adaptados para se integrarem na organização e cooperarem entre si [45]. No contexto de uma organização, vista como um todo, o seu portfolio de *software* contém aplicações fechadas, isoladas, em plataformas heterogéneas, que tendem a possuir duplicação de dados e serviços, originando “ilhas de automação” [2]. Estas, por sua vez, comunicam entre si, recorrendo a sucessivas camadas de integração desenvolvidas à medida das necessidades, com mecanismos de envio de mensagens, ficheiros ou ainda através de interacção humana [46]. A lacuna de um modelo de integração entre sistemas, de forma aberta e cooperativa, causa uma diversidade de problemas que dificultam a automatização dos processos de negócio da organização [2],[45].

Neste quadro, um executor de processos WS-BPEL é visto como um sistema que deve ser integrado segundo um modelo baseado em serviços abertos, através de interfaces não proprietárias, para disponibilizar informação de gestão dos processos que executa aos demais sistemas receptivos a essa informação [1].

Embora a sua definição não seja consensual, um *bus* aberto de serviços é uma proposta de um modelo que ambiciona integrar um conjunto de serviços para que sejam utilizados/reutilizados pelos sistemas da organização. Nesta lógica, cada sistema é visto como um serviço que realiza determinadas funcionalidades e disponibiliza interfaces abertas (não proprietárias). Através destas características, o sistema/serviço é incluído, idealmente de forma *plug-and-play*, neste “backbone virtual” de partilha de informação e integração [2]. Os projectos *open-source* OpenESB [39] e ServiceMix [54], provenientes das comunidades de desenvolvimento Sun e Apache, são referências nesta área que exploram o conceito apresentado. A Figura 4 ilustra o cenário descrito.

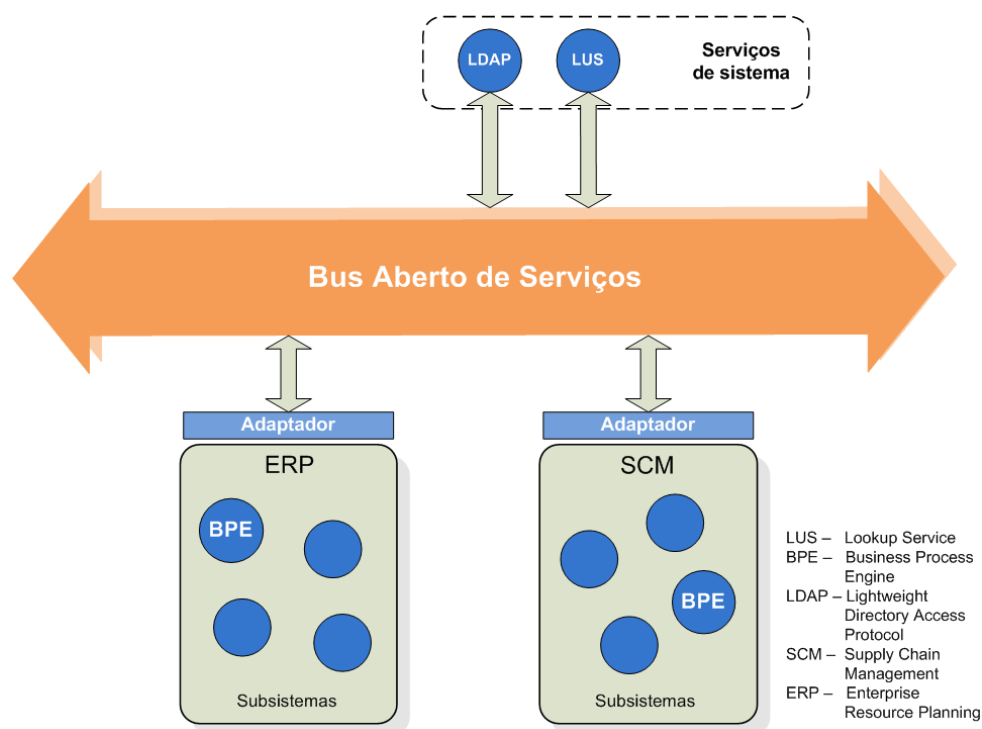


Figura 4 – Conceito de *Bus Aberto de Serviços*

De acordo com Dave Chappel, as aplicações que trocam dados em processos de negócio automatizados necessitam de comunicar umas com as outras, através da utilização de um modelo orientado a serviços, assente no conceito SOA (*Service Oriented Architecture*) [2]. Consta-se uma evolução no paradigma de desenvolvimento nas organizações, no sentido de disponibilizar os seus sistemas de informação como serviços, resultando numa tendência que contraria os actuais sistemas fechados e monolíticos, promovendo ainda a reutilização de componentes desenvolvidos [2], [45]. O sistema de execução de processos WS-BPEL Apache ODE [29], que pode ser instalado num servidor *Web* e disponibiliza o acesso aos processos/instâncias através de *web services*, é um exemplo.

O conceito de “serviço” converge para o conceito de “componente” especificado numa arquitectura orientada a componentes (SCA - *Service Component Architecture*). O objectivo do modelo SCA é simplificar a criação e integração de sistemas/aplicações, utilizando um modelo orientado a serviços (SOA). Nesta abordagem, os componentes que realizam algumas funcionalidades, para atingir um determinado objectivo do negócio, são expostos como serviços, através da definição de interfaces/contratos [47]. O conceito de “serviço aberto” conjuga a definição apresentada com a característica não proprietária da interface.

2.2. Gestão de processos de negócio

Os sistemas de informação existentes numa empresa são cada vez mais fundamentais para o seu sucesso [45][46]. A empresa depende destes sistemas para realizar as operações do seu negócio, contribuindo para melhorar o seu desempenho através da automatização dos processos de negócio. Apesar de este requisito não parecer difícil de realizar em termos abstractos, a situação no mundo real apresenta-se diferente [46]. Os processos de negócio são de natureza dinâmica, o que implica que as empresas necessitem de se actualizar e actuar de forma ágil, com vista a adaptar os seus processos aos clientes e realidades do negócio, melhorando a sua resposta global. Cada modificação num processo de negócio pode ser reflectida em diversos sistemas, coagindo as empresas a adoptarem sistemas que possam ser rápida e eficientemente adaptados às alterações do processo, permitindo que se mantenham competitivas no mercado [45], [46]. Este tipo de alterações constitui uma tarefa dispendiosa em termos de tempo e custos, para implementar, testar e colocar em produção as alterações, o que faz com que os sistemas de informação não consigam, com frequência, reagir atempadamente à dinâmica dos processos [46].

Um processo de negócio é um conjunto de actividades/tarefas que se relacionam, para colectivamente realizarem um objectivo do negócio, no contexto de uma estrutura organizacional baseada em regras e responsabilidades [17]. Embora não exista consenso no que respeita à criação de uma definição exacta de um processo de negócio, pelas diversas interpretações que cada sector da indústria adopta, é certo que este envolve um conjunto de actividades realizadas na organização, associadas às informações manipuladas e aos mais diversos recursos, para formar uma unidade focada num tipo de negócio que, tipicamente, está associado a um determinado mercado/cliente [17]. Como recursos, podem entender-se técnicas, aplicações, sistemas de informação e pessoas de uma organização. A Figura 5 ilustra esta definição.

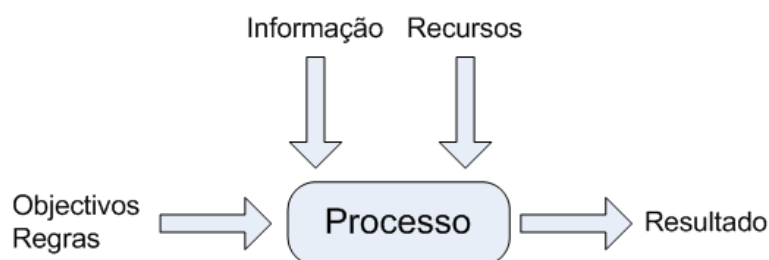


Figura 5 – Definição de processo

O conceito de BPA (*Business Process Automation*) é, ele próprio, um processo que ganha cada vez mais relevância, como o comprovam as várias iniciativas nesta área por parte de entidades referência como OASIS [8], WfMC [15], OMG [18], entre outros. A automatização é realizada através da utilização de componentes tecnológicos para substituir e/ou sustentar as actividades manuais, com o objectivo de aumentar a produtividade, reduzindo custos e riscos de falha. Para atingir uma automatização eficiente dos processos, segundo Matjaz B. Juric [5], é necessário que existam, numa organização:

- Normas para expor e aceder às funcionalidades das aplicações existentes como serviços;
- Infra-estruturas para comunicação e gestão da troca de mensagens entre serviços;
- Arquitecturas de integração entre serviços/aplicações legados e novos;
- Linguagens especializadas que permitam compor as funcionalidades expostas em serviços na definição de processos.

As metodologias clássicas de gestão e modelação de processos assentavam na utilização de diagramas de Gantt, diagramas de fluxo e diagramas de blocos. No entanto, o termo “processo” ganhou especial relevância, tornando-se o novo paradigma de produtividade. Neste sentido, a necessidade de modelação, definição e gestão de processos de negócio surgiu de forma natural e, daí, os conceitos de BPM (*Business Process Management* e *Business Process Modeling*) [51]. Alguns factores como a disparidade de formas de representação gráfica de um processo, a dinâmica do negócio e a necessidade de formas de gestão de processos mais eficazes e controladas, foram o ponto de partida para que diversas organizações/comunidades procurassem atingir o objectivo de criar notações para elementos dos processos (BPMN), linguagens de definição (WS-BPEL, XPD), sistemas de execução e funcionalidades de gestão/monitorização [53].

De acordo com o consórcio BPM (*Business Process Management Consortium*), o conceito de gestão de processos de negócio, ou BPM, surge como um conjunto de técnicas com o objectivo de melhorar, de forma iterativa e contínua, os processos envolvidos na gestão de um negócio [19]. O ciclo de vida das actividades de BPM é constituído por cinco fases: o desenho, a modelação, a execução, a monitorização e a optimização de processos de negócio, como se ilustra na Figura 6.

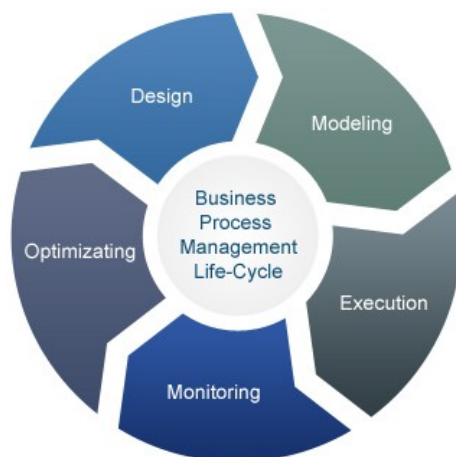


Figura 6 – Ciclo de vida da gestão de um processo de negócio²

Este conceito estabelece uma cooperação entre a gestão do negócio e as tecnologias de informação disponíveis para melhorar o suporte à automatização dos processos das organizações. Esta melhoria é alcançada através da utilização de métodos, técnicas e ferramentas para modelar, publicar, gerir e analisar processos operacionais que envolvem pessoas, aplicações, documentos e outras fontes de informação [51].

Nesta evolução da automatização dos processos de negócio, transita-se de uma filosofia de lógica de negócio embebida nas aplicações, que até então são desenvolvidas para suportar unicamente processos específicos, para uma lógica de negócio presente na definição dos processos [47]. Desta forma, procura-se decompor as aplicações envolvidas nos processos em componentes/serviços que possam ser reutilizados para desenvolvimento de outras aplicações e/ou compor outros processos. Esta característica associa o BPM ao conceito de SCA (*Service Component Architecture*) e SOA (*Service Oriented Architecture*), usufruindo das suas vantagens para construção de aplicações e processos, reutilizando serviços. A abstracção, encapsulamento e reutilização dos serviços são vantagens derivadas desta associação. Entre outros aspectos, contribuem para a redução do risco, tempo e custo de desenvolvimento de novas soluções/processos, promovendo a agilidade e competitividade da organização no mercado [19], [47].

A utilização do conceito de BPM tem crescido de forma significativa ao longo dos últimos anos [19]. Existe um conjunto de sistemas que oferece aos gestores do negócio a capacidade de modelar os seus processos, executá-los e monitorizá-los, durante e pós execução. Como resultado, estes sistemas permitem, entre outras

² Softweb Solutions (Microsoft Gold Certified Partner), <http://www.softwebsolutions.com/BPM.html>

funcionalidades, a centralização dos modelos e métricas de execução dos processos na organização [51].

No entanto, apesar das vantagens enunciadas para a utilização do conceito de BPM, constata-se que existem algumas desvantagens na sua adopção ao nível estratégico e operacional, segundo o estudo elaborado acerca da experiência de adopção das técnicas associadas ao BPM. A ausência de normas, fragilidades nas especificações existentes e a lacuna na aplicação de metodologias sobre o seu conceito, caracterizam as desvantagens ao nível estratégico/táctico. Ao nível operacional, constata-se algumas fragilidades nos sistemas de modelação e execução de processos e custos de adopção e integração de novos sistemas de suporte à tecnologia emergente nas organizações [48].

2.2.1. Definição de processos em WS-BPEL

Os especialistas nos processos de negócio deparam-se com uma diversidade de sistemas para modelação, descrição, execução e análise dos seus processos. A interface de definição de um processo assume um papel fundamental ao estabelecer um formato comum que permite a transferência de definições de processos entre diferentes plataformas [16]. Esta definição é uma das bases introdutórias para a especificação das linguagens XPDL e WS-BPEL.

A linguagem XPDL (*XML Process Definition Language*), formalizada pelo WfMC, define um formato de ficheiro XML (*XML Schema*) que é utilizado para portar definições de processos de negócio entre diferentes sistemas de execução de *workflows* [44].

A especificação WS-BPEL (*Web Services Business Process Execution Language*) é uma norma da OASIS que especifica uma linguagem, baseada em XML, para definição de processos de negócio, permitindo importar e exportar funcionalidades através de interfaces *Web Service* [3]. As origens do BPEL estão relacionadas com o WSFL (*Web Services Flow Language*) e o XLANG (extensão do WSDL (*Web Services Description Language*)), propostos, respectivamente, pela IBM e Microsoft.

A abordagem da definição de processos em WS-BPEL expõe uma filosofia de composição de serviços (*Web Services*) na formação de um processo de negócio. Sendo um dos objectivos dos *Web Services* resolver a interoperabilidade entre aplicações, utilizando normas *Web*, a utilização do modelo de integração de fraco

acoplamento (minimização da dependência entre sistemas), permite uma integração mais flexível de sistemas heterogêneos numa diversidade de domínios de B2C (*Business-To-Consumer*), B2B (*Business-To-Business*) e EAI (*Enterprise Application Integration*) [49],[50].

A definição de um processo em WS-BPEL origina um ficheiro XML, composto por um conjunto de elementos pré-definidos na sua especificação [3], a qual define a sequência de execução do processo de negócio. A sua definição enquadra-se no modelo de orquestração de serviços, assente numa coordenação centralizada, com presença de um orquestrador ou executor capaz de coordenar a execução do processo [4], [19]. A Figura 7 ilustra este modelo.

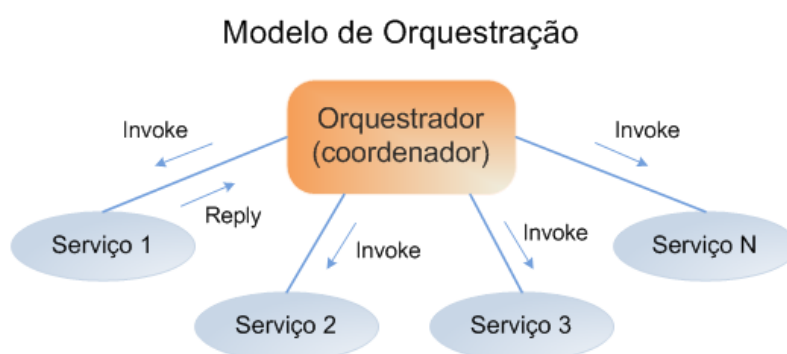


Figura 7 – Orquestração de serviços

Esta linguagem define o modelo e a gramática para descrever o comportamento dos processos e as interações com os seus parceiros (serviços). O elemento *partnerLink*, presente na especificação WS-BPEL [3], contém a informação de ligação para as diferentes partes que interagem com o processo, nomeadamente:

- Cliente do serviço associado ao próprio processo;
- Outros serviços.

Cada *partnerLink* é caracterizado por um *partnerLinkType* e um ou mais *roles* (papéis). Esta informação identifica as funcionalidades que devem ser fornecidas pelo processo e pelos serviços dos parceiros (*partners*). A Figura 8 ilustra a funcionalidade de interação proporcionada pelo elemento *partnerLink*.

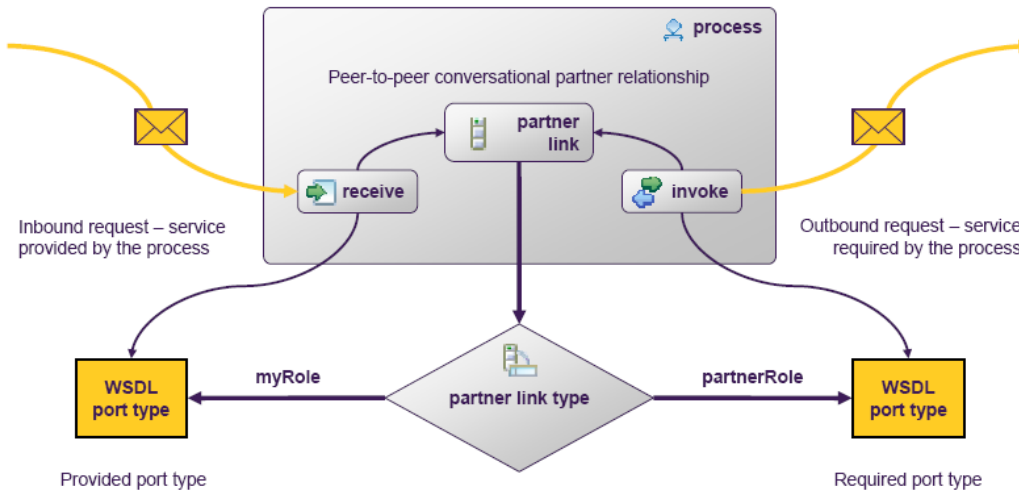


Figura 8 – Funcionalidade do elemento *partnerLink* no WS-BPEL [13]

A Figura 9 ilustra o relacionamento de um processo com os seus parceiros (serviços).

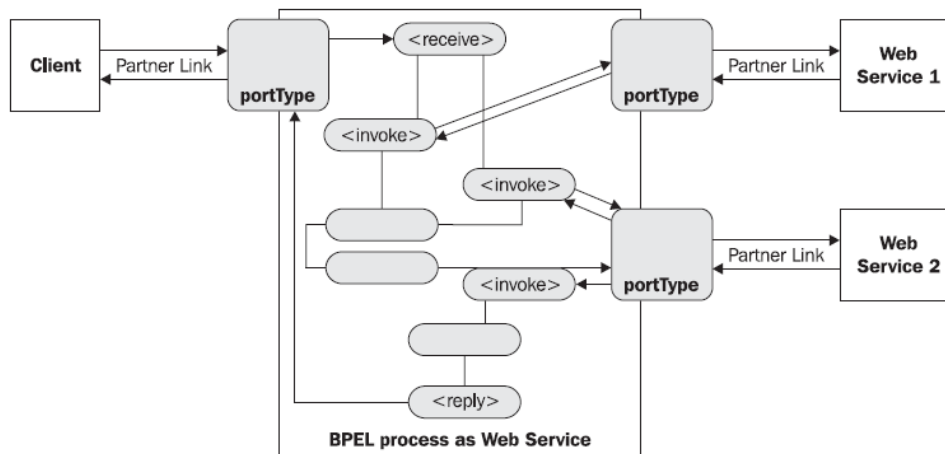


Figura 9 – Processo WS-BPEL e interações com Web Services (*partner links*) [5]

O WS-BPEL é uma proposta de solução para problemas existentes na automatização de processos de negócio, dos quais se destacam:

- Diversidade de aplicações de desenho de processos de negócio;
- Diversidade de sistemas de execução:
 - Sintaxe, terminologia, conceitos, semântica, entre outras;
- Transformação das definições dos processos para formatos de execução em sistemas de diversos fornecedores.

Todos estes factores são críticos na transformação dos processos do nível de definição (gestor de negócio) para o nível tecnológico de execução, o que, por vezes, se traduz em perda de informação para a correcta execução do processo. O WS-BPEL apresenta-se como solução para uniformizar estes aspectos, permitindo:

- Liberdade de escolha da ferramenta de modelação de processos;
- Liberdade de escolha do fabricante do sistema de execução (assumindo a compatibilidade com a especificação WS-BPEL);

A Figura 10 ilustra as vantagens que o papel do WS-BPEL proporciona na uniformização do processo de automatização.

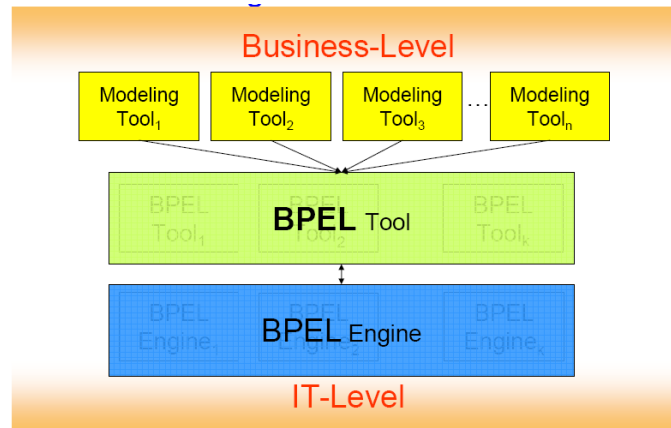


Figura 10 – Uniformização da automatização de processos com WS-BPEL [14]

A necessidade da existência de uma linguagem como o WS-BPEL relaciona-se ainda com o facto de que os *Web Services* apresentam um modelo de interacção sem estado, com mensagens trocadas sem relacionamento entre invocações. Na realidade, os processos de negócio requerem um modelo de interacção mais robusto, capaz de expressar interacções com estado. O WS-BPEL é uma norma desenhada para complemento aos *Web Services*, tal como se pode observar na Figura 11, uma vez que se expressa em XML, utiliza e estende o WSDL 1.1 e é portátil entre plataformas e fornecedores (executável em sistema de execução compatível).

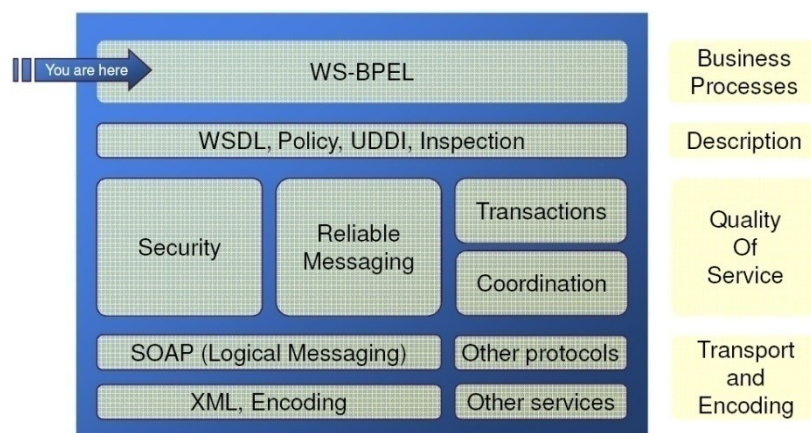


Figura 11 - WS-BPEL no stack WS-* [4]

A valorização da utilização do WS-BPEL é baseada em dois princípios fundamentais:

- Tecnologia independente de plataformas;
- Linguagem focada em abstrações (desenvolvimento orientado a processos).

O primeiro dos princípios está relacionado com a abstracção que a linguagem WS-BPEL tem relativamente ao ambiente de execução onde o processo será executado e, daí, a vantagem de poder responder às necessidades individuais de automatização de processos, em cada organização. Em relação ao segundo princípio, o WS-BPEL não mistura definições de processos com construções mais detalhadas, tais como as que podem ser encontradas nas linguagens de programação, como C# ou Java. Desta forma, esta linguagem permite uma separação entre a definição de um processo, através da composição das suas actividades e a implementação dos componentes que o processo utiliza/invoca, com granularidade mais fina. A definição de um processo em WS-BPEL é vista segundo uma perspectiva declarativa, onde se define a constituição do processo e a sua sequência de execução, comparativamente a uma perspectiva imperativa que explicita como é que as actividades/componentes (serviços, entre outros) concretizam os seus objectivos.

2.2.2. Outras iniciativas de referência

Existem diversas iniciativas organizadas em forma de consórcios ou simples parcerias no contributo para impulsionar a adopção do conceito de BPM nas organizações, nas suas mais diversas áreas, incluindo a gestão/monitorização de processos de negócio. De entre estas organizações, destacam-se o WfMC e OMG para os quais se apresentam contribuições nesta área.

WfMC - *Workflow Management Coalition*

O WfMC [15], fundado em 1993, foi um dos impulsionadores do BPM (*Business Process Management*) e BPA (*Business Process Automation*). A sua abordagem é caracterizada pela automatização dos processos de negócio através de sistemas de gestão de *workflow*, orientados para a gestão documental, durante os quais, documentos, informação ou tarefas são transferidos de um participante para outro, de acordo com a definição de regras de negócio. O processo de *workflow* é tradicionalmente associado ao movimento de documentos e/ou tarefas no contexto dos processos de trabalho numa organização [16].

O WfMC apresenta como objectivos, promover a normalização do mercado de *workflow* e a redução do risco de adopção deste tipo de produtos. Define um sistema de *workflow* como sendo um sistema que gere e executa *workflows*, ou fluxos de trabalho, pela execução de *software* compatível com a linguagem de definição do processo. Existe, segundo estes parâmetros, uma execução centralizada de um plano de actividades, através da definição do processo na linguagem XPD (XML *Process Definition Language*), associação de participantes, utilizadores e outros sistemas, bem como a presença dos conceitos de *worklist/workitem* para envolvimento de utilizadores em tarefas não automáticas. A iniciativa WfMC possui um modelo de referência para especificação de uma infra-estrutura para sistemas de gestão de processos, identificando as suas características, componentes, funções e interfaces [16].

A arquitectura de um sistema de *workflow*, presente neste modelo de referência, é ilustrada na Figura 12, onde é possível observar o ambiente de execução de *workflows* (*Workflow Enactment Service*) e as suas diversas interfaces.

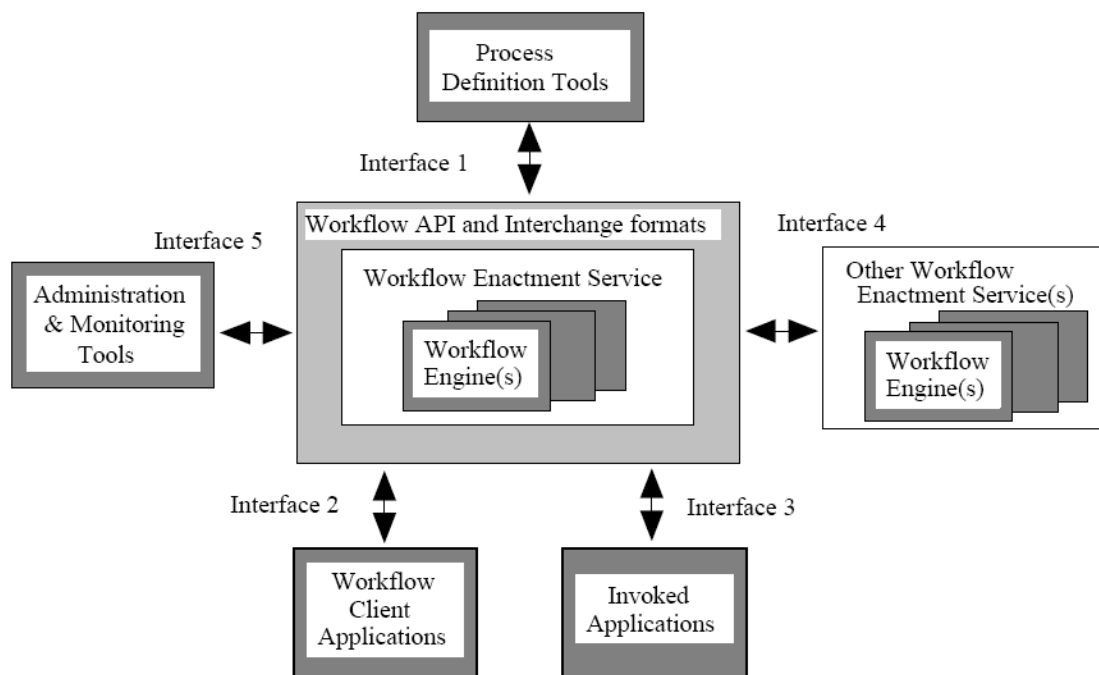


Figura 12 – Arquitectura de um sistema de *workflow* [15]

O sistema de execução de *workflow* é distinto das aplicações e ferramentas do cliente utilizadas para processar *workitems*. Sobre este, em cada uma das suas interfaces, pode ser utilizado um conjunto de especificações ou ferramentas que permitem formar um sistema de gestão de *workflow* tal como ilustrado na figura 12. Cada uma das *interfaces* desempenha um papel específico neste modelo de referência, tal como se descreve de forma sucinta:

- **Interface 1 – *Process Definition Tools***
 - Definição de uma interface normalizada entre a ferramenta de definição dos processos e o motor de *workflow*;
- **Interface 2 – *Workflow Client Application***
 - Definição de interfaces para que as aplicações cliente possam solicitar serviços do motor de *workflow* de modo a gerir o progresso dos processos, actividades e *workitems*;
- **Interface 3 – *Invoked Application***
 - *Interface* normalizada que permite ao motor de *workflow* activar/invocar outras aplicações/serviços;
- **Interface 4 – *Workflow Interoperability***
 - Definição de um conjunto de modelos de interoperabilidade entre sistemas de *workflow*;
- **Interface 5 – *Administrations & Monitoring Tools***
 - Definição de funções de gestão e monitorização normalizadas.

É possível verificar que este modelo de referência prevê as funcionalidades de execução, invocação de aplicações e gestão/monitorização dos processos de negócio através do sistema de execução e das respectivas interfaces 3 e 5. Estas funcionalidades são referências para a definição, execução e monitorização de processos WS-BPEL.

WF - Windows Workflow Foundation

A tecnologia WF [20] da Microsoft surge associada ao conceito de *workflow* mas concretiza uma visão distinta da que é defendida pelo WfMC relativamente à gestão de processos de negócio. A Microsoft enuncia que se trata de um modelo de programação para desenvolvimento de aplicações, aplicando uma lógica de abordagem declarativa (*workflows*) face a uma abordagem de desenvolvimento imperativa, associada a linguagens de programação como C# [21]. A Figura 13 ilustra o *workflow* na perspectiva do WF.

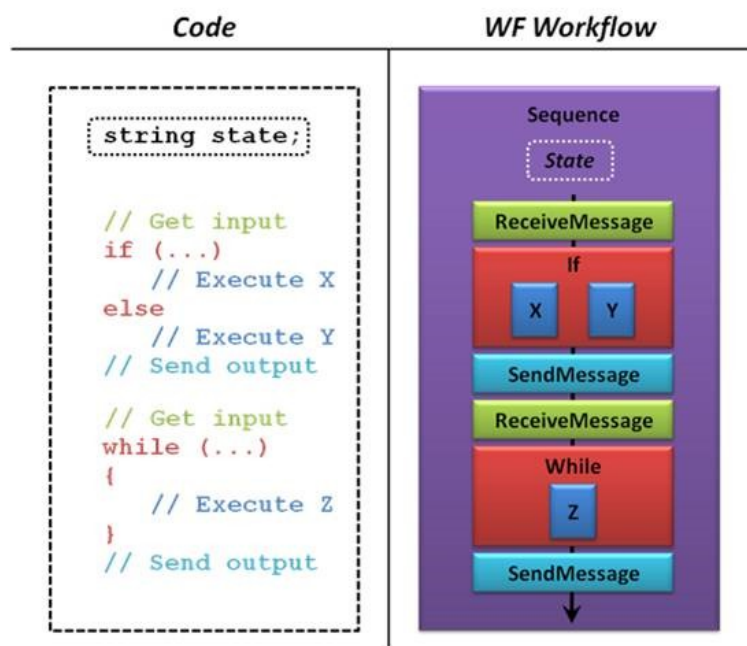


Figura 13 – Workflow na perspectiva do Windows Workflow Foundation [21]

Como se observa na figura 13, a tecnologia WF aplica-se à composição de componentes, explorando a sua reutilização, para desenvolvimento de sistemas/aplicações, no âmbito da engenharia de *software*. A perspectiva declarativa enunciada difere da perspectiva declarativa da composição de processos de negócio, pela forma como está associada à definição de como os componentes devem realizar as suas funções. Desta forma, não constitui objecto de estudo deste projecto.

OMG – Object Management Group

O OMG (*Object Management Group*) [18] é um consórcio internacional da indústria informática, sem fins lucrativos, fundado em 1989. Esta iniciativa especializa-se no desenvolvimento de normas de integração e modelação. As suas especificações, onde se destacam a norma UML (*Unified Modeling Language*) [42] e MDA (*Model Driven Architecture*) [43], permitem enriquecer e melhorar o desenho da arquitectura das soluções, execução e manutenção de *software*, incluindo modelação de sistemas de tecnologias de informação e gestão de processos de negócio.

Este consórcio alberga outras iniciativas especializadas em diversas áreas, das quais é relevante destacar, para este projecto, a iniciativa BMI DTF (*Business Modeling & Integration Domain Task Force*) [23].

Business Modeling & Integration (BMI) Domain Task Force (DTF)

A iniciativa BPMI (*Business Process Management Initiative*) [23] apresenta como objectivo desenvolver especificações de modelos integrados de apoio à gestão de organizações, orientados sobre os conceitos de BPM. As suas especificações, entre as quais constam referências como o BPMN (*Business Process Modeling Notation*) utilizado por diversas organizações como SAP, ORACLE ou Tibco, ou o BPML (*Business Process Modeling Language*), promovem a integração e colaboração inter e intra-organização de pessoas, sistemas e processos. Algumas das actividades do BPMI incluem o refinamento e promoção da norma BPMN (*Business Process Modeling Notation*), soluções de BIM (*Business Information Management*), EAI (*Enterprise Application Integration*), B2B (colaboração *Business to Business*), promoção e educação de princípios e abordagens a tecnologias de BPM [23].

Em Junho de 2005, as iniciativas BPMI (*Business Process Management Initiative*) [23] e OMG anunciaram a fusão das suas actividades de BPM, de forma a disponibilizar novas soluções e normas, adoptando a denominação de BMI DTF [23]. A referida fusão permitiu aliar as normas de modelação e integração de sistemas da comunidade OMG à missão que advém da iniciativa BPMI, unificando um conjunto de normas no âmbito da gestão de processos de negócio [18].

Considera-se que a aplicação destas normas carece de um estudo aprofundado de um conjunto de especificações que, pela abrangência que possuem, se distanciam do objectivo principal de uma solução de gestão integrada da execução de processos definidos em WS-BPEL.

2.3. Execução de processos WS-BPEL

Existe uma diversidade de sistemas de execução de processos WS-BPEL proprietários e alguns *open-source*, derivados de iniciativas de comunidades de referência como Apache [27] ou jBoss [40]. Salieta-se que o termo “orquestrador” pode ser utilizado para referir um “sistema de execução de processos”. Considera-se que os termos são equivalentes porque o sistema de execução de processos tem o comportamento de um orquestrador, segundo a definição de modelo de orquestração apresentada no capítulo “2.2.1 Definição de processos em WS-BPEL”.

A tabela seguinte apresenta exemplos desses sistemas.

Nome	Fabricante	Página do produto	Licença
BizTalk	Microsoft	http://www.microsoft.com/biztalk/en/usa/default.aspx	Proprietária/ Comercial
WebSphere Process Server	IBM	http://www-01.ibm.com/software/integration/wps/	Proprietária/ Comercial
ORACLE Bpel Process Manager	Oracle Corporation	http://www.oracle.com/technology/products/ias/bpel/index.html	Proprietária/ Comercial
Apache ODE	Apache Software Foundation	http://ode.apache.org/	Apache
ActiveBPEL	Active Endpoints	http://www.activeendpoints.com/community-open-source.php	GPL v2
jBPM	jBoss	http://www.jboss.org/jbossjbpm/jbpm_overview/	LGPL

Tabela 2 – Soluções de sistemas de execução WS-BPEL

Relativamente à gestão da execução, estes sistemas disponibilizam interfaces programáticas proprietárias, definidas em função do seu modelo interno de objectos, que permitem obter informação de monitorização em tempo-real e/ou pós execução, através de formatos específicos. Estas interfaces especificam diferentes formatos de informação e tipos de acesso que diferem entre fabricantes, dificultando a uniformização da gestão da execução dos processos em diversos sistemas. Este facto reforça a motivação para a elaboração do projecto, sendo um dos factores para a adopção de uma solução que minimize o impacto das interfaces programáticas proprietárias neste quadro de gestão integrada.

2.4. Monitorização de processos WS-BPEL

A capacidade de monitorizar um processo constitui um aspecto crítico no sentido de acompanhar o comportamento da execução das suas actividades. O acesso à informação do processo revela-se útil em cenários de monitorização em tempo real e/ou pós-execução, para avaliação e correcção de erros. A monitorização permite identificar problemas e é o garante da confiança, disponibilidade e comportamento esperados para o processo [16].

Os sistemas de gestão de processos WS-BPEL, particularmente os enunciados no subcapítulo anterior, incluem subsistemas de gestão que disponibilizam funcionalidades de monitorização dos processos em execução e/ou terminados através de interfaces programáticas proprietárias. Existem iniciativas de criação de normas para gestão de sistemas passíveis de serem enquadrados no âmbito deste projecto, das quais se destaca o DMTF (*Distributed Management Task Force*) [22] que visa criar modelos de gestão de sistemas num ambiente cooperativo.

DMTF – *Distributed Management Task Force*

O DMTF [22] é uma das iniciativas de normalização que contribui para a criação de sistemas de gestão independentes de plataformas e tecnologias. Ao criar especificações abertas, o DMTF promove a interoperabilidade entre sistemas de gestão entre produtos de diferentes fornecedores. Uma das iniciativas do DMTF denomina-se WBEM (*Web-Based Enterprise Management*) e representa um conjunto de normas de tecnologias de gestão e Internet desenvolvido para unificar a gestão em ambientes de computação. Entre estas normas constam o CIM (*Common Information Model*), *xmlCIM Encoding Specification* e *CIM Operations over HTTP*.

A especificação CIM é a linguagem e metodologia para descrição de dados de suporte a processos de monitorização. O *schema* DTD do CIM inclui modelos para sistemas, aplicações, redes e dispositivos, permitindo que aplicações de diferentes fornecedores em diferentes plataformas descrevam o seu modelo de dados de gestão num formato uniforme que possa ser partilhado entre diversas aplicações de gestão. A especificação *xmlCIM Encoding* define elementos XML, descritos no respectivo DTD, que podem ser utilizados para representar classes e instâncias do modelo CIM. A especificação *CIM Operations over HTTP* define o mapeamento das operações do CIM em http [22]. Esta especificação não foi considerada devido à escolha de uma tecnologia de gestão baseada em recursos, de acordo com o artigo de referência estudado e adoptado para a realização do projecto [1].

2.5. Tecnologias de gestão baseadas em recursos

Perante a necessidade de aceder uniformemente à informação de gestão dos processos que um sistema de execução possui, este necessita de expor as suas estruturas internas de dados. A estratégia proposta contempla a exposição da informação de gestão dos processos WS-BPEL através de recursos que podem ser geridos. Esta abordagem, de representação genérica da informação dos processos, possibilita que a solução seja implementada directamente sobre o modelo de dados do sistema de execução ou sobre a informação proveniente de interfaces programáticas específicas. Quanto à obtenção de informação de gestão/monitorização através de interfaces programáticas específicas, existe uma dependência da informação que é possível obter em função das limitações inerentes à interface do sistema de execução. A capacidade de abstracção relativamente ao modelo de dados interno do sistema de execução, existente com uma representação genérica da sua informação, é uma das vantagens desta solução, permitindo aos fornecedores destes sistemas manterem os modelos de dados mais convenientes [1].

Apresenta-se um estudo sobre tecnologias de gestão baseadas em recursos, mencionadas como estratégias de solução no artigo de referência “A Management Framework for WS-BPEL” [1]. O estudo resulta na selecção de uma das tecnologias como fundamentação teórica para a solução implementada no projecto.

JMX - Java Management Extensions

A tecnologia JMX [24] permite construir soluções de gestão para monitorizar dispositivos, aplicações e redes de serviços de forma modular, dinâmica, distribuída e baseada em tecnologias *web*. O JMX pode ser utilizado para gestão de aplicações, localmente ou remotamente, através de RMI (*Remote Method Invocation*), HTTP ou *Web Services*.

A interface de gestão é denominada de MBean (*Managed Bean*) e permite definir as operações de monitorização que podem ser expostas. Os métodos *getters* e *setters* são automaticamente reconhecidos e expostos como atributos e os restantes métodos são expostos como operações para invocações associadas ao processo de monitorização. Esta tecnologia suporta a publicação e subscrição de notificações e, através da sua utilização, um dado recurso pode ser gerido/monitorizado por uma ou mais MBeans, através de agentes JMX. Estes agentes permitem controlar os recursos e disponibilizam-nos para as aplicações de gestão/monitorização.

A Figura 14 ilustra a arquitectura JMX, onde é possível observar os níveis desde a instrumentação até às soluções de gestão de mais alto nível.

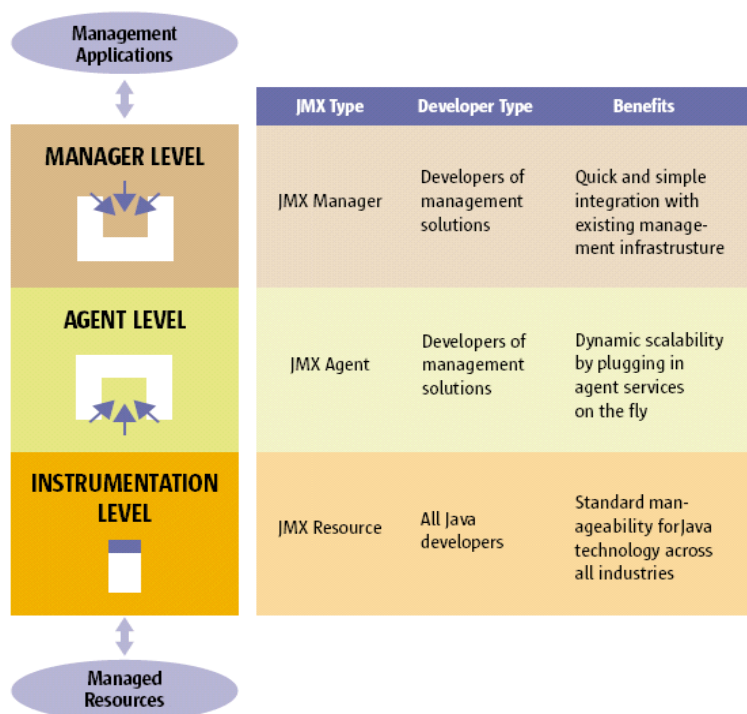


Figura 14 – Arquitectura JMX [25]

Na abordagem ao desafio proposto neste projecto, considera-se que a solução de monitorização de processos WS-BPEL pode ser realizada através de JMX. Esta seria constituída por recursos mapeados para MBeans, as suas propriedades mapeadas para atributos desta tecnologia e os eventos passados às aplicações de monitorização via mecanismo de notificação JMX.

SNMP - Simple Network Management Protocol

O protocolo SNMP é utilizado para gestão de sistemas numa rede. Na sua arquitectura, constam sistemas de gestão e elementos da rede. Os sistemas de gestão executam aplicações de monitorização e controlo dos elementos da rede. Por elementos da rede, entendem-se equipamentos como *routers*, *gateways* ou servidores que possuem agentes responsáveis por desempenhar as funções requeridas pelos sistemas de gestão. O protocolo é utilizado para comunicar a informação de gestão entre sistemas de monitorização e os agentes presentes nos elementos da rede [52]. A referida informação é descrita em MIBs (*Management Information Base*). A entidade MIB define um conjunto de atributos e a interface de gestão do elemento que é gerido, à semelhança do conceito de MBean da tecnologia JMX. Apesar de ser um protocolo de referência, não foi elaborado nenhum estudo aprofundado acerca das suas

características, no âmbito da gestão de execução de processos WS-BPEL, de acordo com as tecnologias propostas pelo artigo de referência [1].

REST - Representational State Transfer

O REST é um estilo de arquitectura muito popular para sistemas distribuídos orientados a recursos (tais como *World Wide Web*) tendo sido introduzido por Roy Fielding na sua dissertação de doutoramento [26]. Segundo o autor, o REST utiliza os verbos básicos de HTTP: *GET* para receber, *PUT* para actualizar, *POST* para criar e *DELETE* para apagar um recurso. Nesta arquitectura, o autor refere que os recursos são identificados univocamente através de URIs (*Uniform Resource Identifier*) e estão interligados através de *hyperlinks*, para tornar o modelo de recursos pesquisável, tanto para pessoas como máquinas. Através da utilização de um modelo de URIs, é possível distinguir recursos de colecções, ou seja, recursos que contêm apenas *hyperlinks* para outros recursos. Desta forma, existe a capacidade de expressar recursos e sub-recursos, tal como se pretende para a solução proposta neste projecto. Relativamente a notificações, estas não são suportadas directamente pela arquitectura REST. No entanto, existem algumas extensões, como GENA (*General Event Notification Architecture*) ou RNA (*RESTful Notification Architecture*), que permitem a publicação/subscrição de eventos via *polling* ou canal de comunicação assíncrona.

Esta tecnologia, ainda que não apresente uma definição concreta de recurso, que se enquadre no âmbito da representação de processos WS-BPEL, é uma candidata para a implementação da estratégia de solução.

WSRF - Web Services Resource Framework

Os sistemas de gestão/monitorização de processos WS-BPEL necessitam de consultar a informação dos recursos através de interfaces e formatos de dados uniformes.

A comunidade OASIS possui um conjunto de especificações que formam um quadro de desenvolvimento genérico e aberto para modelar e aceder a recursos com estado, utilizando *Web Services*. Esse quadro é denominado por WSRF (*WS-Resource Framework*) e inclui, entre outros, mecanismos para descrever vistas sobre o estado dos recursos e gestão do respectivo estado através de propriedades.

O trabalho inicial sobre esta tecnologia foi desenvolvido pela IBM e pela Globus Alliance, com a definição de uma arquitectura e respectiva especificação, que foram

submetidos a aprovação da comunidade OASIS. Após aprovação, foram posteriormente adoptados como normas desta comunidade de referência.

O quadro WSRF visa ainda explicitar um conjunto de operações que os *Web Services* podem implementar para que possuam estado, utilizando um “recurso” como suporte ao armazenamento da informação. Para obter/afectar as propriedades do recurso e, conseqüentemente, a informação que armazena, o cliente invoca o *Web Service* incluindo o seu identificador.

Um “recurso”, segundo a definição presente na especificação WS-Resources [10], é uma entidade identificável, com zero ou mais propriedades para armazenamento de informação. Esta especificação da OASIS, define um WS-Resource como uma composição de um recurso (independentemente do formato como são armazenados os dados) e um *Web Service* através do qual o recurso está acessível. A Figura 15, presente no documento da especificação WS-Resources, ilustra este conceito.

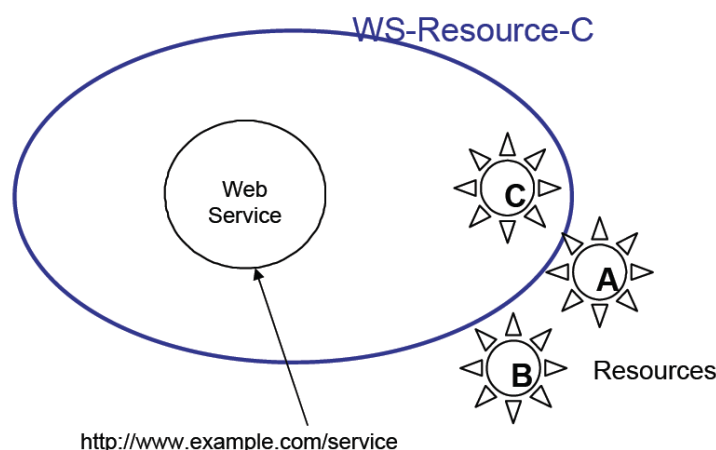


Figura 15 – Conceito de recurso segundo norma WS-Resource [10]

Um recurso é definido através de propriedades, que podem ser interrogadas e modificadas, utilizando a especificação WS-ResourceProperties [11] da OASIS, também, à data deste documento, na versão 1.2. Segundo a comunidade OASIS, esta especificação de referência define uma propriedade de um recurso, enquadrada no modelo WS-Resource, como uma peça de informação que reflecte o seu estado. O documento onde são armazenadas as propriedades do recurso denomina-se por *WS-ResourceProperties Document*. Para além destas definições, a norma WS-ResourceProperties especifica ainda as operações de obtenção e actualização de valores sobre as propriedades do recurso. Das diversas operações, destacam-se as que se consideram relevantes para a implementação de um demonstrador da solução proposta: *GetResourcePropertyDocument* e *GetResourceProperty*. Estas operações

permitem, respectivamente, obter todas as propriedades ou apenas uma propriedade do recurso.

A capacidade de publicação/subscrição de eventos/notificações sobre alterações do estado dos recursos, através da especificação WS-BaseNotification [12] da OASIS, é uma característica que enquadra a tecnologia nos requisitos do projecto.

A análise efectuada ao conjunto de especificações WS-Resource, Ws-ResourceProperties e WS-BaseNotification, constituintes do quadro WSRF, comparativamente com as tecnologias JMX ou REST, permite concluir que esta tecnologia é adequada para a sua adopção na implementação da estratégia de gestão baseada em recursos. Esta conclusão é justificada pelo facto das especificações pertencerem à iniciativa de referência OASIS e terem como objectivo comum garantirem uma gestão integrada baseada em especificações de referência e independente das características dos sistemas envolvidos. A *framework* WSRF permite especificar explicitamente o conceito de recurso e propriedades para armazenamento da informação, método de acesso através de *Web Services* e interface programática uniforme para as operações do serviço, características que se enquadram nos objectivos, requisitos e arquitectura da solução proposta para este projecto.

2.6. Síntese

Uma organização possui uma plataforma computacional composta por diversos sistemas de informação. Estes contribuem para melhorar o seu desempenho através da automatização dos processos de negócio.

Um processo de negócio é um conjunto de actividades/tarefas que se relacionam para, colectivamente, realizarem um objectivo do negócio.

Os especialistas nos processos de negócio deparam-se com uma diversidade de sistemas para modelação, descrição, execução e análise dos seus processos. As normas XPD, da iniciativa WfMC e WS-BPEL, da iniciativa OASIS, são exemplos de linguagens de definição de processos. Estas linguagens estabelecem um formato comum que permite a transferência de definições entre diferentes plataformas de execução.

Os processos definidos em WS-BPEL são executados segundo um modelo de orquestração, constituído por um coordenador que gere toda a execução do processo.

Neste quadro, um executor de processos WS-BPEL é visto como um sistema que deve ser integrado, na plataforma computacional, segundo um modelo aberto e cooperativo, para disponibilizar informação de gestão/monitorização dos processos que executa. O acesso a esta informação revela-se útil em cenários de monitorização em tempo real e/ou pós-execução, como garante da confiança, disponibilidade e comportamento esperados para o processo. A integração é realizada através de interfaces programáticas específicas de cada fabricante. Este facto contribui para dificultar a uniformização dos dados trocados entre sistemas de execução e gestão, bem como para o aumento da dependência entre estes sistemas.

Perante a necessidade de aceder uniformemente à informação de gestão dos processos que um sistema de execução possui, este necessita de expor a informação segundo num formato uniforme. A estratégia proposta contempla a exposição da informação através de recursos que podem ser geridos.

Os sistemas de gestão/monitorização de processos WS-BPEL necessitam de consultar a informação dos recursos através de interfaces abertas, não proprietárias, e formatos de dados uniformes. A comunidade OASIS possui um conjunto de especificações, denominado WSRF, que formam um quadro de desenvolvimento genérico e aberto para modelar e aceder a recursos com estado.

3. Arquitectura proposta

Este capítulo descreve a análise de requisitos, a arquitectura e estratégia de solução proposta para efectuar a gestão integrada de sistemas de execução de processos WS-BPEL.

Os fornecedores de sistemas de execução adoptam uma estratégia, segundo a qual a informação de gestão/monitorização dos processos de negócio é representada em formatos específicos de cada sistema, culminando em interfaces de gestão proprietárias. Os autores do artigo de referência [1] propõem uma estratégia alternativa, assente em normas de referência, que permita representar a referida informação de gestão/monitorização uniformemente para sistemas de execução e gestão. Esta representação permite a criação de interfaces programáticas de gestão normalizadas, minimizando a dependência entre estes sistemas [1].

A Figura 16 ilustra a arquitectura proposta pelos autores.

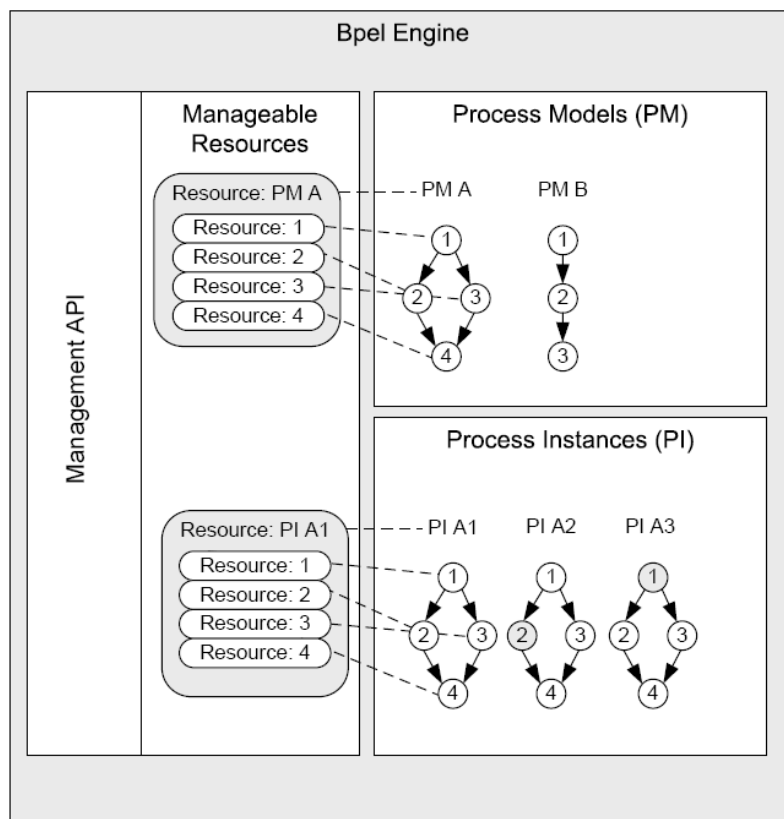


Figura 16 - Arquitectura proposta em "A Management Framework for WS-BPEL" [1]

Um dos objectivos é promover a flexibilidade, através da capacidade de substituir/adicionar sistemas de execução WS-BPEL, sem necessidade de adaptar as aplicações de gestão dos processos, criando e uniformizando a interface programática de gestão/monitorização. Este objectivo denota um enquadramento no conceito de *bus*

aberto de serviços (introduzido no capítulo anterior) pela aproximação do sistema de execução a um serviço que pode ser inserido (*plugged*) num ambiente integrado através da sua interface de gestão/monitorização uniformizada. Os autores validam a proposta de solução através de desenvolvimentos no sistema de execução de processos Apache ODE [29]. A representação dos modelos dos processos/instâncias em recursos e uma nova interface de gestão/monitorização, numa abordagem não proprietária e independente do seu modelo interno de objectos, são alguns dos desenvolvimentos efectuados [1].

3.1. Hipóteses consideradas

A análise resultante do enquadramento deste projecto no âmbito das tecnologias, metodologias e sistemas para gestão de processos de negócio, pelas principais iniciativas e comunidades de referência, aliado ao objectivo de um modelo de integração aberto e cooperativo, permitiu equacionar duas hipóteses para efectuar a gestão integrada de sistemas de execução de processos WS-BPEL.

As abordagens propostas consideram a implementação de adaptadores para encapsular as funcionalidades de integração. Através destes adaptadores, é possível uniformizar o método de acesso e formato da informação trocada entre sistemas de execução WS-BPEL e aplicações de gestão. Esta uniformização contribui para a unificação da gestão de processos de negócio WS-BPEL, em execução em diferentes sistemas, numa única aplicação de gestão/monitorização [1].

Abordagem de uniformização da gestão nos sistemas de execução

A solução inerente a esta hipótese considera que os sistemas de execução de processos WS-BPEL são responsáveis por uniformizar o acesso e formato da informação de gestão de processos. Esta característica permite reduzir a dependência causada pelas interfaces programáticas de gestão proprietárias, relativamente às aplicações de gestão de processos. A hipótese equacionada contempla a implementação de adaptadores nos respectivos sistemas para transformar os seus dados internos de gestão de processos/instâncias para um modelo de representação independente, disponibilizando uma interface programática de gestão/monitorização uniforme, segundo normas de referência transversais aos sistemas de execução.

A Figura 17 ilustra esta hipótese de solução.

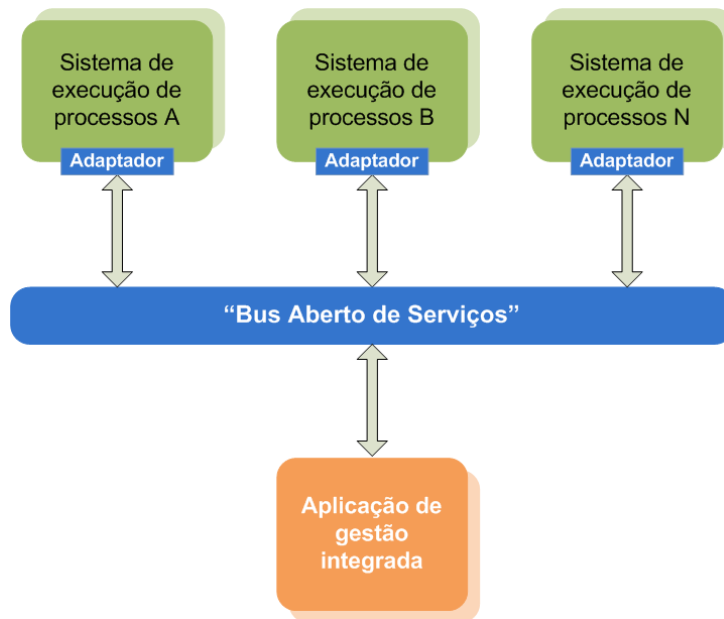


Figura 17 - Hipótese de solução com uniformização dos sistemas de execução

Esta solução evidencia uma aproximação ao conceito de *bus* aberto de serviços, introduzido no enquadramento deste projecto. O desenvolvimento dos adaptadores, segundo este conceito de integração, permite criar uma solução com um modelo de fraco acoplamento entre sistemas de execução e aplicações de gestão de processos. Esta característica adiciona flexibilidade à integração de sistemas heterogéneos, promove a abstracção e o encapsulamento dos detalhes de implementação dos sistemas envolvidos e a adaptabilidade, no sentido de adição/modificação de requisitos/componentes na respectiva gestão integrada. Estes factores contribuem para a obtenção de uma solução escalável relativamente à capacidade de suportar a incorporação de novos sistemas de execução e gestão [2] [47].

Todos estes factores contribuem para a utilização de uma aplicação de gestão de processos unificada, receptiva a formatos de informação uniformizados, sem necessidade de adaptação à especificidade de cada sistema de execução.

Apesar das vantagens enunciadas, constatam-se algumas desvantagens na transição do desenho da solução para a fase de implementação. Existe uma dificuldade de ruptura com a cultura de integração praticada nas organizações, onde existem sistemas de execução de processos com interfaces programáticas de gestão proprietárias e aplicações desenhadas à medida para estas interfaces, com provas dadas de funcionamento, os designados sistemas legados. O tempo e custos, associados ao desenvolvimento ou adaptação de sistemas legados a uma nova metodologia de integração, constituem entraves à adopção desta solução. Aliada a

estes factores de decisão, existe a necessidade de desenvolver uma norma genérica para especificar o formato da informação de gestão/monitorização de processos WS-BPEL, trocada entre sistemas de execução e aplicações de gestão [2].

As desvantagens referidas para esta hipótese de solução são impulsionadoras da consideração de uma outra que se aproxima da realidade da actividade de integração praticada nas organizações, bem como das actuais tecnologias, sistemas de execução e aplicações de gestão de processos WS-BPEL existentes.

Abordagem de integração na aplicação de gestão

A abordagem alternativa considera a realização de adaptadores na aplicação de gestão, sendo esta a responsável por conter a lógica de integração com cada sistema de execução, permitindo manter estes últimos inalterados. Esta solução apresenta um modelo acoplado entre sistemas de execução e aplicação de gestão, tornando-a menos flexível, adaptável e escalável do que a solução anterior (nos mesmos parâmetros de análise).

A Figura 18 ilustra a abordagem considerada nesta hipótese de gestão integrada.

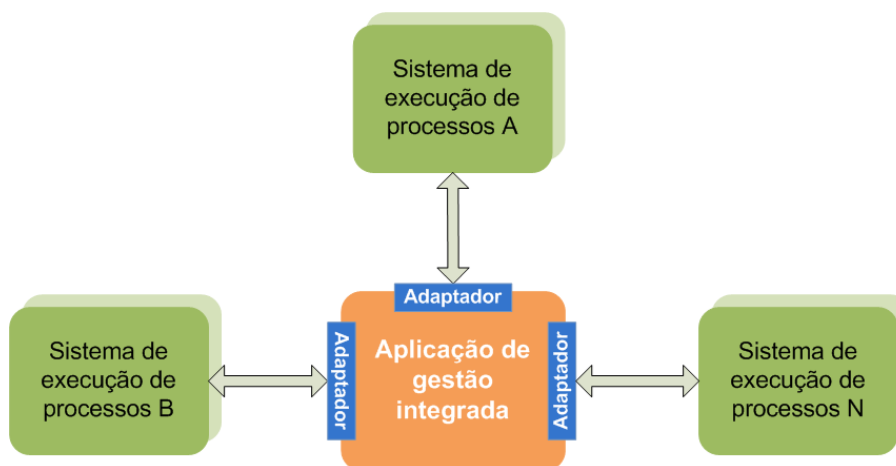


Figura 18 – Hipótese de solução com integração na aplicação de gestão

Contrariamente ao que se sucede no modelo baseado num *bus* aberto de serviços, é necessário adequar a aplicação de gestão à especificidade de cada sistema de execução, relativamente a diversos factores como tipo de acesso/comunicação, interfaces programáticas e formato da informação disponibilizada por estes. Esta especificidade é encapsulada na camada de integração que é constituída pelos adaptadores ilustrados na figura anterior. Este modelo converge para o anterior, se os

adaptadores forem adoptados como norma/iniciativa aberta, promovendo uma integração normalizada entre sistemas de execução e de gestão.

Em termos de implementação da solução, considera-se, nesta proposta, que a integração ocorre na aplicação de gestão, dado que não existe qualquer requisito de alteração dos sistemas de execução ao nível do modelo interno de dados, interfaces de gestão/monitorização e formato da informação disponibilizada.

A hipótese aqui apresentada é uma solução de compromisso de acordo com um tipo de integração que é adoptado actualmente em algumas organizações, constituído por sucessivas camadas de integração entre sistemas, desenvolvidas à medida das necessidades. É o resultado da cultura e tendência natural para desenvolvimento de sistemas fechados e monolíticos e da dificuldade na mudança de padrões tecnológicos, sendo que, uma das causas, é o facto de as empresas serem conservadoras (“se está a funcionar, então não se justifica mudar”) [45]. Estes aspectos são mencionados por Dave Chappell e associados a um conceito que o autor denomina de “arquitetura acidental” [2]. A “arquitetura acidental” é o resultado de diversos anos de soluções de integração, ponto a ponto “*one-of-a-kind*”, desenhadas para responder a objectivos muito específicos de comunicação ou transformação de dados. Estes objectivos são concretizados em infra-estruturas rígidas, pouco coesas, com sistemas fechados, pouco flexíveis e fracamente adaptados à dinâmica da evolução tecnológica e dos requisitos do negócio [2], [45].

Esta abordagem de solução foi a considerada para implementação do sistema de gestão integrada da execução de processos WS-BPEL, em detrimento da hipótese anterior, pelo enquadramento descrito. A redução do investimento (tempo e custos) de alteração dos sistemas de execução já implementados e sobre os quais os gestores do negócio confiam (sem necessidade de os modificar ou criar rupturas com fornecedores) é um factor decisivo [1].

As desvantagens inerentes a esta solução estão directamente associadas à existência da camada de integração (adaptadores) com cada sistema de execução, referida na introdução desta hipótese e ilustrada na Figura 18, o que promove um maior acoplamento entre sistemas. Esta é responsável por encapsular a lógica de acesso e transformações dos dados, necessários para aceder à informação de gestão/monitorização dos processos, gerir a actualização dessa informação e disponibilizá-la, de forma uniforme, para uma aplicação de gestão unificada. Por estas funcionalidades e pela sua localização na hipótese apresentada, constitui a principal

limitação da solução, em termos de flexibilidade e escalabilidade. Por cada sistema de execução WS-BPEL que é adicionado/substituído, é necessário replicar estes adaptadores.

A tabela seguinte sumariza a comparação das duas abordagens.

Abordagem	Vantagens	Desvantagens
<p>Uniformização da gestão nos sistemas de execução</p>	<ul style="list-style-type: none"> • Promoção de modelo de fraco acoplamento entre sistemas envolvidos • Abstracção dos detalhes de implementação de cada sistema • Adaptabilidade da solução para incorporação de novos requisitos ou alteração dos existentes • Flexibilidade de adição de sistemas de execução/ gestão 	<ul style="list-style-type: none"> • Dificuldade de ruptura com cultura de integração (à medida das necessidades) praticada nas organizações • Tempo e custos associados à adaptação de sistemas em execução para nova metodologia de integração • Desenvolvimento de norma genérica, para formato de informação e acesso, transversal aos sistemas
<p>Integração na aplicação de gestão</p>	<ul style="list-style-type: none"> • Aceitabilidade da solução - aproximação à metodologia de integração praticada nas organizações (soluções à medida das necessidades) • Minimização de rupturas com fabricantes de sistemas • Exequibilidade da solução sobre os sistemas desenvolvidos e em execução (não necessitam de alteração) 	<ul style="list-style-type: none"> • Maior acoplamento entre sistemas • Redução da escalabilidade, flexibilidade e adaptabilidade de adição de novos sistemas de execução/gestão

Tabela 3 - Comparação das abordagens propostas

3.2. Objectivos propostos

Os objectivos propostos para a abordagem de solução considerada incluem o estudo e utilização de um sistema de execução de processos WS-BPEL *open source*, com respectiva interface programática de gestão/monitorização e capacidade subscrição de eventos de alterações sobre processos/instâncias. Para normalizar os dados internos do sistema de execução, referentes à gestão de processos WS-BPEL, é necessário especificar um formato uniforme de representação desta informação, em forma de recursos, utilizando uma das tecnologias apresentadas. A capacidade de manter/actualizar o estado de cada recurso, em função das alterações do estado da instância durante a sua execução, constitui outro objectivo. A gestão/monitorização das instâncias é efectuada através de uma interface programática e um método de acesso uniformes para disponibilizar o acesso aos recursos (que contêm a informação de gestão dos processos).

Relativamente à demonstração de resultados e validação dos conceitos da solução, os objectivos incluem a definição de um processo WS-BPEL, colocação do processo num sistema de execução, visualização da representação da definição do processo em recurso, instanciação/execução do processo e desenvolvimento de uma aplicação de gestão/monitorização que unifique a gestão de processos de negócio.

3.3. Análise de requisitos

A arquitectura proposta para a solução de gestão integrada responde a um conjunto de requisitos funcionais e não funcionais do sistema, considerando os objectivos propostos.

Requisitos funcionais

Identificam-se os seguintes requisitos funcionais do sistema de gestão:

- Deve encapsular a forma de acesso e comunicação específica de cada sistema de execução;
- Deve armazenar a informação de gestão/monitorização dos processos num formato uniforme e transversal aos sistemas de execução;
- Deve gerir a actualização da informação referida no ponto anterior de acordo com a especificidade das interfaces programáticas de cada sistema de execução;
- Deve disponibilizar a informação de gestão/monitorização, no referido formato uniforme, para a aplicação de gestão unificada, através de normas *web*;
- Deve permitir efectuar a gestão/monitorização de cada processo nos diversos sistemas de execução.

Requisitos não funcionais

Identificam-se os seguintes requisitos não funcionais do sistema de gestão:

- Utilizar um sistema de execução que seja compatível com a especificação WS-BPEL, possua capacidade de registar um receptor de eventos de execução e disponibilize uma interface programática de gestão/monitorização;
- Representar cada processo existente no sistema de execução sob a forma de um recurso (entidade lógica para armazenar informação de forma persistente), baseado na sua definição e num formato uniforme, com informação adicional de gestão/monitorização;
- Criar um novo recurso por cada instância de um processo que é iniciada (cópia da representação gerada no ponto anterior para processo da instância);
- Agir como receptor de eventos do sistema de execução, para actualizar o recurso relativo à instância do processo cujo estado foi modificado;
- Disponibilizar uma interface programática de gestão, através de um serviço que permita aceder aos recursos através de uma norma *web*;

3.4. Arquitectura da solução

A arquitectura da solução é constituída por um conjunto de sistemas na resposta aos objectivos e requisitos definidos para a implementação da abordagem de solução considerada. A Figura 19 ilustra a arquitectura proposta.

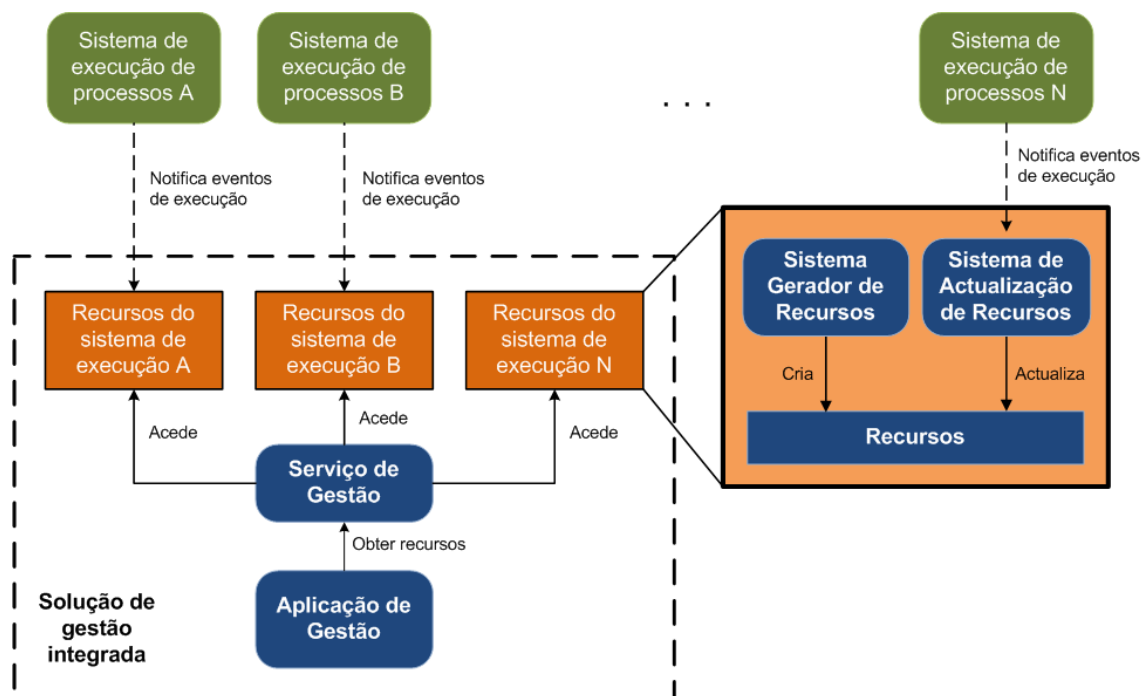


Figura 19 – Arquitectura da solução proposta

A figura ilustra, de forma abstracta, a interacção entre os diversos sistemas envolvidos na solução de gestão integrada. Destes, apenas os sistemas de execução de processos não são constituintes da implementação deste projecto. Os sistemas descritos nos pontos seguintes constituem os componentes desenvolvidos para o protótipo demonstrador, para os quais é efectuada uma breve descrição das suas funcionalidades:

- ✓ **SGR – Sistema Gerador de Recursos**
 - Responsável pela geração da representação de um processo em recurso, com informação de gestão/monitorização. A representação gerada baseia-se na definição do processo e respeita a especificação de um formato uniforme e independente dos sistemas de execução.
- ✓ **SAR – Sistema de Actualização de Recursos**
 - Responsável pela criação/actualização dos recursos, em função das alterações ocorridas no estado das instâncias dos processos. Estas

alterações são notificadas pelo sistema de execução através de interfaces programáticas.

✓ **SG – Serviço de Gestão**

- Serviço que permite o acesso uniforme aos recursos associados às instâncias dos processos em execução e, conseqüentemente, à informação de gestão/monitorização de cada uma.

✓ **Aplicação de Gestão**

- Aplicação que permite visualizar os recursos representativos das instâncias dos processos, nos diversos sistemas de execução, bem como a respectiva informação de gestão de cada uma.

A implementação destes sistemas é abordada no capítulo “4. Implementação” onde são descritos os requisitos e detalhes de implementação da solução.

3.5. Estratégia de solução

A estratégia elaborada visa responder aos objectivos propostos para o desenvolvimento da hipótese de solução considerada e foca-se no desenvolvimento dos adaptadores ilustrados na Figura 18. Estes adaptadores são responsáveis por mapear a definição dos processos para uma representação uniforme na forma de recursos, gerir a actualização dos recursos, em função da execução dos processos e disponibilizar uma forma uniforme de acesso à informação de gestão/monitorização. Esta estratégia inclui a utilização da tecnologia WSRF (WS-Resource Framework) para efectuar uma gestão baseada em recursos.

A Tabela 4 descreve os actores envolvidos nesta solução.

Actor	Descrição
Sistema de execução	Executor de processos de negócio WS-BPEL
Gestor de processos	Gere e monitoriza a execução dos processos de negócio

Tabela 4 – Descrição dos actores do sistema

A Figura 20 contextualiza os actores e sistemas intervenientes na estratégia de solução proposta.

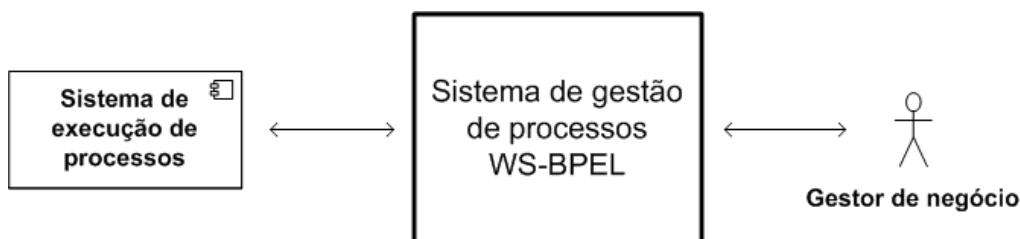


Figura 20 – Diagrama de contexto da solução

Os subcapítulos seguintes descrevem a estratégia elaborada para desenvolvimento dos subsistemas apresentados na arquitectura da solução.

3.5.1. Representação de processos WS-BPEL através de recursos

A definição de um processo WS-BPEL não contempla a persistência da informação de gestão/monitorização da execução de cada instância do processo, por motivos de independência e abstracção do sistema onde este será executado. Aliada a este facto, a especificidade do modelo de dados interno de cada sistema de execução revela a necessidade de criar um formato uniforme para armazenamento e troca da referida

informação de gestão. Segundo a hipótese de solução considerada, ilustrada na Figura 18 do capítulo “3.1 Hipóteses consideradas”, a camada de integração existente entre a aplicação de gestão e o sistema de execução é responsável pela gestão desta uniformização.

A solução para uniformizar a informação de gestão/monitorização dos processos WS-BPEL em cada sistema de execução é representar cada instância sob a forma de um recurso, segundo o quadro de desenvolvimento WSRF e o artigo de referência [1]. A definição WS-BPEL de cada processo inserido no sistema de execução é analisada e, com base nesta, é criado, dinamicamente, o respectivo recurso. As propriedades do recurso estão relacionadas com os elementos XML constituintes do processo e contêm atributos adicionais, inexistentes na norma WS-BPEL, para armazenar a informação de gestão/monitorização.

Esta representação é baseada na definição WS-BPEL [3] do processo e pretende ser genérica, portátil, legível e extensível para ser transversal aos sistemas de execução e independente dos seus modelos de dados internos.

A OASIS, na especificação “Web Services Resource 1.2” [10], define “recurso” como sendo uma entidade lógica com as seguintes características:

- Ser identificável (possuir um identificador único num domínio);
- Ter zero ou mais propriedades;
- Pode ter ciclo de vida.

A solução proposta adopta esta definição: cada recurso é uma entidade com as características supracitadas e concretizada num ficheiro XML. Para além do facto desta linguagem ser a base para a definição de um processo WS-BPEL, é um formato de referência portátil, pesquisável e extensível. Num quadro de evolução para o suporte persistente de informação, com características de atomicidade, consistência, isolamento e durabilidade (propriedades ACID), propõe-se a utilização de um sistema de gestão de base de dados.

Cada ficheiro XML representa o documento de propriedades de um recurso (*WS-ResourceProperties Document*), segundo a especificação WSRF e é identificado univocamente no domínio do sistema de execução, no qual o processo existe. O seu nome é composto por um conjunto de componentes, associadas à informação do processo e da instância, tal como se exemplifica:

- Nome do processo: *LoanRequest*
- Identificador do processo: *LoanRequest-1*

- Identificador da instância do processo: 9801
- Nome do recurso: *pN_LoanRequest_pld_LoanRequest-1_plnslId_9801.xml*

A geração do ficheiro contempla o mapeamento de cada elemento WS-BPEL, existente na definição de um processo, para o elemento correspondente no recurso (por exemplo, o elemento WS-BPEL *sequence* é associado a um elemento *bpelSequence*). No elemento correspondente, são definidos atributos específicos para armazenar a informação de gestão/monitorização. Esta associação é realizada num ficheiro XML *Schema* que permite definir a estrutura de um recurso, nomeadamente, o conjunto de elementos que este pode ter, sem explicitar a sua ordem.

Esta abordagem segue o conceito da definição de um processo WS-BPEL que pode ser constituído pelos diversos elementos da sua especificação sem uma ordenação previamente definida [1], [6]. A Figura 21 ilustra a solução adoptada.



Figura 21 – Geração de recurso para representar um processo WS-BPEL

A metodologia adoptada contempla a representação de cada processo existente sob a forma de recurso, originando um ficheiro XML (RecursoA.xml). Este é criado com base na definição do processo (ProcessoA.bpel) e no ficheiro XML Schema (ResourceSchema.xsd) onde é especificado o modelo dos elementos constituintes do recurso. Cada instância do processo é representada através de uma cópia do respectivo ficheiro recurso (RecursoA_Instância_1.xml). A evolução da execução de uma instância provoca alterações no seu estado, reflectidas no estado das propriedades do recurso associado.

A tabela seguinte apresenta as propriedades de um recurso (associadas a elementos WS-BPEL [3]), com os seus atributos para suporte da informação de gestão/monitorização associada a cada processo.

Propriedade	Atributos
<i>bpelProcess</i>	<i>resourcePropertyId; name; processId; displayName; targetNamespace; exitOnStandardFault; executionMode; continueOnError</i>
<i>bpelInstanceInfo</i>	<i>resourcePropertyId; instanceId; processName; status; dateStarted; dateLastActive</i>
<i>bpelVariable</i>	<i>resourcePropertyId; name; type; state; value</i>
<i>bpelSequence</i>	<i>resourcePropertyId; name; dateStarted; dateEnded; status</i>
<i>bpelReceive</i>	<i>resourcePropertyId; name; dateStarted; dateEnded; status; createInstance; operation; partnerLink; portType; variable</i>
<i>bpelAssign</i>	<i>resourcePropertyId; name; dateStarted; dateEnded; status; validate</i>
<i>bpelReply</i>	<i>resourcePropertyId; name; dateStarted; dateEnded; status; operation; partnerLink; portType; variable</i>

Tabela 5 - Propriedades de um recurso e valores possíveis

Esta metodologia difere da estratégia de solução adoptada no artigo de referência “A Management Framework for WS-BPEL” [1]. Os autores utilizam as seguintes regras de mapeamento da definição do processo para recurso:

1. Cada elemento XML com uma cardinalidade superior a um é mapeado para um recurso, ou seja, todos os elementos XML que ocorram múltiplas vezes dentro do seu elemento pai são mapeados para recursos;
2. Cada elemento que contenha elementos com uma cardinalidade superior a um são mapeados para recursos.

A Figura 22, extraída do mesmo artigo, exemplifica a representação do processo *HelloWorld*, parcialmente ilustrado na parte direita da figura, em recurso, permitindo observar a aplicação das regras apresentadas.

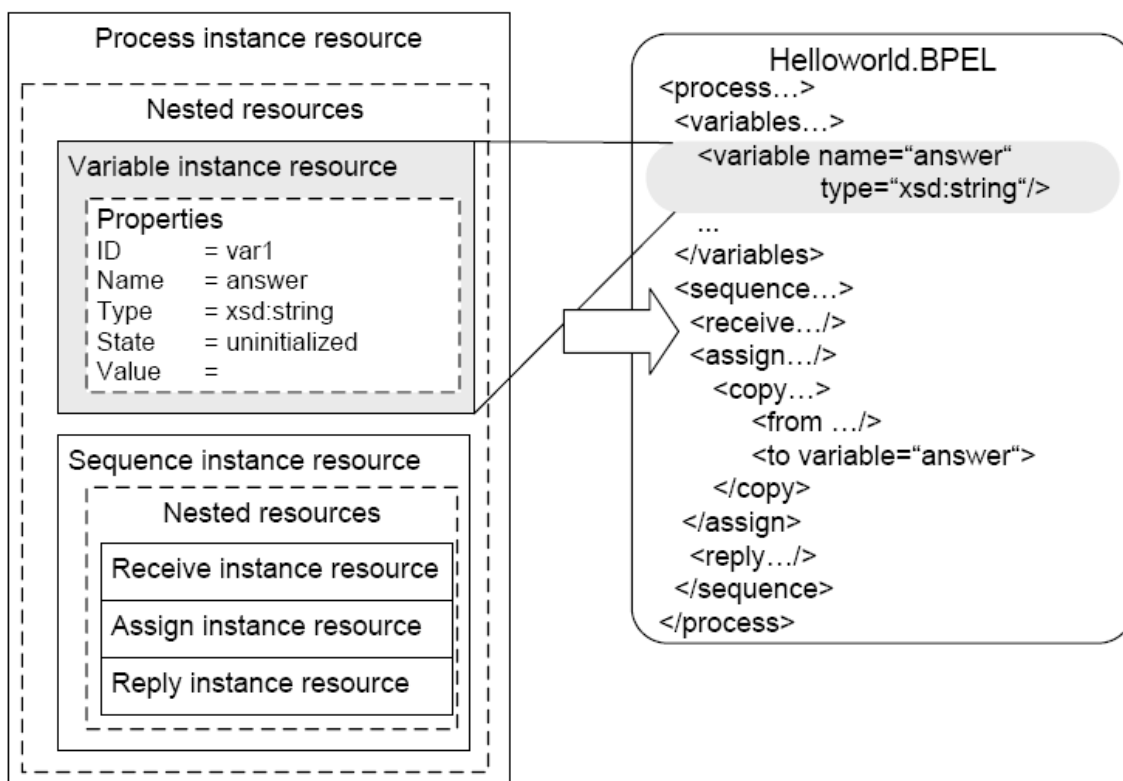


Figura 22 - Processo *HelloWorld* e representação em recurso [1]

A diferença reside no facto dos autores criarem recursos com granularidade ao nível dos elementos constituintes do processo, face à criação de um único recurso onde as suas propriedades representam os elementos do processo. Esta última estratégia simplifica o desenvolvimento da solução pela redução do número de recursos que é necessário gerir para cada processo existente no sistema de execução. Aliado a este facto, os autores do artigo de referência mantêm os recursos no sistema de execução, alterando a sua implementação. A estratégia aqui proposta visa manter a implementação do sistema de execução inalterada, de acordo com a abordagem de solução considerada.

3.5.2. Actualização das propriedades dos recursos

A informação de gestão/monitorização de uma instância de um processo, iniciada num sistema de execução, é mantida no respectivo ficheiro recurso. Como referido, esta informação é armazenada no exterior do sistema de execução, pelo que as propriedades do recurso necessitam de ser actualizadas durante a sua execução.

A solução de implementação proposta é utilizar o padrão *Observer* para criar um receptor dos eventos (*listener*) suportados pelo sistema de execução. A solução depende da capacidade de cada sistema de execução em notificar as alterações ocorridas no estado da instância. Por sua vez, a dependência do sistema de execução, neste aspecto específico, condiciona o nível de detalhe, no que respeita a informação de gestão, que um recurso pode armazenar, ainda que o modelo que define a assinatura das propriedades de um recurso seja genérico.

Uma solução alternativa seria interrogar, de forma repetida, o sistema de execução acerca de alterações no estado das suas instâncias. Esta solução de *poling* produz um acréscimo no número de acessos a cada sistema de execução e o aumento do tempo de propagação das alterações do estado de uma instância para as propriedades do recurso associado. Esta solução não foi considerada pelas desvantagens enunciadas.

A Figura 23 ilustra a solução adoptada, onde se pode verificar a presença de um receptor de eventos de execução. Quando ocorre uma notificação, o receptor é responsável por reunir a informação de gestão/monitorização relevante sobre o estado da instância, que foi alterada, e actualizar o recurso representativo da mesma.

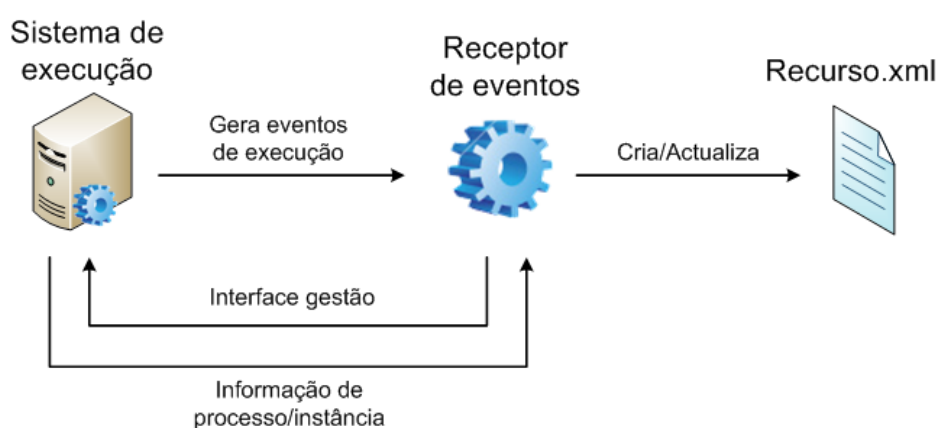


Figura 23 – Diagrama de contexto para actualização de recurso

A informação adicional sobre o estado da instância, que não seja disponibilizada nos objectos de suporte ao evento gerado, é obtida através de outras interfaces programáticas do sistema de execução.

3.5.3. Interface de acesso aos recursos

Considerando a metodologia apresentada, para efectuar a gestão integrada da execução dos processos baseada em recursos, é necessário definir a interface que permite o acesso, por parte da aplicação de gestão/monitorização de processos, às propriedades dos recursos.

Segundo o quadro de especificações WSRF, a forma de acesso aos recursos é efectuada através da utilização de *Web Services*. Os *Web Services* são, por norma, sem estado, não existindo relação entre invocações. Esta característica permite que sejam adequados à comunicação entre sistemas heterogéneos, executados em múltiplas plataformas, fracamente acoplados, sem necessidade de dependência de contexto ou “memória” entre invocações [49]. No entanto, também limita o que pode ser realizado com esta tecnologia, relativamente à necessidade de manter estado. O conjunto de normas WSRF especifica as operações que um *Web Service* pode implementar para que possua estado, recorrendo à utilização de um recurso, segundo a definição anteriormente apresentada [6]. Cada recurso contém um identificador único para que, em cada interacção com o *Web Service*, seja apenas necessário indicar qual o recurso a utilizar para a respectiva operação. A Figura 24 ilustra este conceito, através da invocação de uma operação sobre o *Web Service*, indicando qual o recurso a utilizar.

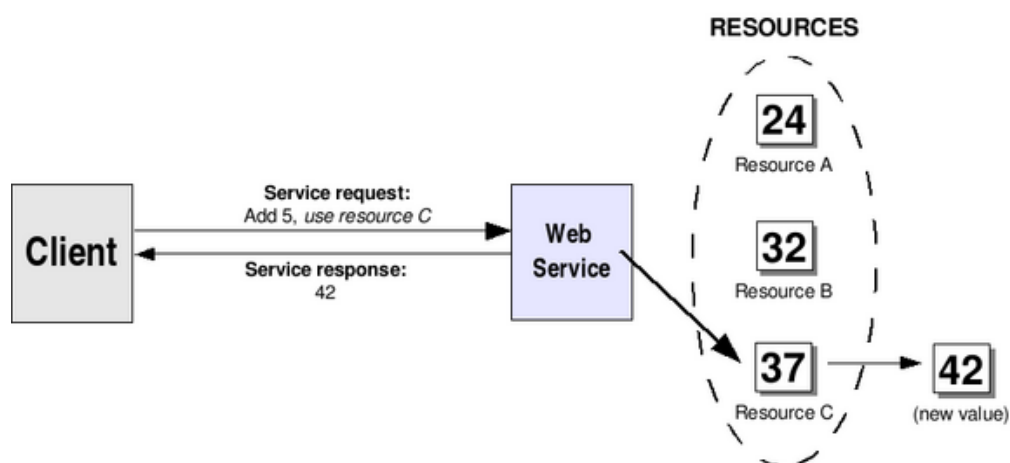


Figura 24 – Manutenção de estado em recursos [9]

Esta abordagem permite conjugar a utilização de uma tecnologia de referência (*Web Service*) para aceder aos recursos, através de operações definidas no quadro WSRF. Este procedimento é efectuado com identificação do recurso, simplificando o sistema que utiliza o serviço e abstraindo-o dos detalhes de implementação/gestão dos recursos.

Segundo a abordagem proposta no quadro de desenvolvimento WSRF, cada recurso, que representa uma instância de um processo, está associada a um *Web Service*, de acordo com o conceito de WS-Resource. A Figura 25 ilustra a relação entre o *Web Service* e os recursos representativos dos processos, aproximando o conceito de WS-Resource presente no quadro WSRF.

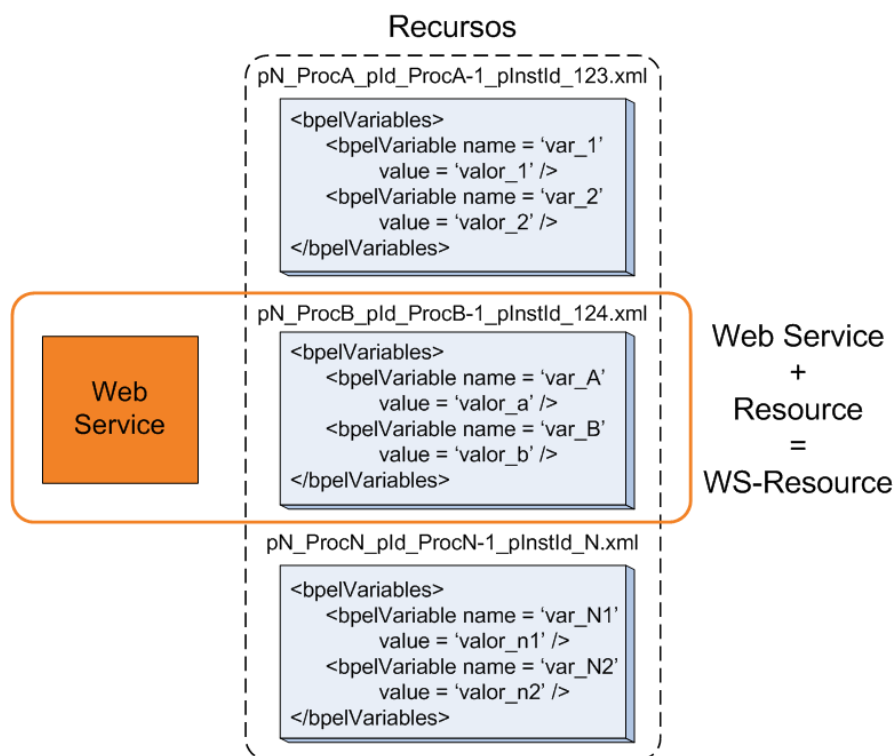


Figura 25 – Relação entre *Web Service* e Recurso

Para concretizar esta abordagem, é necessário ter o conhecimento exacto das propriedades do recurso antes da criação do *Web Service*. Não é viável implementar esta solução, para uma gestão de processos WS-BPEL, porque não é possível ter conhecimento, antes da definição do processo ser disponibilizada, de quais as propriedades que cada recurso terá. Cada processo WS-BPEL tem a sua definição, que está dependente do contexto do negócio e inclui mais ou menos elementos constituintes da especificação WS-BPEL [3]. Estes elementos são, dinamicamente, mapeados para propriedades de um recurso. Desta forma, a estratégia de solução adoptada considera os conceitos teóricos do quadro WSRF, para implementar a representação e acesso aos recursos, mas utiliza apenas um *Web Service* para lhes aceder. Este, por sua vez, acede aos recursos, utilizando um identificador que é passado como parâmetro nas respectivas operações.

O acesso aos dados, presentes no documento de propriedades do recurso, é realizado através de interrogações XPath, usufruindo do seu formato XML, incluindo sempre o

seu identificador. Para obter o documento de propriedades do recurso é retornado directamente o conteúdo do ficheiro XML. Para obter somente a informação de uma determinada propriedade, é aplicada uma expressão XPath, segundo a metodologia apresentada: “obter o valor do elemento XML cujo identificador é X”.

A aplicação de gestão/monitorização invoca operações do serviço para obter o documento de propriedades do recurso ou apenas a informação de uma propriedade, identificando qual o recurso pretendido. O *Web Service* transforma cada pedido numa interrogação XPath sobre o respectivo documento de propriedades do recurso, retornando a informação requerida.

3.5.4. Arquitectura detalhada

A arquitectura detalhada da solução permite acrescentar detalhe à associação entre os sistemas, enunciados no capítulo “3.4 Arquitectura da solução”, que suportam a implementação das funcionalidades, segundo a estratégia apresentada.

A Figura 26 ilustra o diagrama desta associação, detalhando a arquitectura e ilustrando as relações entre os diversos sistemas. É possível identificar o sistema de execução de processos WS-BPEL, o **Sistema Gerador de Recursos (SGR)**, o **Sistema de Actualização de Recursos (SAR)**, o **Serviço de Gestão (SG)**, o repositório de recursos e a **Aplicação de Gestão (AG)**.

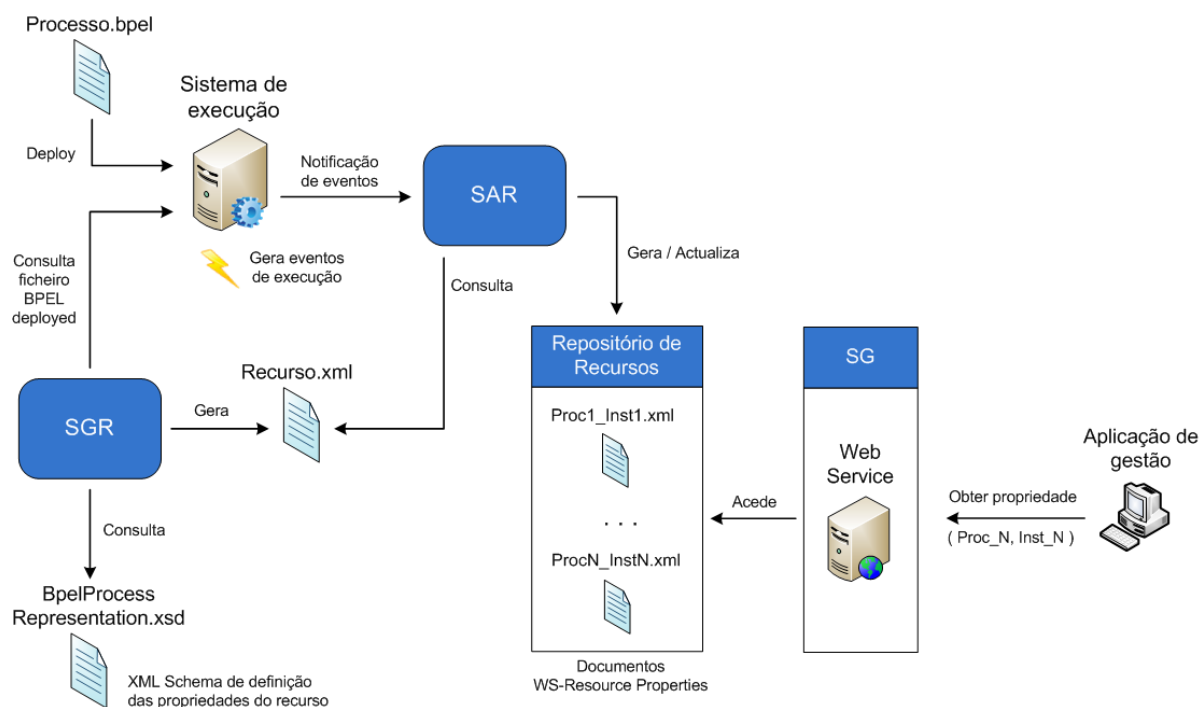


Figura 26 – Arquitectura detalhada da solução

A definição do processo WS-BPEL, existente no sistema de execução, é utilizada pelo sistema SGR, conjuntamente com o ficheiro que especifica o modelo de um recurso (*XML Schema*), para criar a representação do processo WS-BPEL sob a forma de recurso.

O sistema SAR é responsável por actualizar as propriedades dos recursos, em função dos eventos ocorridos sobre as instâncias, com a informação de gestão/monitorização. Esta informação é obtida através das interfaces programáticas de eventos e gestão do sistema de execução.

A aplicação de gestão unificada utiliza o *Web Service* do Serviço de Gestão (SG) para invocar as operações que permitem obter os valores das propriedades dos recursos.

O capítulo “4. Implementação” apresenta os requisitos e os detalhes de implementação de cada um dos sistemas desenvolvidos e ilustrados na Figura 26. Estes suportam a estratégia de solução proposta para um ambiente de gestão integrada de sistemas de execução de processos WS-BPEL.

3.6. Síntese

Os fornecedores de sistemas de execução adoptam uma estratégia, segundo a qual a informação de gestão/monitorização dos processos de negócio é representada em formatos específicos de cada sistema, culminando em interfaces de gestão proprietárias.

As abordagens propostas para resolução deste problema consideram a implementação de adaptadores para encapsular as funcionalidades de integração. Através destes adaptadores, é possível uniformizar o método de acesso e formato da informação trocada entre sistemas de execução e gestão de processos WS-BPEL. A uniformização contribui para a unificação da gestão de processos WS-BPEL, em execução em diferentes sistemas, numa única aplicação de gestão integrada.

Uma das abordagens de solução considera que os sistemas de execução de processos WS-BPEL são responsáveis por uniformizar o acesso e formato da informação de gestão de processos.

A abordagem alternativa considera a realização de adaptadores na aplicação de gestão, sendo esta a responsável por conter a lógica de integração com cada sistema de execução, permitindo manter estes últimos inalterados. É uma abordagem conservadora, de compromisso com o modelo de integração adoptado actualmente em algumas organizações, constituído por sucessivas camadas de integração entre sistemas, desenvolvidas à medida das necessidades. Esta hipótese foi adoptada para o desenvolvimento do projecto no sentido de explorar as suas limitações.

A estratégia delineada contempla a utilização de um modelo de gestão baseado em recursos, para gerir/monitorizar a execução de processos WS-BPEL. Cada instância é representada sob a forma de um recurso, segundo o quadro WSRF, com propriedades para armazenamento da informação de gestão. A actualização dos recursos é efectuada em função dos eventos de execução ocorridos sobre a instância. O acesso às suas propriedades é realizado mediante uma interface programática não proprietária, desenvolvida para o efeito, associada a uma aplicação de gestão integrada que permite visualizar e gerir os processos em execução. Os recursos são mantidos/actualizados fora dos sistemas de execução, permitindo a utilização de especificações não proprietárias para criação e acesso à sua informação.

A arquitectura proposta contempla o desenvolvimento de um conjunto de sistemas que se integram segundo a abordagem proposta e respondem aos requisitos da gestão de processos WS-BPEL baseada em recursos.

4. Implementação

Este capítulo tem por objectivo descrever os detalhes de implementação da arquitectura proposta, na concretização dos objectivos enunciados no capítulo anterior.

4.1. Requisitos de implementação

A implementação e validação da solução requerem a utilização de algumas ferramentas de desenvolvimento. A listagem seguinte apresenta as respectivas ferramentas:

- Netbeans IDE, versão 6.5 (Java)
 - Desenvolvimento do Sistema Gerador de Recursos (SGR) e Sistema de Actualização de Recursos (SAR);
 - *Web Services RiskDeciderWS* e *LoanMakerWS* que compõem o processo WS-BPEL de demonstração (consultar capítulo “5.1 Definição de processo WS-BPEL”).
- Eclipse SDK, versão 3.3.2 (Java)
 - Instanciação de processos WS-BPEL no Apache ODE;
- *Plug-in* Eclipse BPEL Designer (WS-BPEL) [36]
 - Desenvolvimento de processos WS-BPEL;
- Microsoft Visual Studio 2008 (C#)
 - Instanciação de processos WS-BPEL;
 - Elaboração do Serviço de Gestão (SG);
 - Desenvolvimento da Aplicação de Gestão Integrada (AGI) de processos WS-BPEL.

Aliado aos sistemas apresentados, é necessário criar um ambiente de execução para todos os sistemas da solução:

- Sistema operativo Windows XP;
- Tomcat Server, versão 6.0.18, com cliente de *Web Services Axis2* [28];
- Sistema de execução de processos Apache ODE, versão 1.3.3 [29];
- Sun Java 6, *update* 13;
- Microsoft Framework .Net 2.0;
- Microsoft IIS Server, versão 6.0.

A escolha do sistema de execução de processos WS-BPEL revela-se crítica e decisiva. As funcionalidades e limitações deste sistema influenciam a validação e aceitabilidade da solução, na medida em que os sistemas implementados dependem da capacidade que este possui em expor informação de gestão/monitorização dos processos que executa.

Os critérios de escolha incluem a utilização de um sistema de execução *open source*, com funcionalidade de subscrição de eventos de execução e interface programática de gestão/monitorização específica do fornecedor. Após a análise sobre as características que os fornecedores indicam acerca dos sistemas de execução enunciados no capítulo “2.4 Execução de processos WS-BPEL”, a escolha recaiu sobre o sistema Apache ODE, por responder aos critérios enunciados. Segue-se uma descrição do sistema, onde são apresentados os argumentos justificativos da decisão.

4.1.1. Apache ODE - Sistema de execução de processos WS-BPEL

O Apache ODE (*Apache Orchestration Director Engine*) é um sistema de execução de processos *open source*, compatível com a especificação WS-BPEL 2.0 [3] da OASIS e desenvolvido em Java, por parte da comunidade de referência Apache Software Foundation [27].

Este sistema apresenta algumas funcionalidades, para além da compatibilidade com a norma WS-BPEL da OASIS, entre as quais se destacam o suporte a duas camadas de integração, através de Axis2 e da norma JBI (*Java Business Integration*)³, detalhados na arquitectura, *binding* HTTP WSDL, subscrição/notificação de eventos de execução dos processos e uma interface programática de gestão/monitorização de processos e instâncias. As duas últimas características foram determinantes na decisão da sua adopção para a realização do projecto.

Arquitectura

Os componentes da arquitectura do ODE incluem o compilador BPEL, o ambiente de execução, os objectos DAO (*ODE Data Access Objects*) e as camadas de integração. As relações entre estes componentes estão ilustradas na Figura 27.

³ <http://jcp.org/aboutJava/communityprocess/final/jsr208/index.html>

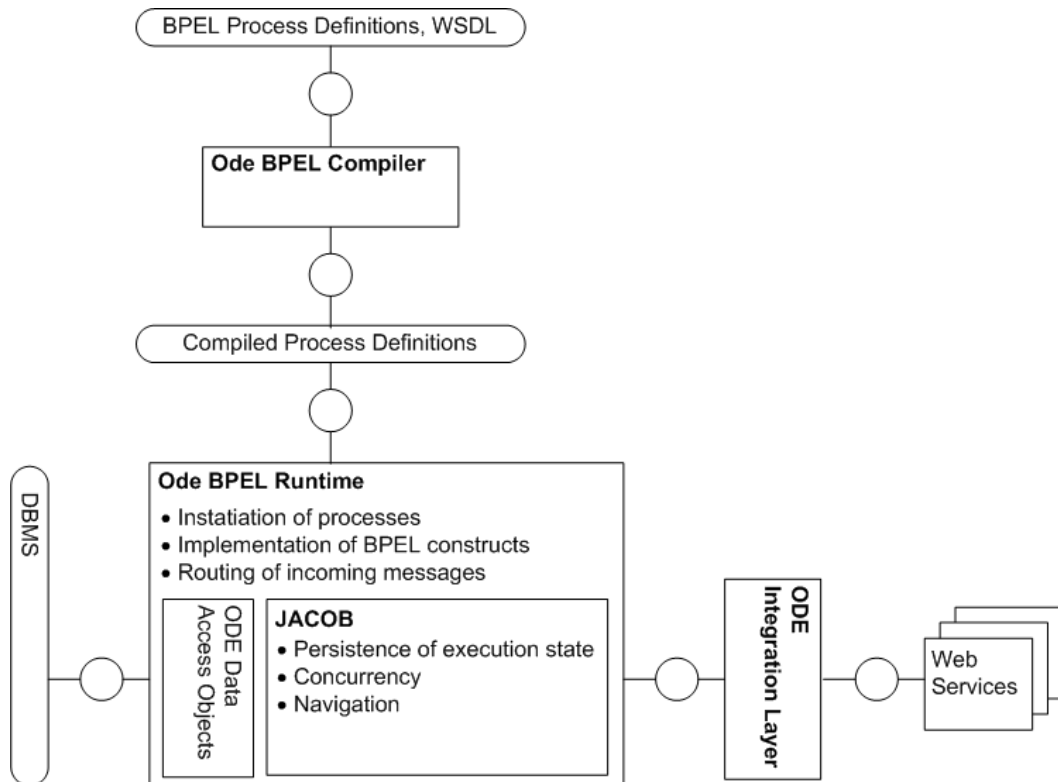


Figura 27 – Arquitectura do orquestrador Apache ODE [30]

O compilador converte os documentos BPEL com a definição do processo num formato executável pelo ambiente de execução. Este executa-os e utiliza, como forma de persistência, uma base de dados acessível via DAOs. O processo é executado no contexto de uma camada de integração que se liga ao exterior através das suas interfaces (por exemplo, *Web Services*) [29].

A camada de integração disponibiliza canais de comunicação exteriores para o motor de execução. O Axis2, motor de execução de *Web Services*, cumpre este objectivo utilizando as suas bibliotecas para permitir a comunicação via *Web Services*. A especificação JBI, desenvolvida sobre JCP (*Java Community Process*) para uma aproximação de implementação a uma arquitectura orientada a serviços (SOA), em complemento com o Apache ServiceMix (*enterprise service bus*), cumpre o mesmo objectivo ligando o sistema de execução ao *bus* de mensagens JBI [30]. O âmbito deste projecto não abrange o estudo destes componentes.

Neste projecto foi utilizado o servidor applicacional Apache Tomcat 6.0 [28], incorporando o motor de *Web Services* Axis2 como ambiente de execução do sistema Apache ODE, permitindo que o Apache ODE e os seus processos sejam vistos como *Web Services*. Esta característica permite invocar o serviço para iniciar, terminar e gerir processos através de SOAP/HTTP.

Interface de gestão/monitorização

O Apache ODE disponibiliza uma interface programática de gestão/monitorização que permite obter informações sobre processos e instâncias em execução ou já terminadas, valores de variáveis, entre outras informações [31]. Dispõe de duas interfaces, *ProcessManagement* e *InstanceManagement*, documentadas no respectivo *Javadoc* [32], cuja implementação está acessível via *Web Services*, na distribuição baseada em Axis2.

Este sistema de execução disponibiliza as funcionalidades mínimas necessárias para a sua adopção neste projecto. Entre estas constam a execução de processos WS-BPEL, capacidade de subscrição de eventos e obtenção de informação de gestão/monitorização através de interfaces programáticas proprietárias [31]. Apesar destas interfaces permitirem obter diversas informações sobre as instâncias, a característica de serem proprietárias causa uma dependência relativamente às aplicações de gestão/monitorização. Estas necessitam de se adequar à informação que é possível obter, bem como ao seu formato, características específicas de cada fabricante. Este problema carece de uma solução baseada em interfaces abertas, não proprietárias, que permitam normalizar a gestão dos processos executados em diferentes sistemas de execução.

4.2. Protótipo demonstrador

O protótipo demonstrador concretiza a arquitectura proposta e a estratégia de solução elaboradas para a gestão integrada de sistemas de execução de processos WS-BPEL, segundo a abordagem considerada. Este protótipo é composto pelo Sistema Gerador de Recursos (SGR), Sistema de Actualização de Recursos (SAR), Serviço de Gestão (SG) e Aplicação de Gestão (AG), descritos nos subcapítulos seguintes.

4.2.1. Sistema Gerador de Recursos (SGR)

O Sistema Gerador de Recursos (SGR) representa parte da camada de integração com os sistemas de execução de processos, sendo responsável pela geração da representação dos processos, definidos em WS-BPEL, para recursos. A Figura 28 destaca este sistema na arquitectura proposta.

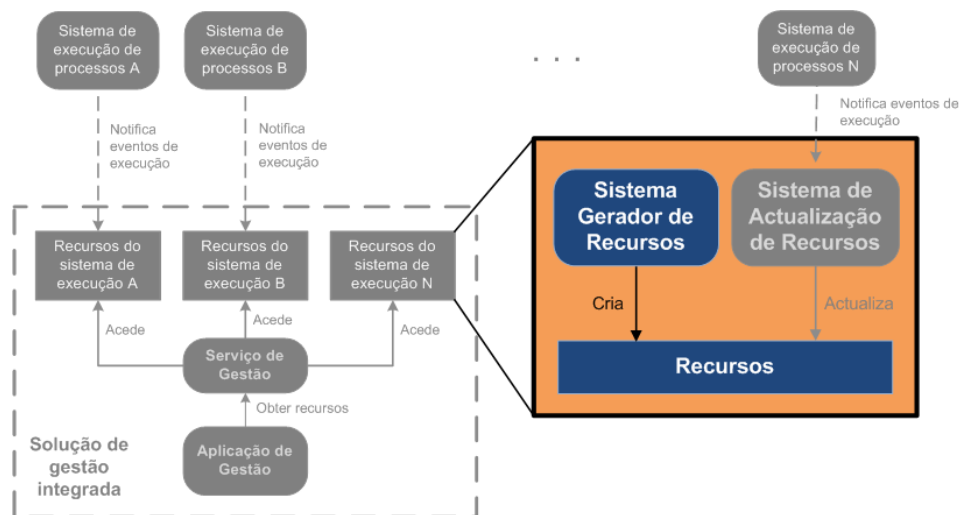


Figura 28 – Sistema Gerador de Recursos na arquitectura proposta

O sistema é composto por uma aplicação Java *desktop*, baseada na *Swing Application Framework* (JSR-296) [41] disponibilizando uma *interface* gráfica para controlo de execução. A Figura 29 ilustra esta interface, onde é possível observar dois botões que permitem, respectivamente, iniciar e suspender a geração automática de recursos.

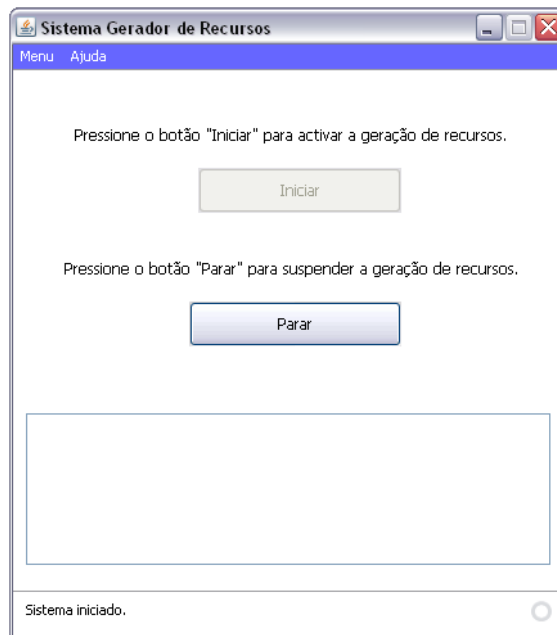


Figura 29 – Interface gráfica do Sistema Gerador de Recursos

O diagrama de casos de utilização, apresentado na Figura 30, ilustra os actores e funcionalidades presentes neste sistema.

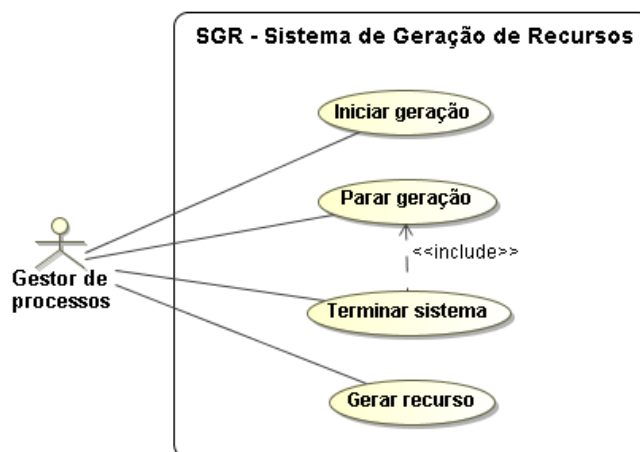


Figura 30 – Diagrama de casos de utilização do Sistema Gerador de Recursos

A descrição de cada caso de utilização está disponível para consulta no Apêndice G.

O sistema é constituído por componentes que interagem entre si com o objectivo de proporcionar a geração automática de recursos. A geração é despoletada após a colocação do ficheiro, com a definição do processo WS-BPEL, no sistema de execução. A Figura 31 ilustra o diagrama de colaboração/comunicação das componentes constituintes.

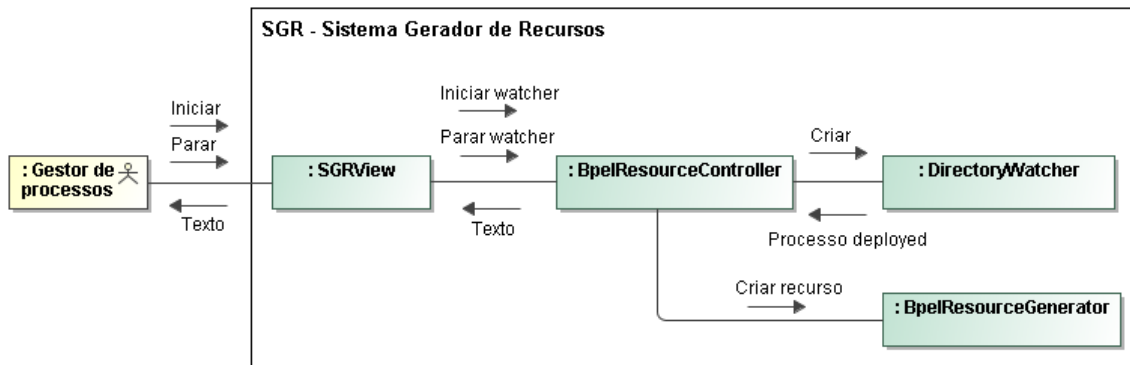


Figura 31 – Diagrama de comunicação do sistema SGR

Quando o sistema é executado é construída uma nova instância do tipo *BpelResourceController*, responsável pela coordenação da geração de recursos.

A acção de “iniciar o sistema” despoleta a criação de uma instância da classe *DirectoryWatcher*, cujo objectivo é actuar como receptor de processos existentes no sistema de execução. Esta interacção é ilustrada no diagrama de sequência seguinte.

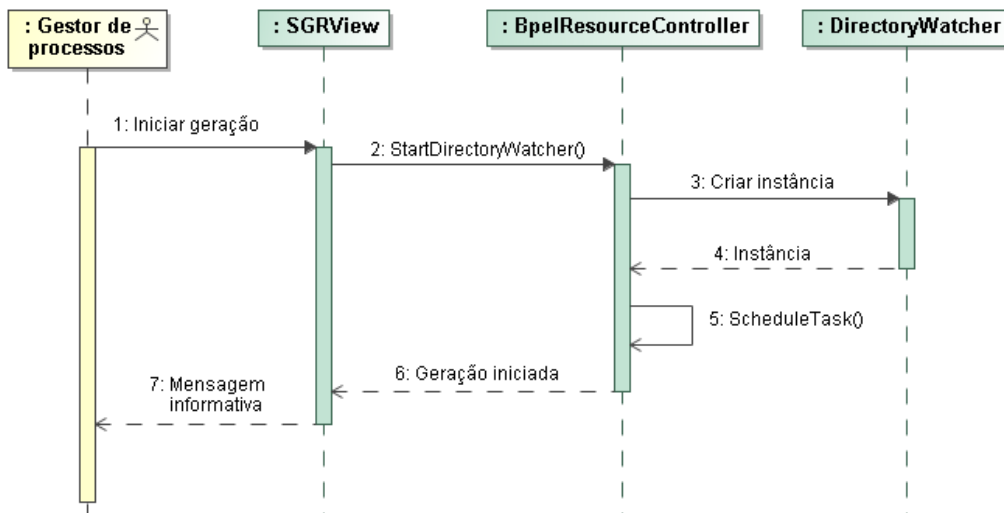


Figura 32 – Diagrama de sequência do caso de utilização *Iniciar geração*

A acção de “terminar a geração” é semelhante à acção de “iniciar”, sendo ilustrada no diagrama presente na Figura 33.

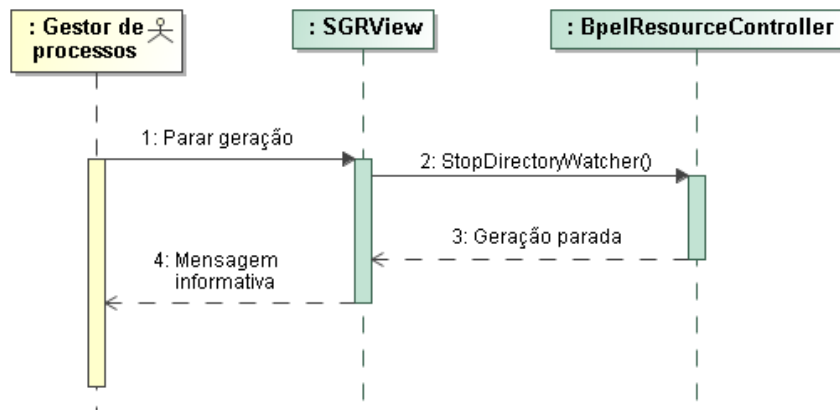


Figura 33 – Diagrama de sequência do caso de utilização *Parar geração*

A execução de um processo no sistema Apache ODE inicia-se com a compilação da sua definição, após inserida no sistema de execução, para o seu modelo interno de objectos. Caso não sejam detectados erros na sua definição, passa a estar automaticamente disponível para ser instanciado. Por inserção entende-se a colocação do ficheiro com a definição do processo numa pasta do sistema de execução, ficando disponível para ser, posteriormente, instanciado [29].

A solução aqui proposta, relativamente à geração de recursos após a inserção do processo, é genérica para qualquer sistema de execução, com características idênticas às do Apache ODE, e baseia-se numa abordagem semelhante ao modo de funcionamento de inserção de processos, acima descrito. Para tal, foi utilizada uma *TimerTask* (classe *DirectoryWatcher*) que actua como receptor de ficheiros inseridos na pasta específica do sistema de execução. Este receptor possui um filtro para a extensão “.bpel” e por cada novo ficheiro presente na pasta é invocado um método de *callback* na classe *BpelResourceGenerator* (classe que instancia um objecto do tipo *DirectoryWatcher*). Este método, denominado *onProcessDeployed*, efectua a geração de um recurso, recorrendo ao método *GenerateResource* da classe *BpelResourceGenerator*. A Figura 34 ilustra o respectivo diagrama de sequência destas acções.

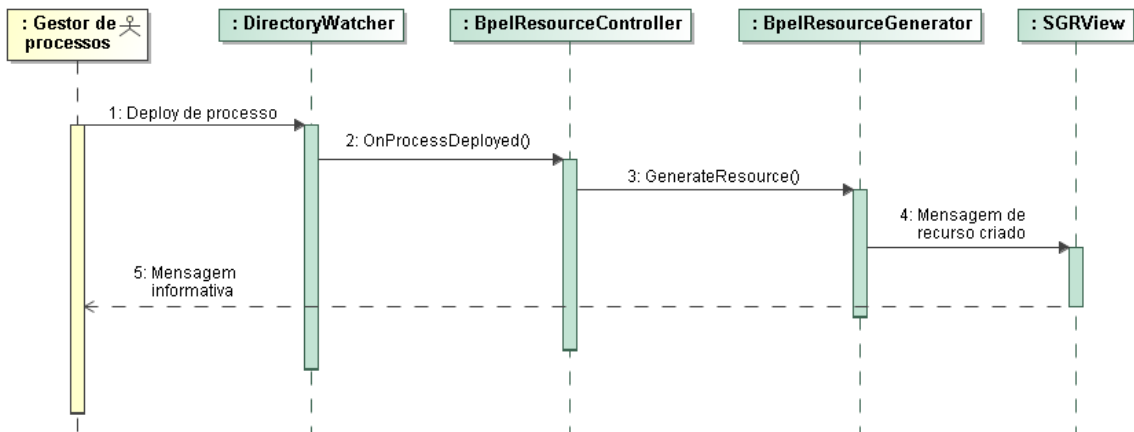


Figura 34 – Diagrama de sequência do caso de utilização *Gerar recurso*

Relativamente ao término do sistema, a Figura 35 ilustra o diagrama de sequência do respectivo caso de utilização.

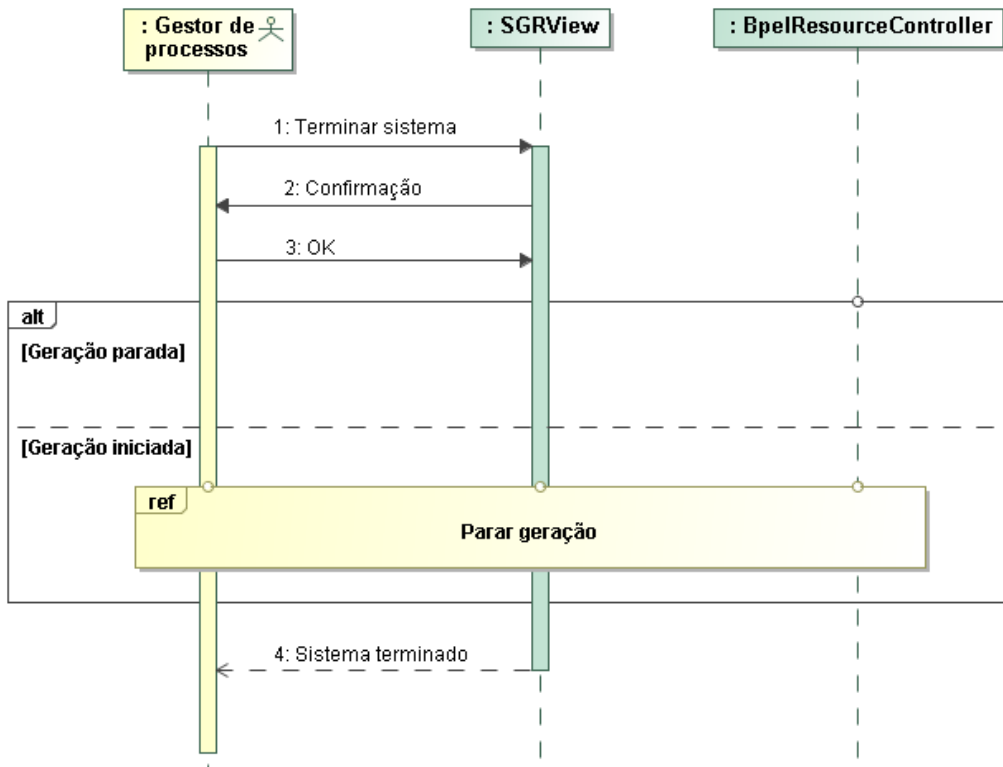


Figura 35 – Diagrama de sequência do caso de utilização *Terminar sistema*

A Figura 36 ilustra o diagrama de classes deste sistema, onde estão incluídas, entre outras, as classes referidas nos diagramas de sequência apresentados e que constituem a organização estática do sistema.

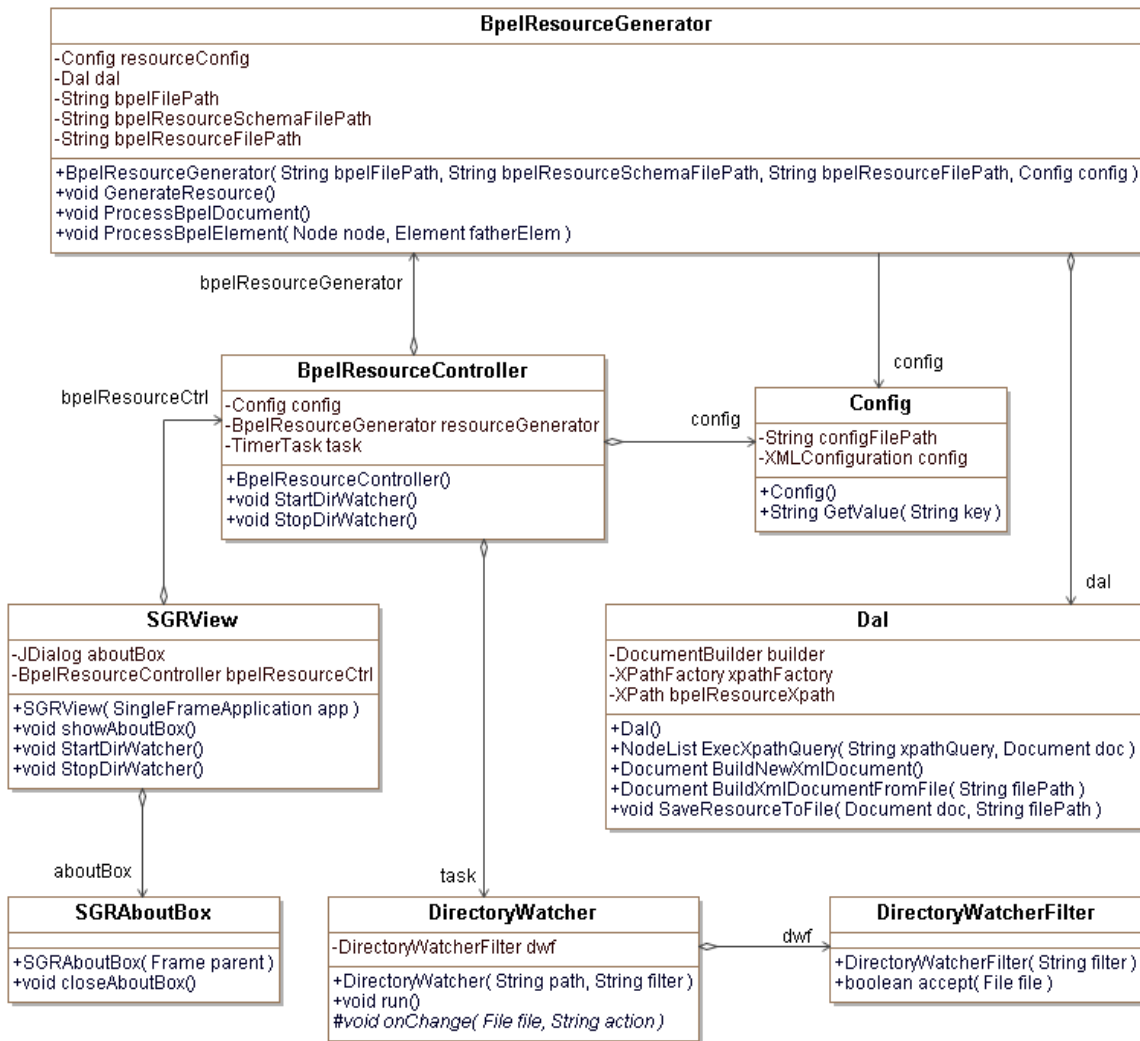


Figura 36 – Diagrama de classes do Sistema Gerador de Recursos

Segundo a estratégia de solução considerada, o estado do recurso é mantido num ficheiro XML, denominado por *WS-Resource Properties Document* (de acordo com o conceito existente no quadro WSRF). Os seus elementos constituem as propriedades do recurso e permitem armazenar a informação de gestão/monitorização das instâncias. A geração deste ficheiro é dinâmica, ou seja, perante a inserção de um processo WS-BPEL é gerado um recurso em função dos elementos WS-BPEL existentes na sua definição. Para tal, este sistema utiliza dois ficheiros auxiliares para concretizar os seus objectivos: “BpelProcessRepresentation.xsd”, ficheiro de associação entre elementos WS-BPEL e propriedades de recurso e “config.xml”, ficheiro de configuração do sistema. Estes ficheiros estão disponíveis para consulta, respectivamente, nos Apêndices A e B.

A classe *Dal*, ilustrada no diagrama de classes, é responsável pela criação do ficheiro XML do recurso e pelo acesso aos dados dos ficheiros WS-BPEL (definição do processo) e XML Schema “*BpelProcessRepresentation.xsd*”.

A Figura 37 permite ilustrar o diagrama de contexto destes elementos no sistema de geração de recursos.

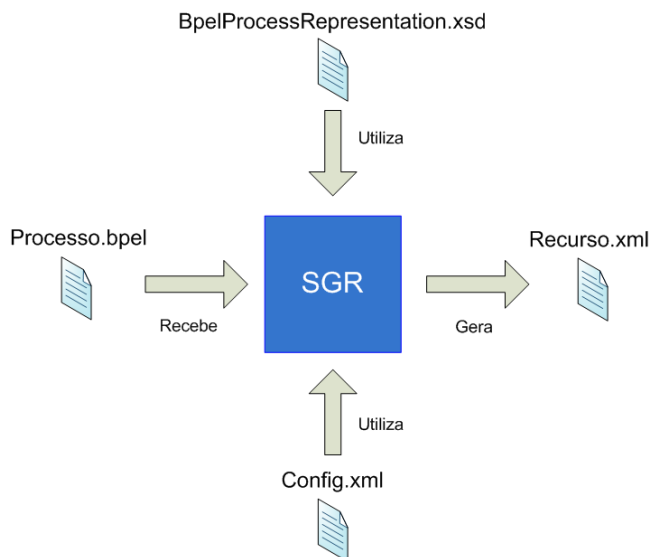


Figura 37 – Diagrama de contexto da geração de um recurso

Associação de elementos WS-BPEL a propriedades do recurso

O ficheiro “*BpelProcessRepresentation.xsd*” é o ficheiro XML *Schema* onde são definidos os elementos e atributos constituintes das propriedades do recurso (*WS-Resource Properties*). Este ficheiro especifica o modelo de um recurso. As propriedades são representações de alguns elementos XML do processo, definidos na especificação WS-BPEL [3], com alguns dos seus atributos e outros adicionais para suporte à informação de gestão/monitorização. A Figura 38 ilustra o modelo de dados dos elementos XML deste *schema*.

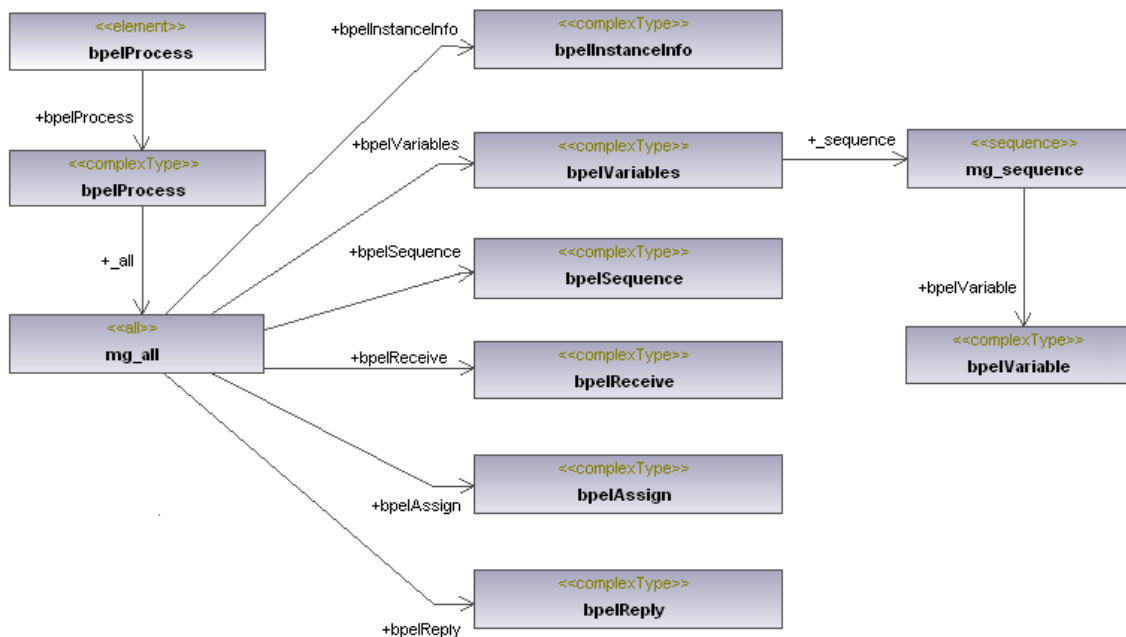


Figura 38 – Modelo de dados dos elementos XML do *Schema* do recurso

Um elemento da especificação WS-BPEL tem associado um elemento XML neste *schema*, com mais ou menos atributos, em função da informação de gestão/monitorização que se pretende manter acerca do elemento (por exemplo, o atributo “valor” tem expressividade na representação de um elemento WS-BPEL *variable* mas não na representação de um elemento WS-BPEL *sequence*). A tabela seguinte ilustra uma noção do estado das propriedades do recurso e possíveis valores para alguns elementos da definição WS-BPEL, segundo o artigo de referência “A Management Framework for WS-BPEL” [1].

Instance Resource	Notion of the state resource property and possible values
process instance	specifies the current state of the process instance. One of {instantiated, running, suspended, terminated, faulted, complete}
activities	Specifies the current state of the activity. One of {initial, inactive, ready, dead path, executing, waiting, terminated, faulted, complete}
loops	Specifies the current state of a looping structured activity (e.g. while, repeatUntil, forEach) One of {initial, inactive, ready, dead path, executing, waiting, terminated, faulted, complete, iteration_complete, check_condition }
scopes	Specifies the current state of the scope. One of {initial, inactive, ready, dead path, executing, event_handling, waiting, termination_handler, terminated, complete, fault_handling, compensation_executing, compensated }
links	Specifies the current state of a link. One of {undetermined, ready, evaluated, true, false}

Tabela 6 - Valores das propriedades dos recursos [1]

Para o protótipo demonstrador da solução, apenas foram considerados alguns dos elementos da especificação WS-BPEL na geração dos recursos. Os elementos seleccionados permitem demonstrar a implementação da abordagem proposta, sem necessidade de representar todos os elementos da respectiva especificação. A Tabela 7 apresenta a associação entre os elementos WS-BPEL considerados e a respectiva denominação em propriedade de recurso. A nomenclatura das propriedades é constituída pelo nome do elemento WS-BPEL prefixado com a palavra “bpel”.

Elemento WS-BPEL	Propriedade do recurso
Process	<i>bpelProcess</i>
Variables	<i>bpelVariables</i>
Variable	<i>bpelVariable</i>
Sequence	<i>bpelSequence</i>
Receive	<i>bpelReceive</i>
Assign	<i>bpelAssign</i>
Reply	<i>bpelReply</i>

Tabela 7 – Associação dos elementos WS-BPEL a propriedades de recurso

A Figura 39 ilustra os atributos da entidade *bpelVariable*, definidos no ficheiro XML Schema “BpelProcessRepresentation.xsd”.






<<complexType>> bpelVariable	
	<<attribute>> resourcePropertyId:string=empty
	<<attribute>> name:string=empty
	<<attribute>> type:string=empty
	<<attribute>> state:string=uninitialized
	<<attribute>> value:string=empty

Figura 39 – Definição do elemento XML *bpelVariable*

Salvaguarda-se uma excepção relativamente às propriedades definidas no modelo de um recurso: a especificação WS-BPEL não contém o elemento *InstanceInfo*. No entanto, foi adicionado um elemento *bpelInstanceInfo*, como filho do elemento raiz *bpelProcess*, com o objectivo de representar e armazenar a informação da respectiva instância para a qual o recurso foi gerado. A Figura 40 ilustra este elemento, com os atributos para armazenamento da informação de monitorização.

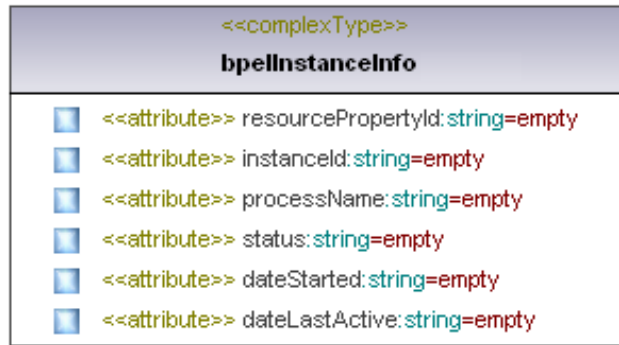


Figura 40 – Definição do elemento XML *bpellInstanceInfo*

Configuração do sistema

O ficheiro “config.xml”, disponível no Apêndice B, contém as configurações de execução do sistema SGR. Entre estas, constam a localização do ficheiro “BpelProcessRepresentation.xsd” e as associações entre nome dos elementos WS-BPEL e o nome das suas representações em propriedades de recurso. A classe *Config*, ilustrada no diagrama de classes, encapsula os métodos de acesso a este ficheiro. Foi utilizada a classe *XMLConfiguration*, presente no *package org.apache.commons.configuration*, para simplificar a escrita do código que permite obter o valor de um elemento XML. Após a sua instanciação, é utilizada a seguinte sintaxe: `variável_de_instância.getString("nome_do_elemento_xml")`.

Geração do recurso

A geração do recurso inclui um interpretador dos elementos WS-BPEL presentes no ficheiro com a definição do processo. Para cada elemento WS-BPEL analisado, existe uma verificação de correspondência a uma propriedade de recurso. Caso a correspondência exista, é utilizada a interrogação XPath associada a essa propriedade (presente no ficheiro de configuração “Config.xml”) para interrogar o ficheiro de *schema* “BpelProcessRepresentation.xsd”. Esta interrogação permite obter os atributos para criar o elemento XML no ficheiro do recurso. O gerador necessita de consultar ambos os ficheiros para gerar dinamicamente os recursos, tal como se ilustra na figura seguinte.

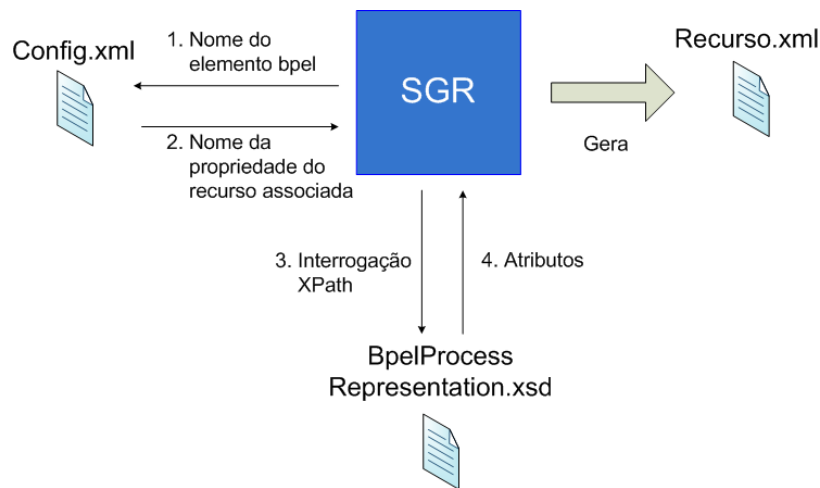


Figura 41 – Diagrama de contexto da geração dinâmica de recursos

O ficheiro XML gerado apresenta uma estrutura de elementos filhos do elemento raiz *bpelProcess*, tal como se encontra disponível para consulta no Apêndice C. Cada elemento contém, obrigatoriamente, um atributo denominado *resourcePropertyId*. Este atributo representa o identificador unívoco da propriedade no recurso. É utilizado pela Aplicação de Gestão e pelo Serviço de Gestão para identificar a propriedade a interrogar.

Extensibilidade da solução

A solução implementada permite adicionar uma nova associação de um elemento WS-BPEL para propriedade de recurso, através da alteração dos ficheiros auxiliares do sistema. No ficheiro “Config.xml”, é necessário criar a associação entre o nome do elemento e a expressão XPath para obter os atributos definidos no modelo do recurso. No ficheiro “BpelProcessRepresentations.xsd”, é necessário incluir a definição do novo elemento XML, que representa a propriedade do recurso, com os respectivos atributos para gestão/monitorização. O sistema SGR gerará, posteriormente, de forma dinâmica, o ficheiro do recurso incluindo a nova propriedade.

O ficheiro gerado constitui a especificação de um processo, sob a forma de recurso, para todas as suas instâncias. Os recursos a serem criados, como consequência das várias instanciações do processo, serão cópias deste ficheiro. Estas cópias são utilizadas pelo sistema SAR, descrito no subcapítulo seguinte.

4.2.2. Sistema de Actualização de Recursos (SAR)

O Sistema de Actualização de Recursos (SAR) é responsável pela actualização dos documentos de propriedades dos recursos, representativos das instâncias dos processos. A Figura 42 ilustra o SAR no contexto na arquitectura proposta.

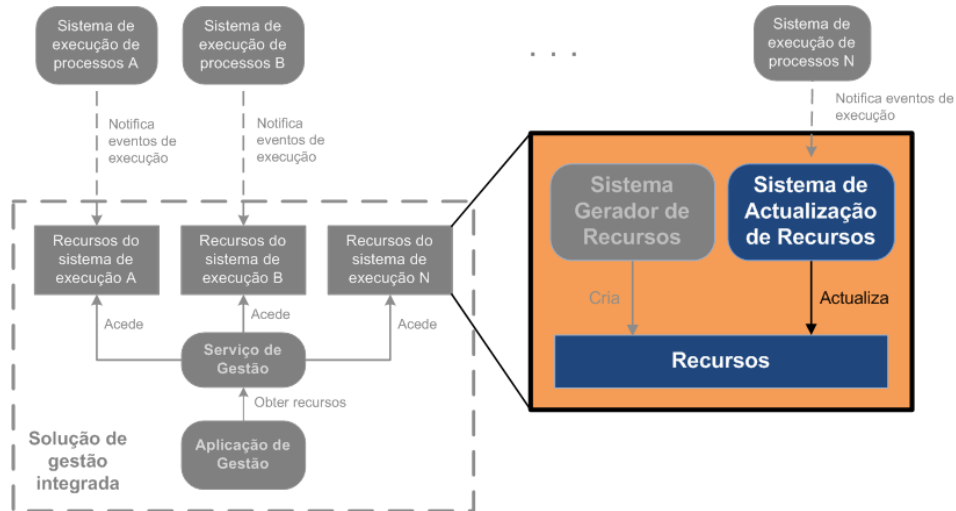


Figura 42 – Sistema de Actualização de Recursos na arquitectura proposta

O sistema actua como receptor de eventos sobre as instâncias dos processos no sistema de execução. Recebe como entrada os ficheiros XML dos recursos, gerados pelo sistema SGR e os eventos do sistema de execução, ocorridos durante a execução das instâncias. O sistema permite criar um ficheiro recurso por cada nova instância ou actualizar a informação de gestão/monitorização das propriedades do recurso, associado à instância em execução, sobre a qual ocorreu um evento. A Figura 43 ilustra o diagrama de contexto deste sistema.

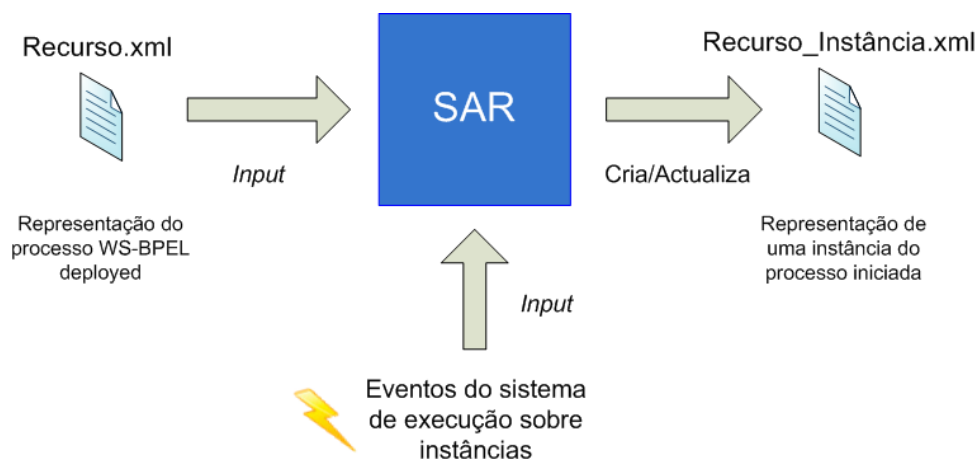


Figura 43 – Diagrama de contexto do Sistema de Actualização de Recursos

Eventos de execução do sistema Apache ODE

A tabela seguinte apresenta os eventos de execução suportados pelo sistema Apache ODE, detalhando o nome, a descrição e tipo de cada um [29].

Event Name	Process/Scope	Description	Type
ActivityEnabledEvent	Scope	An activity is enabled (just before it's started)	activityLifecycle
ActivityDisabledEvent	Scope	An activity is disabled (due to dead path elimination)	activityLifecycle
ActivityExecStartEvent	Scope	An activity starts its execution	activityLifecycle
ActivityExecEndEvent	Scope	An activity execution terminates	activityLifecycle
ActivityFailureEvent	Scope	An activity failed	activityLifecycle
CompensationHandlerRegistered	Scope	A compensation handler gets registered on a scope	scopeHandling
CorrelationMatchEvent	Process	A matching correlation has been found upon reception of a message	correlation
CorrelationNoMatchEvent	Process	No matching correlation has been found upon reception of a message	correlation
CorrelationSetWriteEvent	Scope	A correlation set value has been initialized	dataHandling
ExpressionEvaluationFailedEvent	Scope	The evaluation of an expression failed	dataHandling
ExpressionEvaluationSuccessEvent	Scope	The evaluation of an expression succeeded	dataHandling
NewProcessInstanceEvent	Process	A new process instance is created	instanceLifecycle
PartnerLinkModificationEvent	Scope	A partner link has been modified (a new value has been assigned to it)	dataHandling
ProcessCompletionEvent	Process	A process instance completes	instanceLifecycle
ProcessInstanceStartedEvent	Process	A process instance starts	instanceLifecycle
ProcessInstanceStateChangeEvent	Process	The state of a process instance has changed	instanceLifecycle
ProcessMessageExchangeEvent	Process	A process instance has received a message	instanceLifecycle
ProcessTerminationEvent	Process	A process instance terminates	instanceLifecycle
ScopeCompletionEvent	Scope	A scope completes	scopeHandling
ScopeFaultEvent	Scope	A fault has been produced in a scope	scopeHandling
ScopeStartEvent	Scope	A scope started	scopeHandling
VariableModificationEvent	Scope	The value of a variable has been modified	dataHandling
VariableReadEvent	Scope	The value of a variable has been read	dataHandling

Tabela 8 - Eventos de execução suportados pelo Apache ODE [29]

O protótipo demonstrador, não sendo uma solução totalmente funcional, não inclui suporte a todos os eventos apresentados na tabela anterior. A implementação do receptor de eventos deste sistema suporta o tratamento dos seguintes eventos:

- ✓ *NewProcessInstanceEvent*
- ✓ *ProcessTerminationEvent*
- ✓ *ProcessInstanceStateChangeEvent*
- ✓ *ProcessInstanceStartedEvent*
- ✓ *VariableModificationEvent*
- ✓ *ActivityEnabledEvent*
- ✓ *ActivityExecStartEvent*

- ✓ *ActivityExecEndEvent*
- ✓ *ActivityFailureEvent*

Tratamento de eventos para actualização de recursos

O sistema de execução Apache ODE suporta a publicação/subscrição de eventos relativos à execução das instâncias. Para registar um receptor destes eventos foi necessário desenvolver a classe *OdeEventListener*, a qual implementa a interface *org.apache.ode.bpel.iapi.BpelEventListener*. Após a implementação, foi necessário adicioná-la ao *classpath* do sistema de execução e adicionar uma propriedade no ficheiro “ode-axis2.properties”, tal como se exemplifica na Listagem 1.

```
ode-axis2.event.listeners=sar.OdeBpelEventListener
```

Listagem 1 – Configuração e adição de receptor de eventos no Apache ODE

Para o tratamento dos eventos e conseqüente actualização dos recursos, é necessário recolher, entre outros dados, informações acerca do nome do processo, identificador do processo e identificador da instância, de modo a permitir identificar o recurso (de acordo com a terminologia descrita no capítulo 3.2.1). Estes dados são obtidos através da utilização da interface programática *InstanceManagement*, do sistema Apache ODE [31], que disponibiliza, entre outros, os métodos *listAllInstances* e *getInstanceInfo*.

A Figura 44 ilustra o diagrama de estrutura composta do sistema SAR, relativamente à interacção entre o sistema de execução Apache ODE e o receptor de eventos.

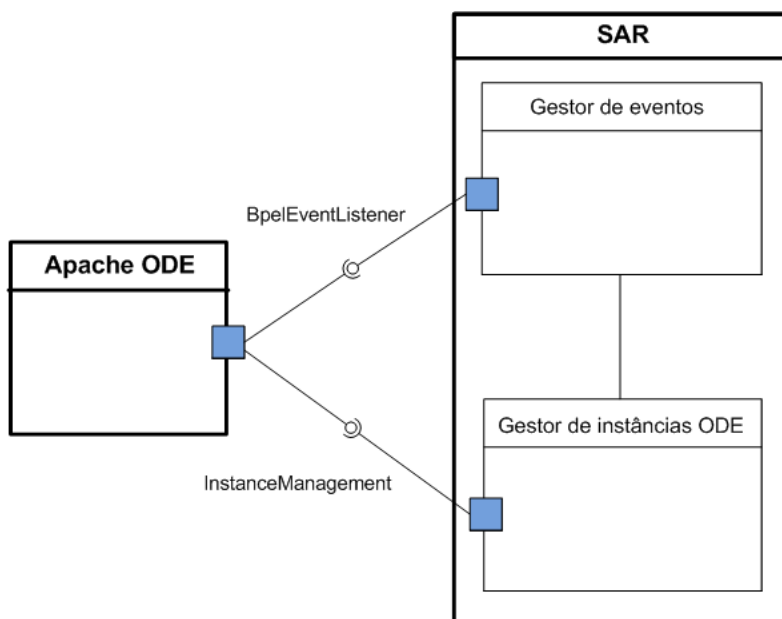


Figura 44 – Diagrama de estrutura composta do SAR

A interface *BpelEventListener* é apresentada na Figura 45.



Figura 45 – Interface *BpelEventListener* do sistema Apache ODE [31]

O método *onEvent* é responsável pela gestão das notificações de eventos do sistema de execução. Neste método está implementado o código que diferencia o tipo de evento gerado e o trata de forma adequada.

A Figura 46 ilustra o diagrama de sequência do SAR, relativamente à criação/actualização de um recurso, após a notificação de um evento ocorrido sobre uma instância.

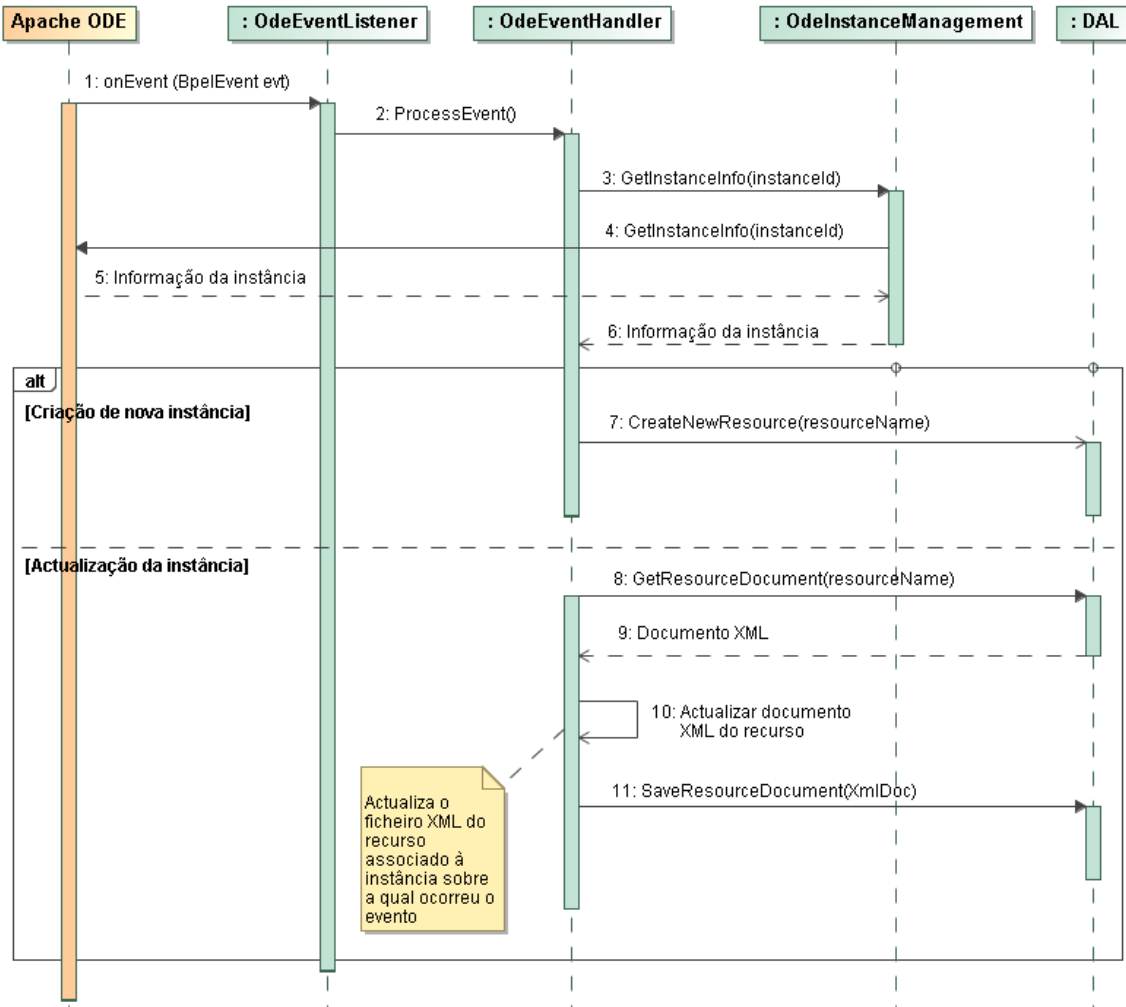


Figura 46 – Diagrama de sequência da criação/actualização de recursos no SAR

A Figura 47 ilustra o diagrama de classes deste sistema, onde é possível observar a hierarquia de classes de tratamento de eventos. Cada um dos tipos de eventos gerados origina a chamada a um método de tratamento, na respectiva classe associada ao evento.

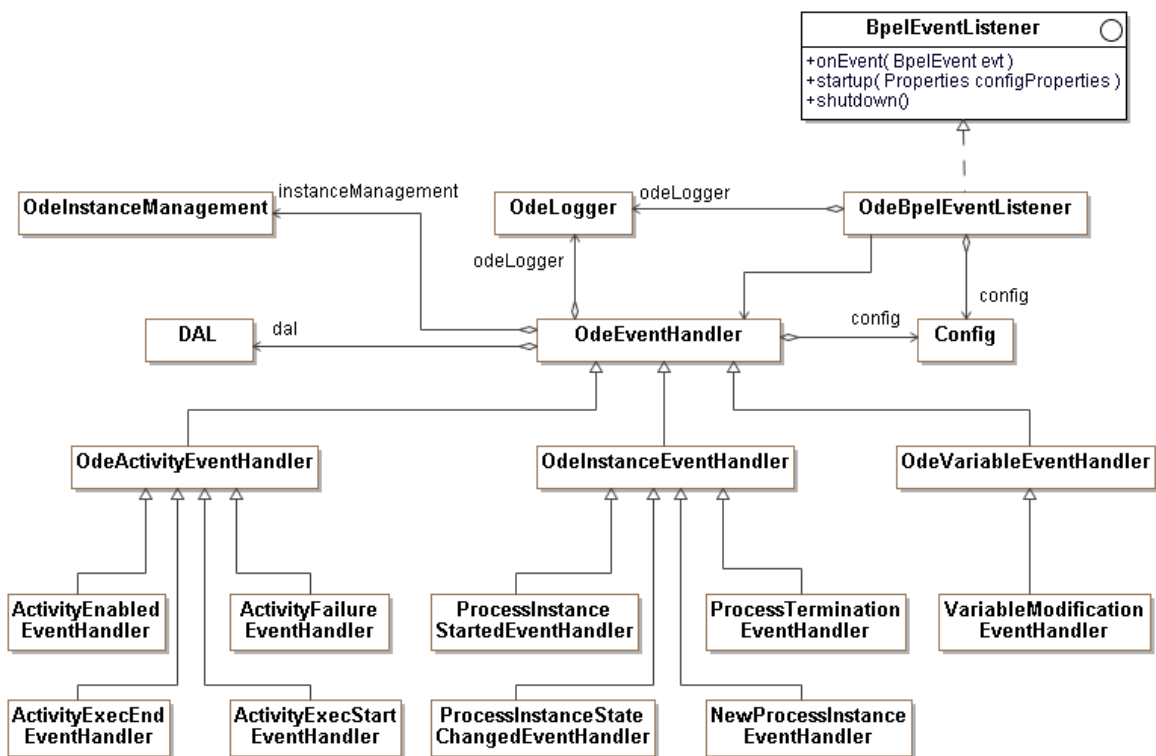


Figura 47 – Diagrama de classes do sistema SAR

A classe base *OdeEventHandler* contém um objecto do tipo DAL, responsável por criar novos ficheiros de recursos e afectar as suas propriedades. Contém ainda um objecto do tipo *OdeInstanceManagement*, que encapsula o acesso à informação da instância, realizado através da interface programática *InstanceManagement* do sistema Apache ODE [31]. A classe auxiliar *OdeLogger* permite efectuar o log, em ficheiro, da evolução de cada operação de processamento de eventos e, caso existam, os seus erros.

O evento *NewProcessInstanceEvent* ocorre quando uma instância é iniciada e tem um objectivo de tratamento distinto dos restantes. A criação de uma nova instância determina também a criação de um novo recurso. A criação é efectuada através da cópia integral do ficheiro XML, representativo da definição do processo WS-BPEL, de acordo com a metodologia apresentada no capítulo “3.5.1 Representação de processos WS-BPEL em recursos”.

O processamento dos restantes eventos é semelhante entre si. O objectivo comum é actualizar o documento de propriedades do recurso, associado à instância sobre a qual ocorreu o evento. O objecto que constitui o parâmetro do método *onEvent*, presente na interface *BpelEventListener*, não possui toda a informação de gestão/monitorização necessária para actualizar um recurso. Devido a esta limitação, é utilizada a interface programática *InstanceManagement*, implementada por um *Web Service* do Apache ODE. Os métodos desta interface permitem obter informações acerca da própria instância, variável ou actividade relacionada com o evento. A invocação dos métodos deste serviço está implementada na classe *OdeInstanceManagement*. Por recomendação do sítio oficial do Apache ODE [29], para simplificação da invocação quando é utilizado o cliente Axis2 [33], é utilizada a classe auxiliar *org.apache.ode.axis2.service.ServiceClientUtil*, presente em *ode-axis2.jar*. O retorno dos métodos deste serviço é efectuado através de XML, pelo que é necessário interpretar a resposta e obter os dados para actualizar as propriedades dos recursos. Segue-se um exemplo do retorno do método *GetInstanceInfo*, invocado para obter informações acerca de uma instância identificada pelo identificador "59402".

```

<axis2ns3:getInstanceInfoResponse
xmlns:axis2ns3="http://www.apache.org/ode/pmapi">
  <instance-info>
    <ns:iid xmlns:ns="http://ns">59402</ns:iid>
    <ns:pid xmlns:ns="http://ns">
      {http://loanrequest}LoanRequestProcess-57
    </ns:pid>
    <ns:process-name xmlns:ns="http://ns"
xmlns:loan="http://loanrequest"> loan:LoanRequestProcess
    </ns:process-name>
    <ns:root-scope xmlns:ns="http://ns" siid="59453" tatus="ACTIVE"
name="__PROCESS_SCOPE:LoanRequestProcess" modelId="3" />
    <ns:status xmlns:ns="http://ns">ACTIVE</ns:status>
    <ns:dt-started xmlns:ns="http://ns">
      2009-09-06T19:20:09.343+01:00
    </ns:dt-started>
    <ns:dt-last-active xmlns:ns="http://ns">
      2009-09-06T19:20:09.406+01:00
    </ns:dt-last-active>
    <ns:event-info xmlns:ns="http://ns" />
    <ns:failures xmlns:ns="http://ns">
      <ns:dt-failure>2009-09-06T19:20:09.406+01:00</ns:dt-failure>
      <ns:count>1</ns:count>
    </ns:failures>
  </instance-info>
</axis2ns3:getInstanceInfoResponse>

```

Listagem 2 – Exemplo de retorno do método *GetInstanceInfo* do *Web Service InstanceManagement* (Apache ODE)

Na Listagem 2 identificam-se informações para actualização do recurso, tais como o nome e identificador do processo, o identificador da instância, o seu estado, as datas em que foi iniciada, terminada ou quando ocorreu uma falha.

A actualização do documento de propriedades de um recurso implica a sua obtenção do repositório de recursos do sistema de execução. O ficheiro a actualizar inclui, na sua denominação, o nome e identificador do processo e o identificador da instância, segundo a metodologia apresentada no capítulo “3.5.1 Actualização das propriedades dos recursos”. Aliado à identificação do ficheiro, é necessário reunir toda a informação de gestão/monitorização da instância ou do elemento WS-BPEL sobre o qual ocorreu o evento. Reunidas as condições anteriores, é necessário obter a propriedade do recurso (elemento XML) para actualizar o valor dos seus atributos, de acordo com o novo estado da instância/elemento WS-BPEL. A obtenção deste elemento é realizada através da aplicação de uma expressão XPath. Estas expressões estão definidas no ficheiro de configuração do sistema SAR, “Config.xml”, disponível no Apêndice E. A Listagem 3 apresenta as expressões XPath para obter as propriedades *bpelInstanceInfo*, *bpelVariable* e *bpelActivity*.

```
<resourceXPathUpdateExpr>
  <updateInstanceInfo>
    //bpelInstanceInfo
  </updateInstanceInfo>
  <updateVariableInfo>
    //bpelVariable[@name="varName"]
  </updateVariableInfo>
  <updateActivityInfo>
    //*[@name="activityName"]
  </updateActivityInfo>
</resourceXPathUpdateExpr>
```

Listagem 3 – Expressões XPath para obter propriedades do recurso a actualizar

Os parâmetros *varName* e *activityName*, presentes nestas expressões, são substituídos, dinamicamente, pelo nome da variável/actividade sobre a qual ocorreu o evento.

Os documentos de propriedades dos recursos, representativos de cada instância iniciada no sistema de execução, são disponibilizados num repositório de recursos associado ao sistema de execução. Este repositório é uma pasta que contém todos os ficheiros XML gerados pelo sistema SAR. O sistema descrito no capítulo seguinte, Serviço de gestão, utiliza o respectivo repositório para responder aos pedidos de gestão/monitorização das instâncias.

4.2.3. Serviço de Gestão (SG)

O Serviço de Gestão (SG) constitui a interface de acesso da aplicação de gestão aos recursos. A Figura 48 evidencia o enquadramento deste sistema na arquitectura proposta.

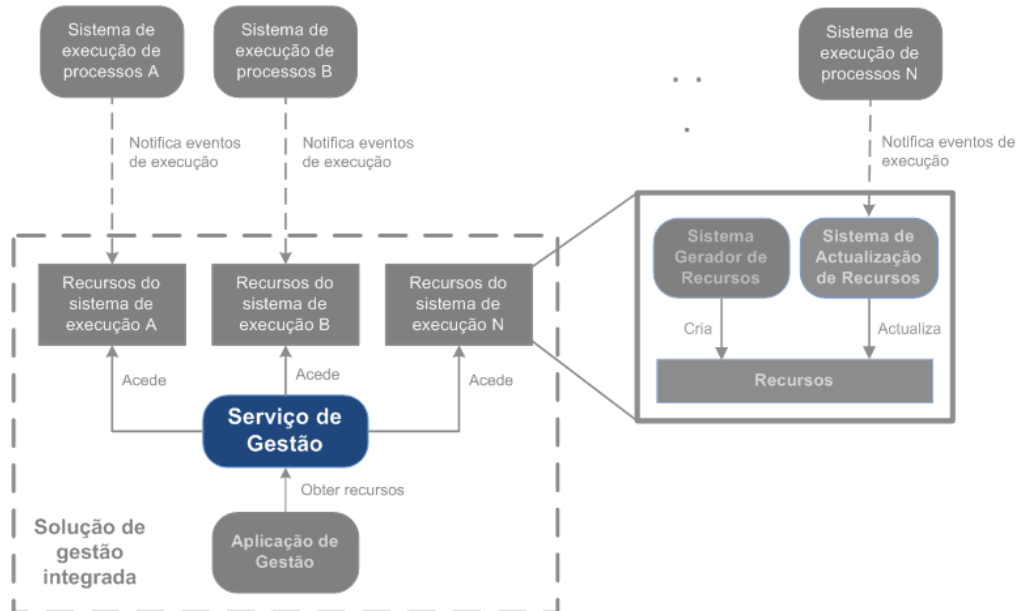


Figura 48 – Sistema de Gestão na arquitectura proposta

O sistema é constituído por uma interface programática de gestão e pela implementação dos métodos da respectiva interface através de um *Web Service*, segundo a estratégia apresentada no capítulo “3.5.3 Interface de acesso aos recursos”.

Requisitos da interface de gestão

A definição da interface programática de gestão foi baseada na análise de um quadro de requisitos funcionais, associados à gestão de processos em sistemas de execução. A listagem seguinte identifica os requisitos funcionais desta interface:

- ✓ Listar sistemas de execução;
- ✓ Listar processos do sistema de execução;
- ✓ Listar instâncias dos processos (recursos);
- ✓ Obter o documento de propriedades de um recurso;
- ✓ Obter uma propriedade de um recurso;
- ✓ Suspender a execução de uma instância;
- ✓ Prosseguir a execução de uma instância;
- ✓ Terminar uma instância.

O conjunto de requisitos apresentado permite elaborar o diagrama de casos de utilização, ilustrado na Figura 49, na perspectiva de um gestor de processos de negócio. A sua descrição está disponível no Apêndice H e é válida para a compreensão do serviço de gestão.

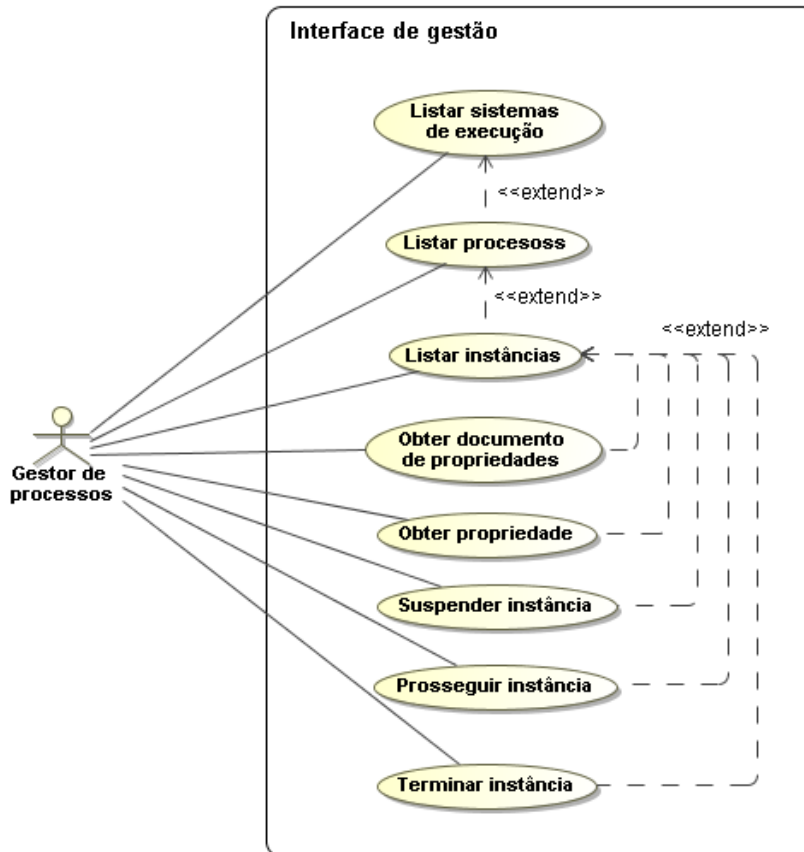


Figura 49 – Diagrama de casos de utilização da interface de gestão

Os casos apresentados constituem parte do diagrama de casos de utilização elaborado para a aplicação de gestão, descrita no subcapítulo seguinte. No diagrama da Figura 49 constam apenas os casos de utilização associados à interface programática de gestão, apresentada neste subcapítulo.

A identificação das funcionalidades de gestão/monitorização e a elaboração dos respectivos casos de utilização, na perspectiva de um gestor de negócio, permitiu definir a interface programática aqui descrita e a implementação dos seus métodos.

Definição da interface programática de gestão

A interface programática *IBpelManagement* especifica a assinatura dos métodos que implementam as funcionalidades identificadas para este sistema. A Figura 50 apresenta a respectiva interface, definida no ficheiro “IBpelManagement.cs”.



Figura 50 – Interface programática do Serviço de Gestão

O termo “orquestrador”, apresentado no capítulo “2.3 Execução de processos WS-BPEL”, é adoptado na nomenclatura de alguns métodos como forma de redução do seu tamanho (*ListOrchestrators* em detrimento de *ListBpelExecutionSystems*).

O método *ListOrchestrators* permite obter a identificação de cada sistema de execução que pode ser gerido através do serviço de gestão. Todos os restantes métodos recebem como parâmetro o identificador do sistema de execução, obtido a partir da invocação do referido método. Esta decisão foi tomada para resolver o problema da localização do repositório de recursos. Através do identificador do sistema de execução, é possível determinar a localização do seu repositório e aceder aos respectivos documentos de propriedades. Esta solução permite centralizar o conhecimento destes identificadores no serviço de gestão, evitando possíveis inconsistências destes dados entre o serviço e as aplicações de gestão. Num quadro de evolução deste protótipo demonstrador, uma abordagem alternativa seria centralizar esta informação num sistema de gestão de bases de dados.

Implementação da interface de gestão

A implementação dos métodos está acessível via *Web Service*, devido à sua adaptabilidade a uma solução que requer fraco acoplamento entre sistemas e de acordo com a abordagem proposta pelo quadro de desenvolvimento WSRF. Como justificação adicional, o sistema Apache ODE disponibiliza o acesso às interfaces de gestão de processos e instâncias, respectivamente *ProcessManagement* e *InstanceManagement*, através de *Web Services* [31].

O *Web Service BpelManagement* implementa os métodos especificados na interface *IBpelManagement*, descrita anteriormente, e é disponibilizado como a forma de acesso uniforme à informação de processos e instâncias (recursos) de um sistema de execução. Este sistema foi desenvolvido utilizando a tecnologia Microsoft .Net, contrariamente aos sistemas anteriores desenvolvidos em Java. Este aspecto demonstra que é possível realizar a integração de sistemas desenvolvidos em tecnologias diferentes através de normas de referência ou especificações abertas.

Os modelos de análise apresentados nas figuras seguintes ilustram a colaboração entre os componentes do serviço de gestão, desenvolvidos para realizar as funcionalidades associadas aos casos de utilização. A elaboração destes modelos considerou a utilização do sistema de execução Apache ODE e as suas interfaces programáticas *ProcessManagement* e *InstanceManagement*.

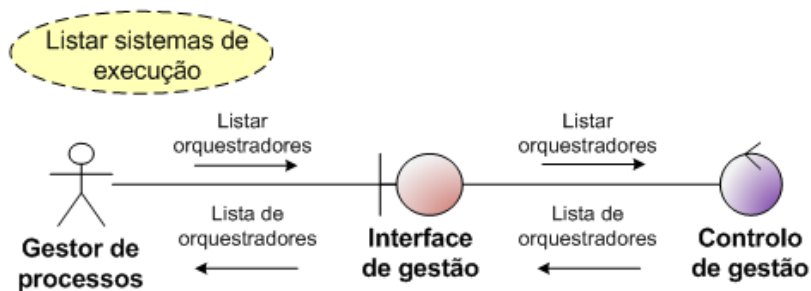


Figura 51 – Modelo de análise do caso de utilização *Listar sistemas de execução*

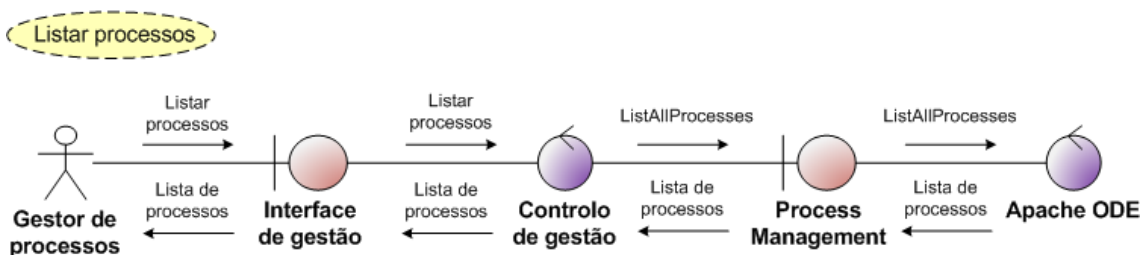


Figura 52 – Modelo de análise do caso de utilização *Listar processos*

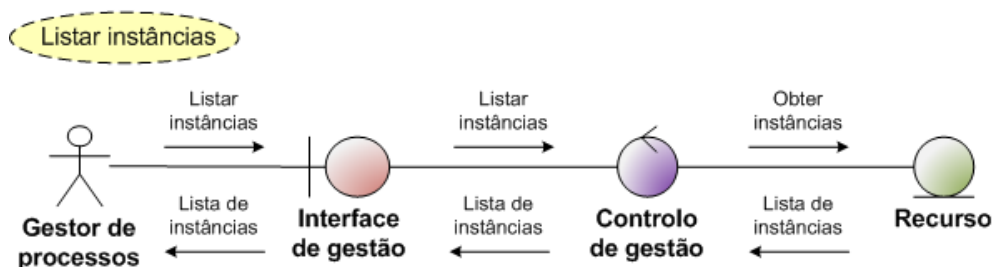


Figura 53 – Modelo de análise do caso de utilização *Listar instâncias*

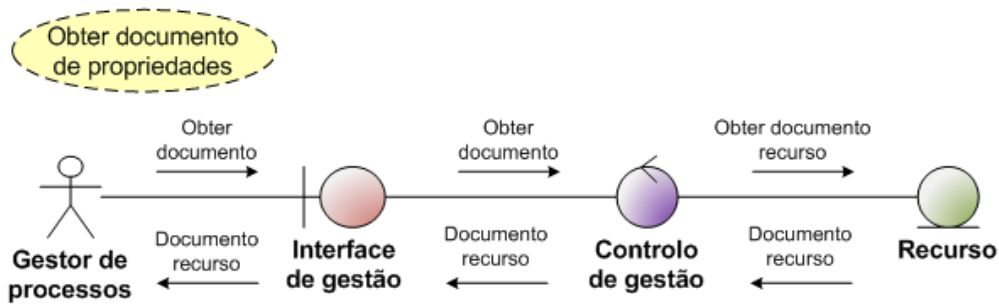


Figura 54 – Modelo de análise do caso *Obter documento de propriedades*

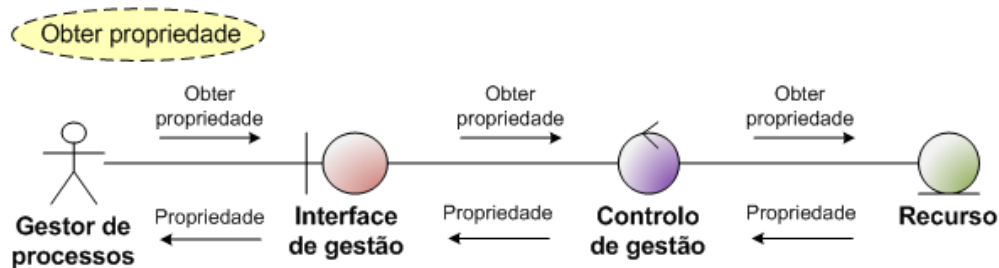


Figura 55 – Modelo de análise do caso de utilização *Obter propriedade*

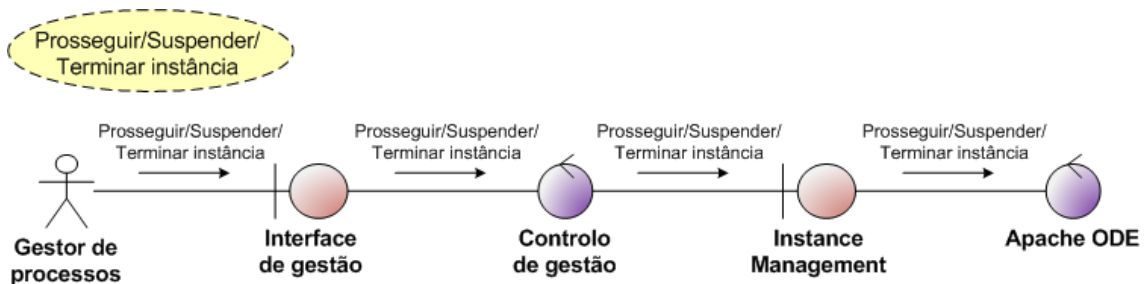


Figura 56 – Modelo de análise dos casos *Prosseguir/Suspender/Terminar instância*

Cada um dos modelos de análise apresentados origina a criação de diagramas de sequência, com as acções para realizar os casos de utilização.

O diagrama da Figura 57 ilustra a sequência de acções para realizar os casos de utilização *Listar processos*, *Prosseguir instância*, *Suspender instância* e *Terminar instância*. Estes casos de utilização foram agrupados num único diagrama porque necessitam de utilizar os *Web Services* de gestão de processos e instâncias do Apache ODE. Este facto evidencia a dependência existente no desenvolvimento de um sistema de gestão, como o que aqui é descrito, e as interfaces programáticas de um sistema de execução.

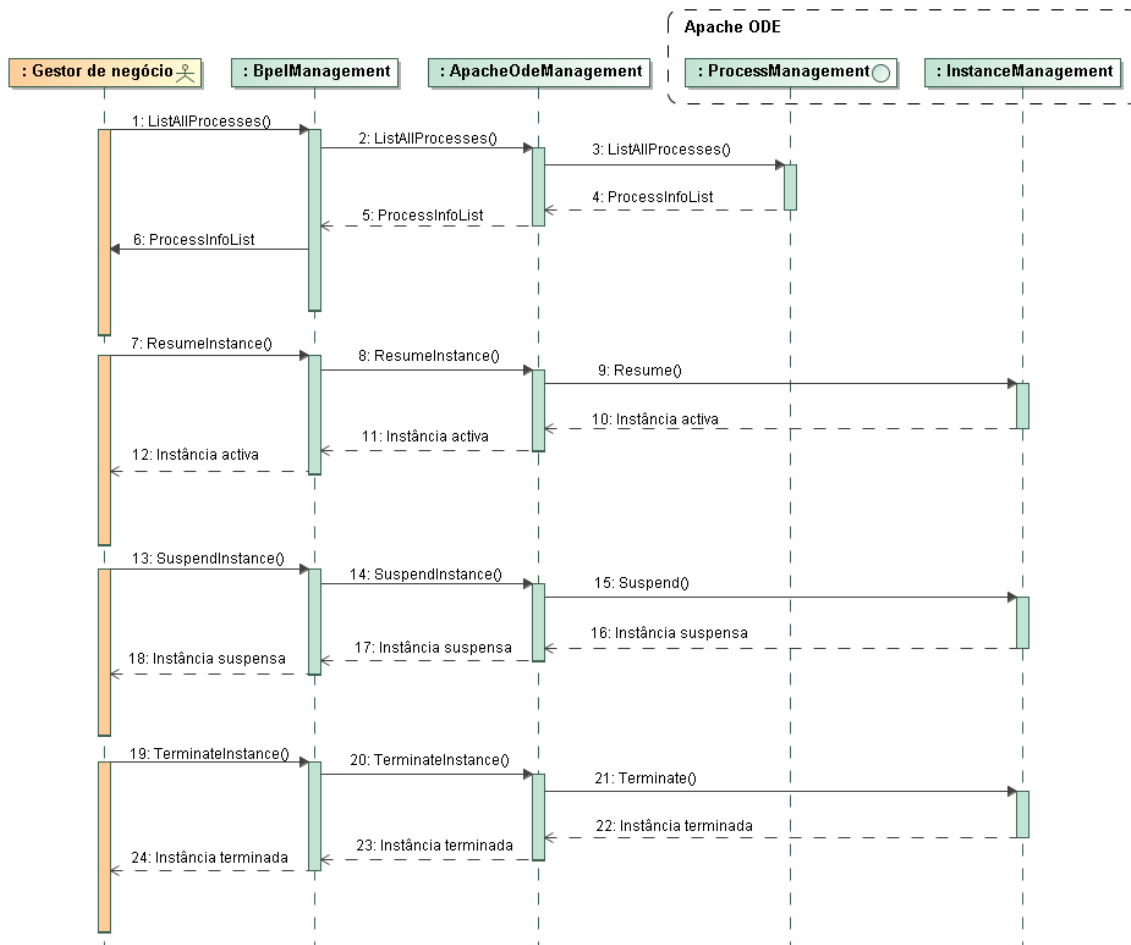


Figura 57 – Diagrama de seq. do Serviço de Gestão (interacção com Apache ODE)

O diagrama da Figura 58 ilustra a sequência de acções para realizar os casos de utilização *Listar sistemas de execução*, *Listar instâncias*, *Obter recurso* e *Obter propriedade do recurso*. A realização destes casos de utilização apresenta, no conjunto, a necessidade de aceder aos documentos de propriedades dos recursos.

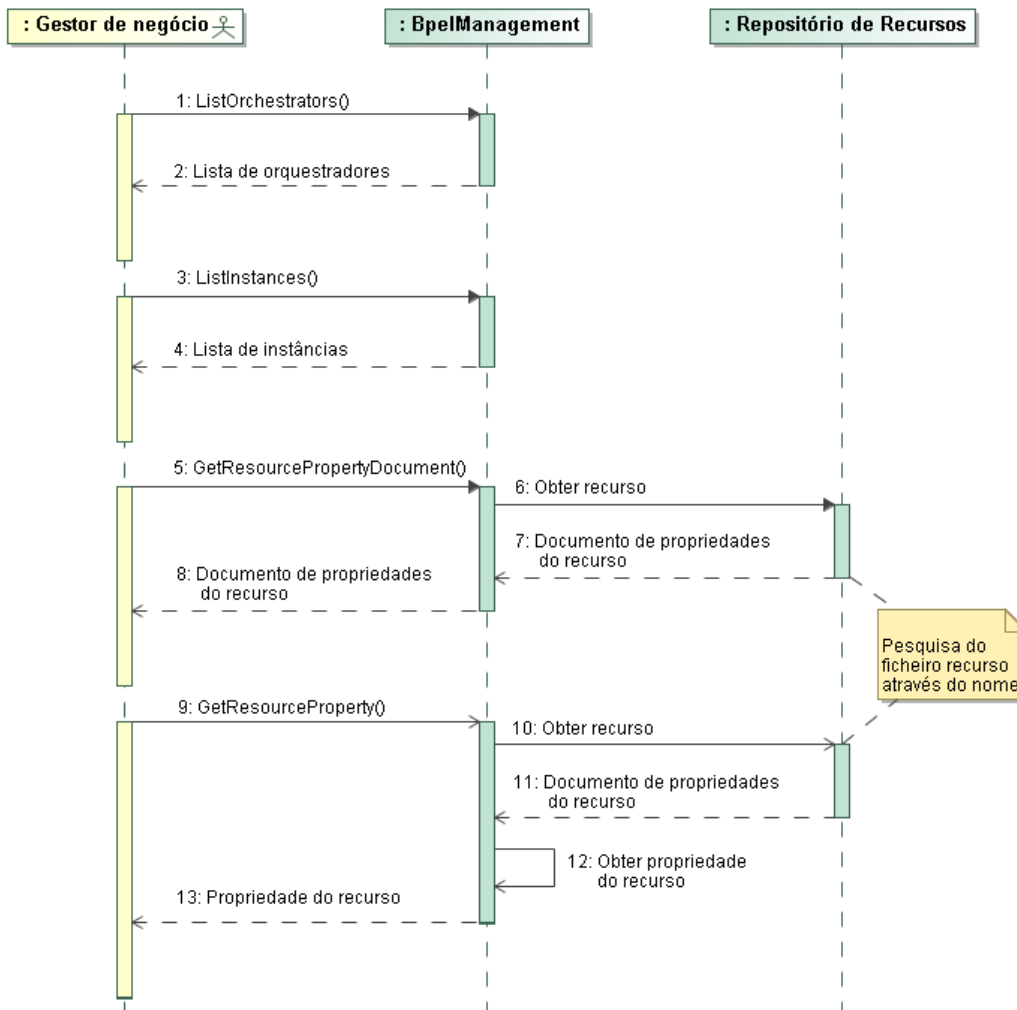


Figura 58 – Diagrama de seq. do Serviço de Gestão (interacção com recursos)

Na Figura 59 consta o diagrama de estrutura composta deste sistema. Neste diagrama são ilustradas as interfaces de comunicação com o sistema de execução e a aplicação de gestão. O diagrama apresentado é genérico para qualquer sistema de execução que disponibilize interfaces programáticas de gestão de processos e instâncias.

O controlo de gestão de processos é realizado pela classe *BpelManagement* que contém a implementação dos métodos da interface de gestão *IBpelManagement*. O módulo de controlo de acesso ao sistema de execução constitui a camada de acesso para obtenção de informações dos processos e instâncias.

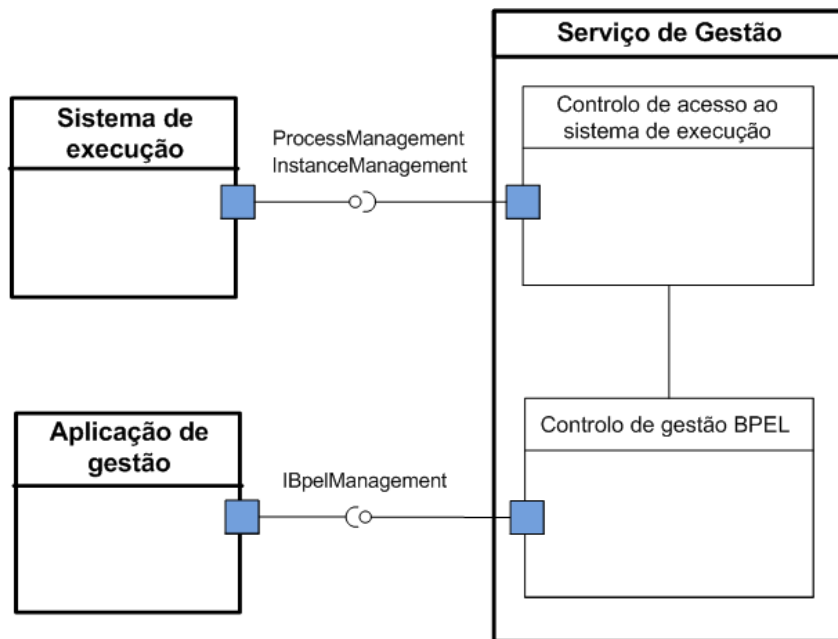


Figura 59 – Diagrama de estrutura composta do Serviço de Gestão

A análise do comportamento descrito nos diagramas apresentados, aliada aos requisitos definidos para este sistema, permitiu elaborar o diagrama de classes que suporta a estrutura da sua implementação.

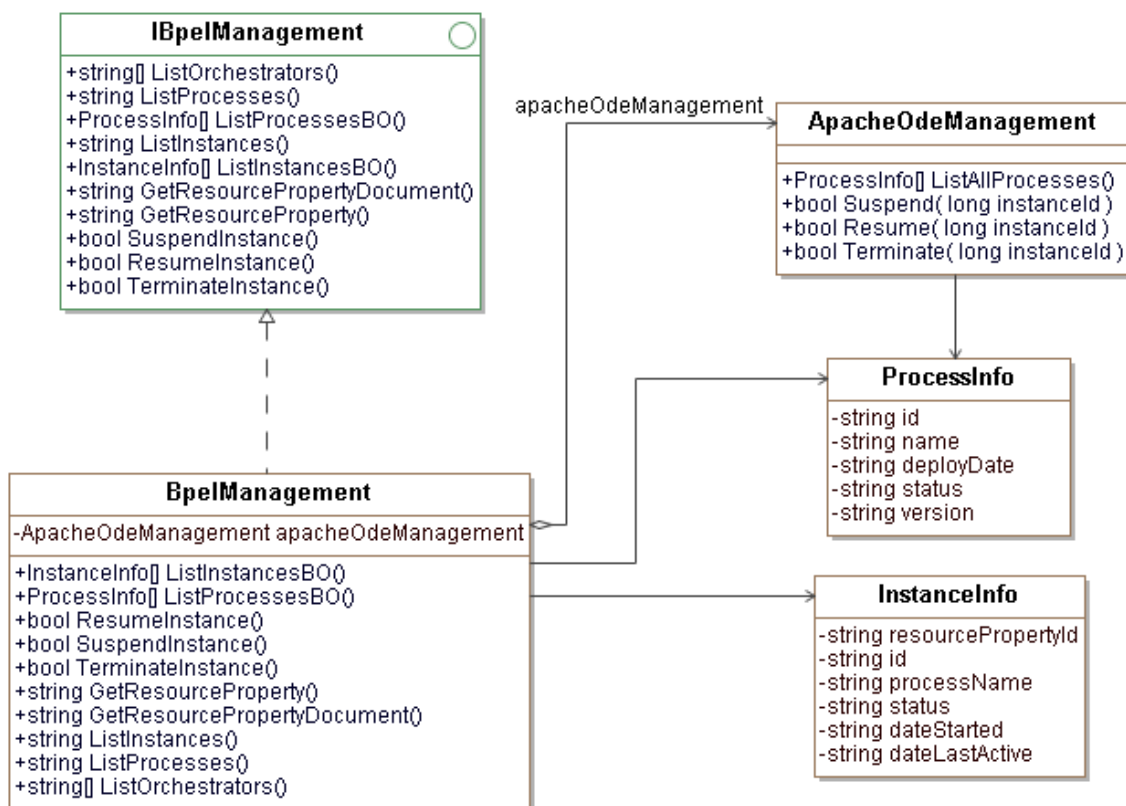


Figura 60 – Diagrama de classes do Serviço de Gestão

A Figura 61 ilustra a classe *ApacheOdeManagement*, onde está implementado o controlo de acesso ao sistema de execução. Esta classe permite encapsular as invocações aos *Web Services ProcessManagement* e *InstanceManagement*, do sistema Apache ODE.



Figura 61 – Classe *ApacheOdeManagement*

Retorno dos métodos

O retorno dos métodos da interface programática de gestão, que envolvem obtenção de informação acerca de processos e recursos, segue a abordagem proposta pelo quadro WSRF. Segundo este quadro de referência, o tipo de retorno é especificado através de uma *string* no formato XML. Desta forma, é possível usufruir de vantagens como a utilização de um formato de referência para troca de informação entre sistemas [6]. A utilização desta abordagem aplica-se aos métodos *ListProcesses*, *ListInstances*, *GetResourcePropertyDocument* e *GetResourceProperty*.

Os parágrafos seguintes detalham a implementação dos métodos da interface programática *IBpelManagement*.

Método *ListOrchestrators*

O método *ListOrchestrators* permite listar os sistemas de execução cujos processos podem ser geridos. A sua assinatura é a seguinte:

```
string[] ListOrchestrators();
```

Listagem 4 – Assinatura do método *ListOrchestrators*

Este método retorna um *array* de objectos do tipo *string*, onde cada posição representa o identificador do sistema de execução.

Método *ListProcesses*

O método *ListProcesses* permite listar os processos do sistema de execução. A assinatura do método é a seguinte:

```
string ListProcesses(string orchestratorId);
```

Listagem 5 – Assinatura do método *ListProcesses*

A implementação deste método considera a utilização do método *ListAllProcesses* da classe *ApacheOdeManagement*. Este método invoca o método *ListAllProcesses*, referente ao *Web Service ProcessManagement*, do sistema Apache ODE.

O retorno do método *ListProcesses* contém a lista de processos, no formato de *string* XML. A Listagem 6 exemplifica o XML retornado, onde se observa o elemento raiz *bpelProcessInfoList* e o elemento filho *bpelProcessInfo*, cujos atributos contêm a informação do processo.

```
<?xml version="1.0"?>
<bpelProcessInfoList>
  <bpelProcessInfo
    id="LoanRequestProcess-57"
    name="LoanRequestProcess"
    deploydate="06-09-2009 18:47:31"
    status="ACTIVE"
    version="57"/>
</bpelProcessInfoList>
```

Listagem 6 – Exemplo de resposta do método *ListProcesses*

Método *ListInstances*

O método *ListInstances* lista as instâncias (recursos) de um processo no sistema de execução. A sua assinatura é a seguinte:

```
string ListInstances(string orchestratorId, string processName,
                    string processId);
```

Listagem 7 – Assinatura do método *ListInstances*

O parâmetro *orchestratorId* permite identificar o sistema de execução e o correspondente repositório de recursos onde estão armazenados os documentos de propriedades. Após identificado o repositório de recursos, a listagem de instâncias de um processo é obtida através da pesquisa de todos os documentos de propriedades dos recursos, cujo nome inclui os parâmetros *processName* e *processId*. Para cada documento de propriedades seleccionado, na pesquisa anterior, é obtido o elemento *BpelInstanceInfo*. Este elemento é filho do elemento raiz *bpelProcess*, de acordo com implementação detalhada no capítulo “4.3.1 Sistema Gerador de Recursos (SGR)”, e contém a informação da instância. O retorno do método é uma *string* XML, composta pelo elemento raiz *bpelInstanceInfoList* e os elementos filho *bpelInstanceInfo*, cujos atributos contêm a informação da instância. O formato XML de resposta é exemplificado na listagem seguinte.

```

<?xml version="1.0"?>
<bpelInstanceInfoList>
  <bpelInstanceInfo
    dateLastActive="2009-09-06T19:20:09.406+01:00"
    dateStarted="2009-09-06T19:20:09.343+01:00"
    instanceId="59402"
    processName="LoanRequestProcess"
    resourcePropertyId="2"
    status="ACTIVE" />
</bpelInstanceInfoList>

```

Listagem 8 - Exemplo de resposta do método *ListInstances*

Métodos auxiliares *ListProcessesBO* e *ListInstancesBO*

Foram desenvolvidos dois métodos auxiliares na interface programática de gestão: *ListProcessesBO* e *ListInstancesBO*. O fundamento para o seu desenvolvimento foi simplificar a aplicação de gestão, descrita no subcapítulo seguinte. Esta necessitaria de interpretar os dados retornados pelos métodos *ListProcesses* e *ListInstances* no formato XML. Para evitar este comportamento, estes dois métodos retornam *Business Objects* (BO) – objectos que representam entidades no domínio da lógica de negócio associada a este sistema.

O retorno destes métodos auxiliares contém, respectivamente, uma lista de objectos do tipo *ProcessInfo* ou *InstanceInfo*. Em cada posição da lista está presente um objecto (*Business Object*) com a informação de um processo ou instância, permitindo o acesso às suas propriedades e consequente obtenção da informação sem necessidade de interpretação dos dados em XML. Realça-se que estes métodos foram desenvolvidos exclusivamente para efeitos de simplificação da demonstração no âmbito de um protótipo demonstrador. As classes associadas aos tipos de dados de retorno são ilustradas na figura seguinte.

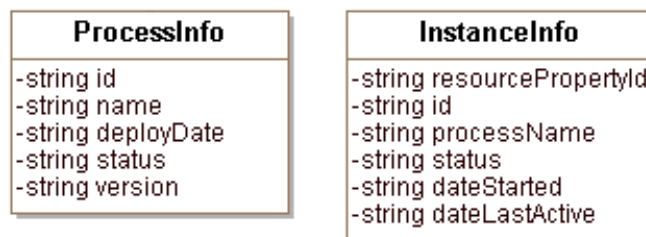


Figura 62 – Definição das classes *ProcessInfo* e *InstanceInfo*

Métodos *GetResourcePropertyDocument* e *GetResourceProperty*

A invocação destes dois métodos pressupõe a invocação prévia dos métodos *ListProcesses* e *ListInstances*, para obtenção dos valores dos seus parâmetros. Entre

estes constam o nome do processo e os identificadores do sistema de execução, processo, instância (recurso) e propriedade do recurso.

O método *GetResourcePropertyDocument* retorna todo o documento de propriedades do recurso, ou seja, o conteúdo do ficheiro XML representativo de uma instância.

```
string GetResourcePropertyDocument(string orchestratorId, string  
    processName, string processId, string instanceId);
```

Listagem 9 – Assinatura do método *GetResourcePropertyDocument*

O método *GetResourceProperty* permite obter a informação de gestão/monitorização de uma propriedade do recurso, identificada através do seu atributo *resourcePropertyId*. O retorno é efectuado no formato de uma *string* XML que contém todo o elemento XML associado.

```
string GetResourceProperty(string orchestratorId, string  
    processName, string processId, string  
    instanceId, string resourcePropertyId);
```

Listagem 10 – Assinatura do método *GetResourceProperty*

Métodos *SuspendInstance*, *ResumeInstance* e *TerminateInstance*

Os métodos *SuspendInstance*, *ResumeInstance* e *TerminateInstance* permitem, respectivamente, suspender, prosseguir ou terminar a execução de uma instância. A assinatura destes métodos é apresentada na listagem seguinte.

```
bool SuspendInstance(string orchestratorId, string instanceId);  
bool ResumeInstance(string orchestratorId, string instanceId);  
bool TerminateInstance(string orchestratorId, string instanceId);
```

Listagem 11 – Assinatura dos métodos *Suspend*, *Resume* e *Terminate Instance*

A implementação destas operações inclui a utilização da classe *ApacheOdeManagement*. Esta classe disponibiliza métodos para invocar as operações do *Web Service InstanceManagement* (Apache ODE). O tipo de retorno é um valor indicativo de sucesso ou falha na operação.

O subcapítulo seguinte descreve os detalhes de implementação da aplicação de gestão/monitorização, que utiliza o serviço aqui descrito, para efectuar a gestão dos processos WS-BPEL nos sistemas de execução.

4.2.4. Aplicação de Gestão Integrada (AGI)

A aplicação de gestão de processos WS-BPEL constitui a camada de apresentação da solução descrita. Este sistema unifica a gestão de processos em diversos sistemas de execução, disponibilizando uma interface gráfica para demonstração das funcionalidades da solução. Ainda que no âmbito de um protótipo demonstrador, esta aplicação pode ser considerada exemplificativa para o desenvolvimento de uma outra, que permita gerir/monitorizar processos WS-BPEL, num ambiente de gestão integrada numa organização. A Figura 63 ilustra o posicionamento da aplicação na arquitectura da solução.

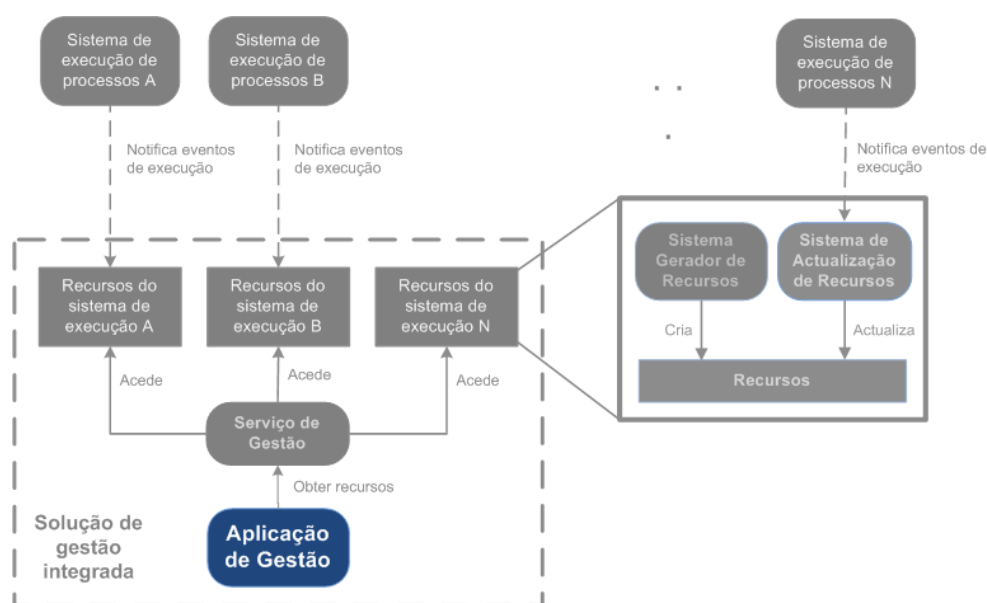


Figura 63 – Aplicação de Gestão na arquitectura proposta

A listagem seguinte enumera as funcionalidades deste sistema, de acordo com os requisitos de uma interface de gestão, das quais a terceira e a quinta foram adicionadas a este sistema, face ao descrito no subcapítulo anterior:

1. Listar sistemas de execução;
2. Listar processos do sistema de execução;
3. Apresentar informação do processo;
4. Listar instâncias dos processos (recursos);
5. Apresentar informação da instância;
6. Obter o documento de propriedades de um recurso;
7. Obter uma propriedade de um recurso;
8. Suspender a execução de uma instância;
9. Prosseguir a execução de uma instância;
10. Terminar uma instância.

A Figura 64 ilustra o diagrama de casos de utilização da aplicação de gestão, cuja descrição é disponibilizada no Apêndice H. Apenas os casos *Seleccionar processo* e *Seleccionar instância* diferem do diagrama de casos de utilização apresentado no Serviço de Gestão, por serem exclusivamente relevantes no âmbito da interface gráfica desta aplicação.

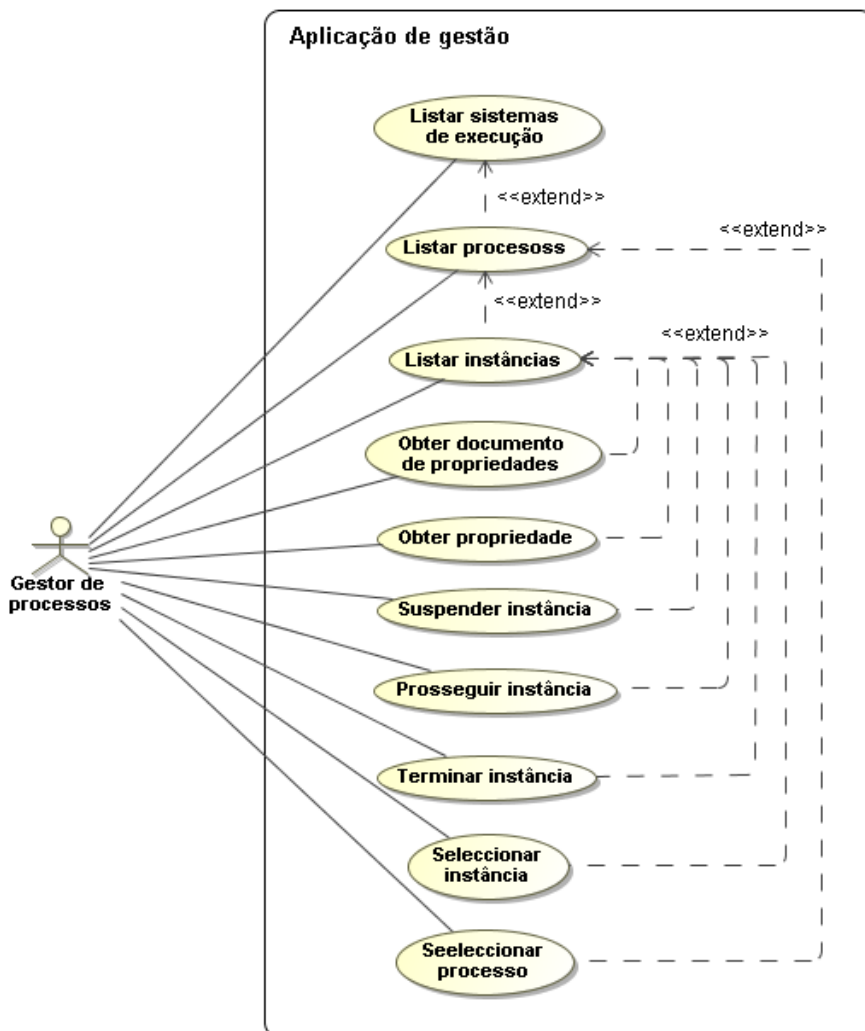


Figura 64 – Diagrama de casos de utilização da aplicação de gestão

À semelhança das funcionalidades identificadas, o diagrama de casos de utilização apresentado é semelhante ao ilustrado no Serviço de Gestão. Como consequência deste facto, salienta-se que os diagramas de sequência das acções a realizar, para implementar as funcionalidades descritas, foram apresentados no capítulo “4.2.3 Serviço de Gestão” e são elucidativos das acções a realizar nesta aplicação.

Relativamente à implementação do código da aplicação, a opção recaiu sobre a utilização da tecnologia .Net *Windows Forms*. Esta opção promove a heterogeneidade

de sistemas nesta solução de integração. A Figura 65 apresenta a respectiva interface gráfica, onde é possível identificar as funcionalidades anteriormente enumeradas.

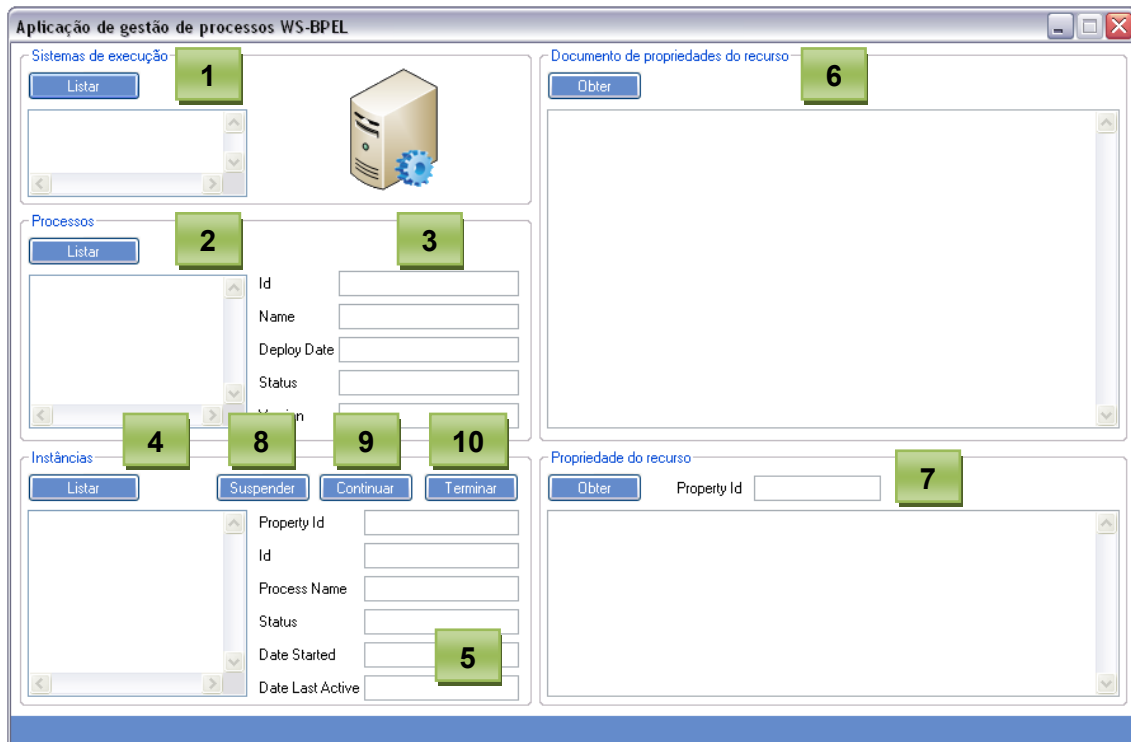


Figura 65 – Interface da aplicação de gestão/monitorização

A demonstração das funcionalidades é apresentada no capítulo 5. Demonstração. A aplicação utiliza uma referência para o *Web Service* do Serviço de Gestão que permite invocar métodos do respectivo serviço. A Figura 66 ilustra o diagrama de estrutura composta deste sistema, onde é possível identificar a relação com o Serviço de Gestão, através da interface *IBpelManagement*.

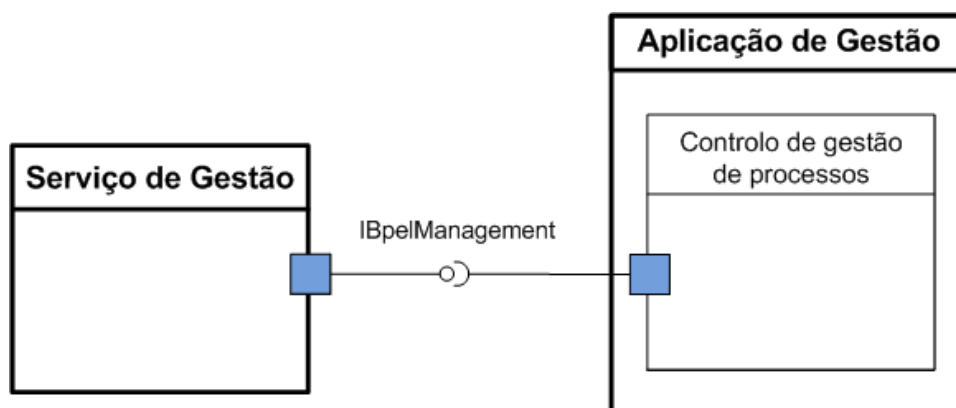


Figura 66 – Diagrama de estrutura composta da Aplicação de Gestão

Segundo o âmbito do desenvolvimento desta aplicação de gestão, como parte integrante do protótipo demonstrador da solução, a qualidade e optimização da

interface gráfica não foram considerados como requisitos prioritários na sua implementação. Por consequência deste facto e para simplificação consciente do desenvolvimento, as operações do *web service* são invocadas directamente pelos métodos de processamento de eventos da interface gráfica. Salvaguarda-se que, num quadro de evolução deste protótipo demonstrador, não deve ser adoptada esta metodologia. A invocação das operações do *web service* não deve ocorrer de forma síncrona no processamento dos eventos da interface gráfica, de modo a garantir a sua disponibilidade e evitar o efeito de bloqueio. Sugere-se a delegação do processamento dos pedidos ao *Web Service* a outra *thread* diferente da *thread* de *user-interface*.

O subcapítulo seguinte apresenta algumas considerações no âmbito do desenvolvimento do protótipo demonstrador. Algumas destas considerações referem-se a pontos críticos de decisão durante o desenvolvimento do projecto, que influenciaram a fase de implementação e demonstração da solução.

4.3. Considerações sobre demonstrador

A implementação da abordagem de solução considerada foi condicionada por alguns detalhes que são descritos neste subcapítulo.

A definição de cada elemento XML do recurso, representativo de um elemento da especificação WS-BPEL, apresenta um nível de detalhe de informação dependente da informação disponibilizada pelo sistema de execução. A possibilidade de incluir mais ou menos detalhe nas propriedades do recurso, ou seja, mais ou menos informação de gestão/monitorização, depende da interface de gestão de processos e instâncias e da interface de eventos disponíveis no sistema de execução. Se este não suportar nenhuma funcionalidade de notificação de alterações do estado das instâncias, durante a sua execução, a implementação deste sistema fica comprometida. Seria necessário concretizar uma estratégia de actualização de recursos alternativa ao padrão *Observer*, por exemplo, através de interrogação constante do estado das instâncias e seus elementos constituintes (mecanismo de *polling*).

Existe uma diferença da abordagem proposta neste documento relativamente à que é proposta no artigo de referência “A Management Framework for WS-BPEL” [1]. Os autores deste artigo representam a informação do processo e das suas instâncias em recursos distintos. Na proposta deste projecto, apenas as instâncias são representadas em recursos e actualizadas em função dos eventos ocorridos no sistema de execução. Esta decisão foi tomada, à data de implementação, pela inexistência de suporte, por parte do sistema Apache ODE, a eventos de alteração na informação do processo. Entre estes, constata-se a lacuna na notificação de alterações na versão, número de instâncias iniciadas, entre outros [29].

Ainda que o sistema de execução disponibilize interfaces de gestão sobre as instâncias, a limitação da informação que disponibiliza influencia a gestão/monitorização que é possível efectuar sobre as mesmas em execução. Após a implementação do receptor de eventos de execução das instâncias, do tratamento os eventos e da utilização da interface programática de gestão de instâncias (*InstanceManagement*), verificou-se que o sistema de execução apresenta alguns erros, não mencionados no sítio oficial [29]. Um desses erros é o facto do método *ListAllInstances*, do *Web Service InstanceManagement*, especificado no Javadoc em [32], listar a informação de todas as instâncias excepto a da instância corrente, sobre a qual ocorre um evento. Este facto compromete o desenvolvimento deste projecto e a validação da solução proposta durante a execução da instância, dado que não é

possível efectuar qualquer tipo de teste ou demonstração sobre eventos ocorridos numa instância de um processo em execução.

Este problema foi reportado no fórum ASF JIRA [34], pertencente à comunidade de desenvolvimento de várias iniciativas, entre as quais se encontra o Apache ODE, como um erro que afecta a versão do Apache ODE 2.0 *beta*. No entanto, também afecta a versão 1.3.3, utilizada neste projecto, que era a versão estável mais recente à data de implementação. Salvaguarda-se que foi efectuada uma pesquisa de informações sobre soluções para contornar este problema e, inclusive, foi submetido um *post* no referido fórum para o qual não se obteve resposta até à data de realização deste documento. O problema reportado está disponível no endereço <https://issues.apache.org/jira/browse/ODE-514>.

A solução encontrada para minimizar as consequências desta limitação consistiu na utilização de uma instância, anteriormente executada e concluída, para realização dos testes e demonstração da solução. Realça-se que o sistema SAR foi desenvolvido de forma genérica e está preparado para que, numa futura versão do Apache ODE que apresente este problema resolvido, opere sem qualquer restrição. No entanto, para efeitos de teste e demonstração da solução, existe uma instância de teste para a qual foi gerado, propositadamente, um recurso. Quando ocorre um evento de execução sobre uma instância em execução, o seu identificador é manipulado de forma a conter o valor do identificador da instância de teste e todas as alterações surtem efeito na actualização do seu recurso.

A restrição descrita não permite que o protótipo demonstrador acompanhe a evolução da actualização das propriedades de um recurso em tempo real, enquanto uma instância é executada. Esta restrição tem como consequência que apenas seja possível monitorizar instâncias numa fase pós-execução, constituindo uma limitação relativamente ao objectivo de efectuar a monitorização durante e pós-execução.

O desenvolvimento do protótipo demonstrador não considerou nenhuma característica de segurança, relativamente aos dados armazenados e comunicação entre sistemas. Considera-se que, segundo o objectivo de desenvolvimento de um propósito demonstrador, este requisito não se encontra no âmbito da análise da arquitectura de gestão integrada que é proposta.

4.4. Síntese

O protótipo demonstrador concretiza a arquitectura proposta através do desenvolvimento de um conjunto de sistemas que responde aos objectivos e estratégia de solução definidos.

O Sistema Gerador de Recursos é responsável pela geração da representação dos processos WS-BPEL para recursos.

O Sistema de Actualização de Recursos permite actualizar as propriedades de cada recurso, em função dos eventos ocorridos no sistema de execução, através da utilização das interfaces programáticas, de eventos e gestão, do sistema de execução.

O Serviço de Gestão constitui a interface de acesso à informação dos recursos. Disponibiliza uma interface programática que permite à Aplicação de Gestão Integrada consultar a informação de gestão/monitorização das instâncias dos processos, persistente nas propriedades dos recursos.

A Aplicação de Gestão Integrada é a camada de apresentação da solução. Este sistema unifica a gestão de processos em diversos sistemas de execução, disponibilizando uma interface gráfica para demonstração das funcionalidades da gestão/monitorização.

5. Demonstração de resultados

Este capítulo tem como objectivo validar os conceitos e a estratégia considerados para a abordagem proposta neste projecto.

A demonstração é constituída por diversas fases, enumeradas em seguida:

1. Realização de um processo de negócio em WS-BPEL;
2. Inserção do processo no sistema de execução Apache ODE;
3. Geração da representação do processo em recurso;
4. Instanciação do processo;
5. Criação de um recurso para a respectiva instância;
6. Actualização das propriedades do recurso durante a execução do processo;
7. Utilização da aplicação de gestão para monitorizar o processo e visualização das propriedades do recurso.

Os subcapítulos seguintes incluem a descrição destas fases, ilustrando, sempre que possível, os resultados obtidos pela execução de cada subsistema descrito no capítulo “4. Implementação”. A demonstração culmina com a ilustração da aplicação de gestão, onde é possível visualizar a informação de monitorização do processo WS-BPEL realizado.

5.1. Definição de processo WS-BPEL

O processo WS-BPEL desenvolvido enquadra-se na área bancária e tem por objectivo permitir a realização de empréstimos.

Descrição

É requerido um empréstimo para um determinado montante sendo avaliado, numa primeira fase, o seu risco. Se o montante requerido representar um valor inferior a 10.000, conclui-se que não há risco de empréstimo. Caso contrário, é necessário invocar o *Web Service RiskDeciderWS* (actividade *InvokeRiskDecider*), que recebe como parâmetro o montante e avalia o seu risco. O retorno da invocação é utilizado para determinar se o empréstimo pode ou não ser concedido. Caso não possa ser concedido, a variável de retorno do processo é afectada com a *string* “Empréstimo não realizado” e o processo retorna de imediato. Caso não exista risco, é invocado o *Web Service LoanMakerWS* (actividade *InvokeLoanMaker*) que permite efectivar o empréstimo, sendo retornada a *string* “Empréstimo realizado”. A Figura 67 ilustra o processo descrito.

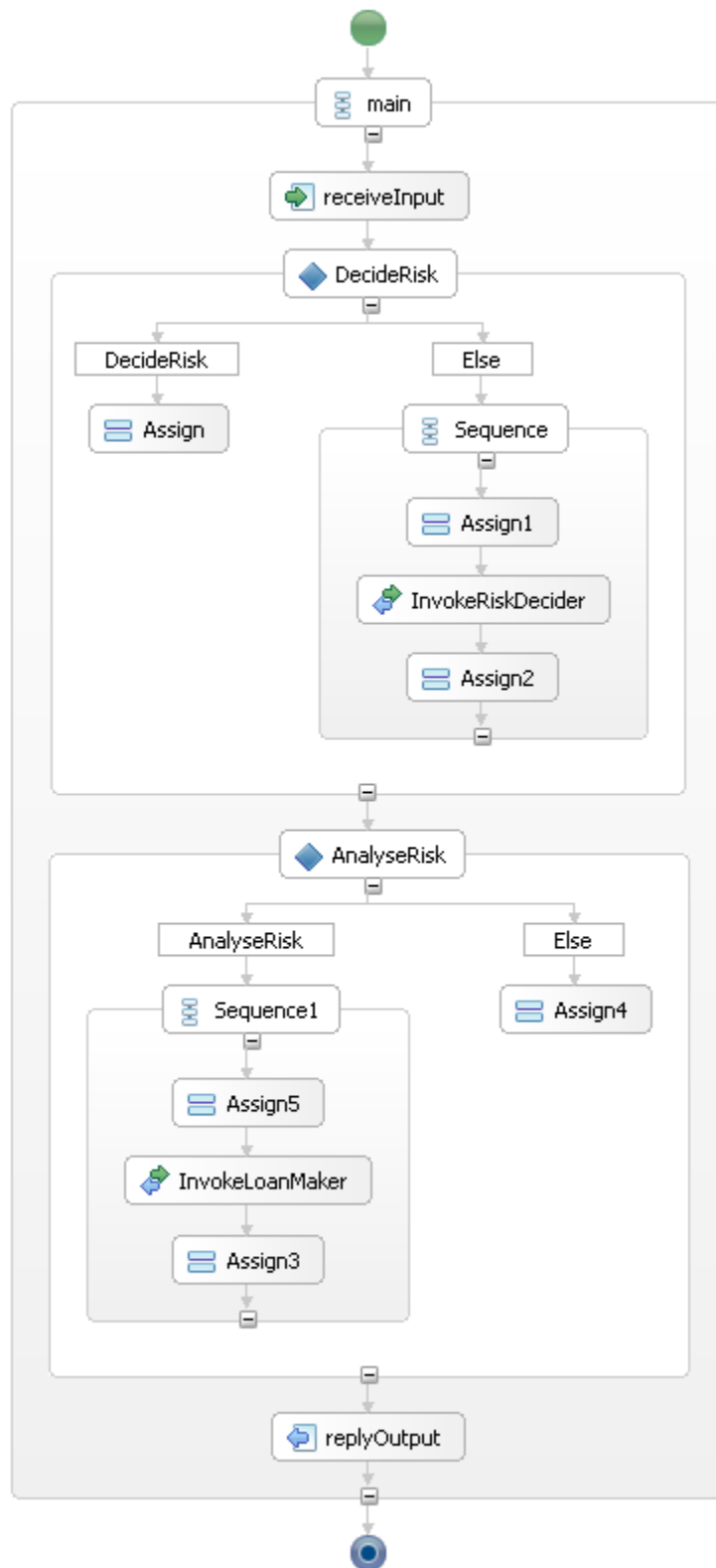


Figura 67 – Processo WS-BPEL *Loan Request*

O ficheiro “LoanRequestProcess.bpel” contém a definição WS-BPEL gerada para este processo e está disponível para consulta no Apêndice F.

Inserção do processo no Apache ODE

A inserção de processos WS-BPEL no sistema de execução Apache ODE necessita, para além do ficheiro de definição do processo, de um ficheiro auxiliar de configuração. Esse ficheiro é denominado “deploy.xml” e contém, entre outras informações, o nome do processo, o seu *partnerLink* e os *partnerLinks* para os serviços que utiliza (neste caso concreto, os *Web Services RiskDeciderWS* e *LoanMakerWS*). A listagem do ficheiro é apresentada em seguida.

```
<deploy xmlns="http://www.apache.org/ode/schemas/dd/2007/03"
  xmlns:pns="http://loanrequest"
  xmlns:wns="http://riskdecider" xmlns:mns="http://loanmaker">
  <process name="pns:LoanRequestProcess">
    <active>true</active>
    <provide partnerLink="client">
      <service name="pns:LoanRequestProcessService"
port="LoanRequestProcessPort"/>
    </provide>
    <invoke partnerLink="RiskDecider">
      <service name="wns:RiskDeciderWS"
port="RiskDeciderWSHttpSoap11Endpoint"/>
    </invoke>
    <invoke partnerLink="LoanMaker">
      <service name="mns:LoanMakerWS"
port="LoanMakerWSHttpSoap11Endpoint"/>
    </invoke>
    <cleanup on="always" />
  </process>
</deploy>
```

Listagem 12 – Configuração para inserir o processo *Loan Request* no Apache ODE

Para realizar a inserção é necessário copiar este ficheiro e o ficheiro com a definição do processo WS-BPEL para uma pasta interna do sistema Apache ODE. Caso a inserção ocorra com sucesso, é possível observar o novo processo na *interface web* que o sistema de execução disponibiliza. Este é o procedimento de referência para definição de um processo WS-BPEL e inserção no sistema Apache ODE [35].

A adopção de ferramentas como o NetBeans ou Eclipse, juntamente com o *plug-in* Eclipse BPEL Designer, constituem dois ambientes de desenvolvimento de processos WS-BPEL. Estes sistemas apresentam fragilidades ao nível da usabilidade e simplicidade de criação/inserção de processos, tornando esta fase morosa e com um tempo de aprendizagem elevado. As funcionalidades que se dispersam da definição de processos WS-BPEL ou os erros no funcionamento dos editores gráficos são alguns dos problemas verificados. Considera-se que estes sistemas devem ser reavaliados, relativamente à perspectiva de desenvolvimento de processos WS-BPEL, para que o seu contributo promova a automatização dos processos nas organizações.

5.2. Representação de processo em recurso

Quando o processo é inserido com sucesso, o sistema SGR, que se encontra a observar os ficheiros inseridos na respectiva pasta de inserção, gera a sua representação em recurso. A representação gerada está disponível no Apêndice C.

O facto dos processos existentes no sistema de execução serem disponibilizados como *Web Services* facilita a sua instanciação. Foi desenvolvida uma aplicação, denominada *InstantiateBpelProcesses*, que permite instanciar processos WS-BPEL no Apache ODE. Para o efectuar, apenas é necessário incluir uma referência *web* para o endereço do processo, utilizar o objecto *proxy* correspondente para realizar a sua invocação e conseqüente criação de uma nova instância.

Quando ocorre a criação de uma instância, associada ao evento *NewProcessInstanceEvent*, é efectuada uma cópia do ficheiro acima referido e renomeada segundo a terminologia descrita no capítulo “3.5.1 Representação de processos WS-BPEL em recursos”. Com o evoluir da execução do processo, são gerados diversos eventos e o receptor desenvolvido no sistema SAR permite actualizar as propriedades do recurso com os novos valores. Demonstra-se, em seguida, as diferenças presentes numa propriedade *bpelAssign*, antes e depois da sua execução, relativamente à informação de monitorização que nela pode ser consultada.

```
<bpelAssign dateEnded="empty" dateStarted="empty" name="Assign"
resourcePropertyId="21" status="inactive" validate="no"/>
```

Listagem 13 - Propriedade *bpelAssign* original

```
<bpelAssign dateEnded="2009-09-06 23:58:49" dateStarted="2009-
09-06 23:58:49" name="Assign" resourcePropertyId="21"
status="COMPLETED" validate="no"/>
```

Listagem 14 - Propriedade *bpelAssign* após actualização

É possível verificar a diferença entre os valores dos atributos do elemento XML. Esta actualização específica ocorre em função dos eventos de actividade iniciada (*ActivityEnabledEvent*) e executada (*ActivityExecStartEvent*).

No final da execução do processo, o recurso apresenta as suas propriedades com a informação de monitorização que permitirá uma avaliação do processo pós-execução. O ficheiro com o recurso actualizado está disponível para consulta no Apêndice D.

5.3. Gestão/monitorização da execução do processo

A conclusão das fases anteriores permite avançar para a última fase da demonstração. Nesta fase pretende-se verificar que é possível gerir/monitorizar a execução do processo WS-BPEL *Loan Request*, em execução no sistema Apache ODE, segundo a proposta de solução apresentada.

A execução da instância de um processo pode ser despoletada através da invocação do *Web Service* que o expõe. A Listagem 15 apresenta o código desenvolvido para iniciar uma instância do processo *Loan Request*.

```
LoanRequestProcessPortTypeClient loanRequestProc = new
LoanRequestProcessPortTypeClient ("LoanRequestProcessSOAP11port_http");

LoanRequestProcessRequest req = new LoanRequestProcessRequest ();
req.input = "5000";
LoanRequestProcessResponse response = loanRequestProc.process (req);
```

Listagem 15 - Criação de instância do processo *Loan Request*

A Figura 68 ilustra a interface gráfica da aplicação de gestão, onde é possível observar a monitorização de uma instância do processo *Loan Request*.

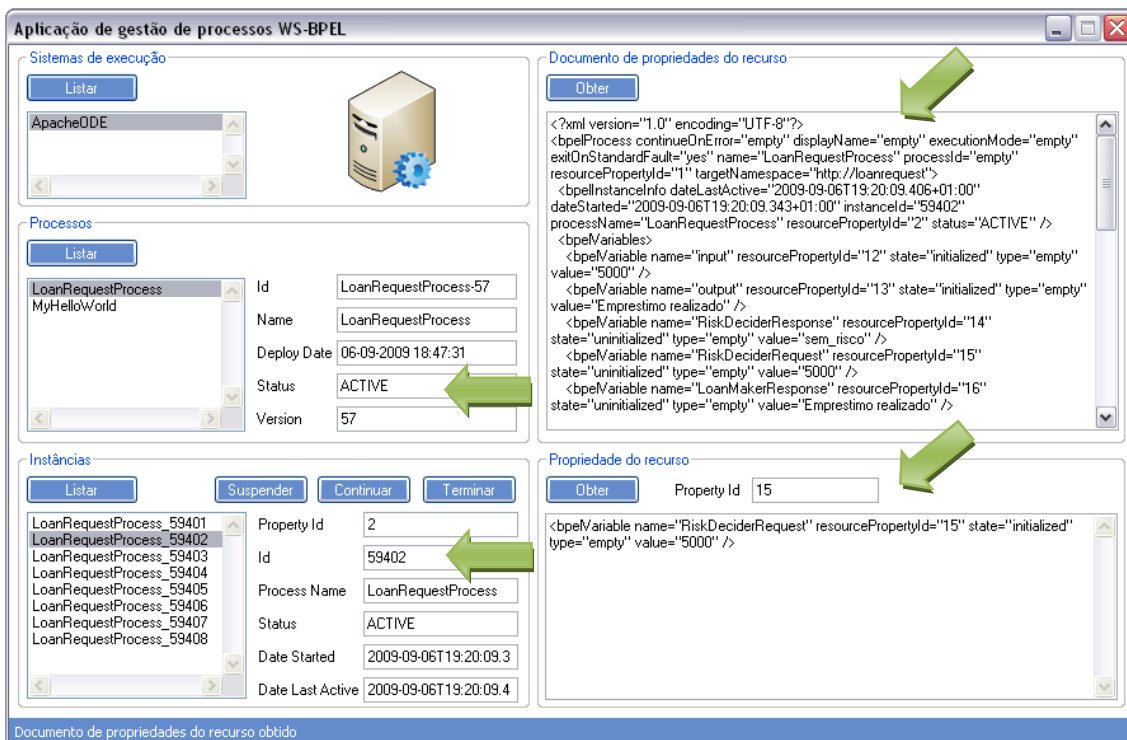


Figura 68 – Monitorização do processo WS-BPEL *Loan Request*

É possível observar a lista de sistemas de execução, os processos e as instâncias do sistema/processo seleccionados. A informação do processo (*LoanRequestProcess-57*)

e instância seleccionada (59402), o documento de propriedades do recurso, representativo da instância e a propriedade do recurso, identificada pelo valor “15” (valor do elemento XML *resourcePropertyId*), estão destacados nas setas da imagem.

As funcionalidades de obtenção do documento de propriedades do recurso e de uma propriedade do recurso, apresentam os dados retornados do Serviço de Gestão, no formato em XML, segundo a especificação WSRF. Considera-se que não é relevante, para a demonstração da solução, que seja efectuada alguma transformação no formato da apresentação destes dados. De acordo com esta consideração, a decisão tomada foi manter o formato original, usufruindo de legibilidade do formato XML.

A demonstração efectuada neste capítulo permite validar a solução proposta, ao nível da estratégia e detalhe de implementação, correspondendo às expectativas esperadas.

6. Conclusão

Uma organização possui uma plataforma computacional, constituída por um ou mais sistemas de informação que promovem a automatização dos processos de negócio. Nesta plataforma podem existir diversos sistemas de execução de processos WS-BPEL. A capacidade de gerir e monitorizar a sua execução, através de um modelo integrado, promove a uniformização da gestão, actualmente dispersa e dependente destes sistemas específicos.

A inexistência de um modelo uniforme para representação e acesso à informação de gestão dos processos, baseado em interfaces abertas, compromete a utilização de um sistema capaz de unificar a respectiva gestão, em diversos sistemas de execução.

O modelo de integração proposto é uma solução de compromisso com um tipo de integração adoptado, actualmente, em algumas organizações. É uma estratégia conservadora, constituída por um modelo que contempla sucessivas camadas de integração entre sistemas, desenvolvidas à medida das necessidades. A cultura e tendência natural para desenvolvimento de sistemas fechados e monolíticos e a dificuldade na mudança de padrões, são alguns dos seus factores impulsionadores. A visão estanque da integração de sistemas e a dificuldade de ruptura com processos organizacionais existentes, assentes em sistemas com provas dadas de funcionamento, são outros exemplos. Estes aspectos resultam do desenvolvimento de soluções de integração desenhadas para responder a objectivos muito específicos de comunicação ou transformação de dados. Estes objectivos são concretizados em infra-estruturas rígidas e pouco coesas, que originam sistemas fechados, pouco flexíveis e fracamente adaptados à dinâmica da evolução tecnológica e dos requisitos do negócio.

A solução desenvolvida unifica a gestão de diversos sistemas de execução segundo este modelo de integração. Foi desenvolvida uma aplicação de gestão integrada que encapsula a especificidade de acesso a cada executor de processos numa camada de integração. Esta camada é responsável pela obtenção da informação através das interfaces programáticas proprietárias de cada executor. O esforço de uniformização é realizado através da transformação dos dados específicos de cada sistema de execução para um formato neutro, baseado em especificações abertas, do quadro de referência WSRF. Desta forma, promove-se a confiança dos especialistas do negócio nos sistemas de execução que possuem, evitando desenvolvimentos nestes últimos.

Os dados de gestão/monitorização de cada instância são representados sob a forma de recursos. Um recurso é uma entidade identificável num domínio, com zero ou mais propriedades para armazenamento de informação, acessível através de *Web Service*, segundo a norma OASIS *WS-Resource*, do quadro WSRF. Cada instância de um processo é representada num ficheiro XML, constituindo os recursos que podem ser geridos segundo o referido quadro. A geração é baseada num modelo de recurso e na definição WS-BPEL do processo. A geração da representação das instâncias, sob a forma de recursos, é dinâmica, no sentido em que se baseia na definição do processo WS-BPEL e o recurso é gerado em função dos elementos constituintes desta definição. A adopção da metodologia e dos conceitos teóricos do quadro WSRF revelou adaptabilidade à solução proposta, para modelar e aceder a recursos com estado.

A manutenção do estado dos recursos em ficheiros XML promove a simplicidade do acesso à informação através de interrogações XPath, assegurando a portabilidade, legibilidade e extensibilidade da solução proposta. Num quadro de evolução desta solução, pode ser considerada a utilização de sistema de gestão de base de dados (SGBD), pelas suas características de controlo de concorrência, segurança, replicação, integridade e consistência da informação, implícitas nestes sistemas. Este projecto não se foca nestas problemáticas e a solução adoptada restringiu-se à utilização de ficheiros XML, usufruindo das vantagens enunciadas.

A actualização dos recursos, durante a execução da instância, é baseada nos eventos ocorridos sobre a mesma. As alterações ocorridas nos seus elementos são notificadas, segundo o padrão *Observer*, a um receptor responsável pela actualização dinâmica das propriedades do recurso associado. A proposta de solução apresentada é genérica e compatível com qualquer executor WS-BPEL que disponibilize interfaces programáticas de gestão e notificação de eventos de execução. As limitações destas interfaces programáticas influenciam a informação de gestão/monitorização disponibilizada aos gestores de processos.

A utilização de apenas um sistema de execução, para demonstração da solução, está relacionada com a dificuldade de integrar outros sistemas, devido à especificidade das suas interfaces programáticas de gestão e notificação de eventos. Cada fabricante possui as suas especificações e, como não existe uma convergência para a sua uniformização, é necessário desenvolver a lógica de recepção de eventos e actualização de recursos para cada sistema de execução. A solução desenvolvida é genérica e permite, para efeitos de demonstração, efectuar a gestão de processos

segundo a abordagem proposta. Em termos de manutenção, a adição de outros sistemas de execução requer que estes cumpram os requisitos mínimos apresentados pelo sistema Apache ODE. Aliado a estes requisitos, é necessário desenvolver um sistema SAR, associado a cada sistema de execução, que utilize as suas interfaces programáticas proprietárias, para ser receptor de eventos e actualizar os recursos representativos das suas instâncias.

A abordagem proposta, segundo o modelo de integração considerado, reforça a dependência causada pela utilização de interfaces programáticas proprietárias. Existe necessidade de adoptar um modelo que promova a cooperação e uniformização, baseada em interfaces programáticas abertas, no sentido de criar soluções flexíveis, extensíveis e adaptáveis à evolução dos sistemas envolvidos. As arquitecturas orientadas a componentes, baseadas em arquitecturas orientadas a serviços, apresentam modelos que se enquadram nestas características e que devem ser considerados numa estratégia de integração de sistemas. A proposta de uma solução aberta conjuga as vantagens inerentes a estas arquitecturas com a característica não proprietária das interfaces programáticas envolvidas.

Considera-se que a concretização desta proposta e consequente implementação de um protótipo demonstrador representam um contributo na área de gestão integrada de processos WS-BPEL. O desenvolvimento efectuado promove a adopção de uma solução genérica, exequível sobre diversos sistemas de execução, contemplando conceitos de iniciativas de referência (OASIS, WfMC, OMG) e especificações abertas (WSRF, XML, WSDL). O aumento da dependência entre os sistemas de execução e gestão, a lacuna na abstracção dos detalhes de implementação destes sistemas, a redução da flexibilidade de adição/substituição dos sistemas envolvidos e da adaptabilidade da solução, para incorporação de novos requisitos ou alteração dos existentes, são algumas das limitações desta estratégia conservadora.

O desenvolvimento dos subsistemas constituintes do protótipo demonstrador e a demonstração de resultados permitiram concluir que é possível realizar a gestão integrada da execução de processos WS-BPEL, em diversos sistemas de execução, segundo o modelo de integração adoptado. Num quadro de evolução dos sistemas de execução, abandonando as especificações proprietárias para a adopção de normas abertas, o modelo de integração proposto pode manter-se inalterado. Para tornar solução desenvolvida funcional, assente neste modelo, é necessário adequar os adaptadores desenvolvidos para considerarem as novas interfaces programáticas.

6.1. Trabalho futuro

A adopção da hipótese de solução alternativa, enquadrada no conceito de *bus* aberto de serviços, é uma estratégia para considerar em trabalho futuro. Os objectivos seriam o desenvolvimento de uma especificação genérica e extensível para representar a informação de gestão/monitorização dos processos (possivelmente baseada no modelo de recurso apresentado neste projecto), subscrição de eventos não proprietários, adopção de mecanismos de descoberta de serviços, entre outras funcionalidades. Estes objectivos visam criar um modelo de integração *plug-and-play* para sistemas de execução e gestão.

A criação de uma norma genérica para troca de dados entre sistemas de execução e gestão, a adopção da norma por parte dos fabricantes destes sistemas, implicando o abandono das especificações proprietárias adequadas a cada sistema e a incorporação de mecanismos de descoberta de serviços na plataforma computacional da organização são algumas das problemáticas inerentes a esta estratégia.

Neste sentido, a continuidade de adopção de soluções *open source* e especificações abertas contribuem para a redução da dependência entre os sistemas envolvidos na execução e gestão de processos WS-BPEL. Para tal, sugere-se o aprofundamento do estudo do quadro de desenvolvimento WSRF, aliado às diversas especificações WS-*, de iniciativas de referência como a OASIS. A adopção da especificação WS-BaseNotification [12], desta iniciativa, para implementação de um sistema de publicação/subscrição de eventos/notificações sobre propriedades de recursos, é um dos exemplos.

Entre as sugestões, inclui-se o estudo e implementação de uma solução baseada na abordagem proposta no quadro WBEM (*Web-Based Enterprise Management*), da iniciativa DMTF (*Distributed Management Task Force*), usufruindo das vantagens enunciadas no capítulo “2.4 Monitorização de processos WS-BPEL”.

7. Referências

- [1] Tammo van Lessen, Frank Leymann, Ralph Mietzner, Jörg Nitzsche, Daniel Schleicher, "A Management Framework for WS-BPEL", Institute of Architecture of Application Systems, University of Stuttgart, 6th IEEE European Conference on Web Services, November 12-14, 2008, Dublin, Ireland, disponível em http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?isnumber=4711635&arnumber=4711662&count=33&index=26, visitado a 17-03-2009
- [2] Dave Chappell, "Enterprise Service Bus", O'Reilly, June 2004, ISBN 0-596-00675-6, pp 274
- [3] D. Jordan, J. Evdemon, A. Alves, A. Arkin, S. Askary, C. Barreto, A. Systems, B. Bloch, F. Curbera, M. Ford, A. Endpoints, Y. Goland, N. Kartha, S. Commerce, C.K. Liu, R. Khalaf, M. Marin, V. Mehta, S. Thatte, D.V. Rijn, P. Yendluri, A. Yiu, "Web Services Business Process Execution Language", Version 2.0, OASIS Standard, 11 April 2007, pp. 1-264, disponível em <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>, visitado a 18-06-2009
- [4] Dieter König, Charlton Barreto, "Web Services Business Process Execution Language Workshop", OASIS, disponível em <http://www.oasis-open.org/events/symposium/2007/slides/WS-BPELWorkshop.ppt>, visitado a 20-06-2009
- [5] Matjaz B. Juric, Benny Mathew , Poornachandra Sarang, "Business Process Execution Language for Web Services", Second Edition, January 2006, 350 pág., Packt Publishing, ISBN 1904811817
- [6] K. Czajkowski, Donald F. Ferguson, I. Foster, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke, W. Vambenepe, "The WS-Resource Framework," WSRF version 1.0, 2004, pp. 1-18, disponível em <http://www.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf>, visitado a 16-06-2009
- [7] I. Foster, J. Frey, S.G. (IBM), S. Tuecke, K. Czajkowski, D. Ferguson, F. Leymann, M. Nally, I. Sedukhin, D. Snelling, T. Storey, W. Vambenepe, S. Weerawarana, "Modeling Stateful Resources with Web Services,", version 1.1, 2004, disponível

- em <http://www.ibm.com/developerworks/library/ws-resource/ws-modelingresources.pdf>., visitado a 18-06-2009
- [8] Organization for the Advancement of Structured Information Standards (OASIS), disponível em <http://www.oasis-open.org/home/index.php>, visitado a 17-05-2009
- [9] The Globus Toolkit 4.0, "The resource approach to statefulness", disponível em: <http://gdp.globus.org/gt4-tutorial/multiplehtml/ch01s03.html>, visitado a 09-07-2009
- [10] S. Graham, A. Karmarkar, J. Mischkinsky, I. Robinson, I. Sedukhin, "Web Services Resource" (WS-Resource) 1.2, 1 April 2006, disponível em http://docs.oasis-open.org/wsr/wsr/ws_resource-1.2-spec-os.pdf, visitado a 15-06-2009
- [11] S. Graham, J. Treadwell, "Web Service Resource Properties" 1.2 (WS-ResourceProperties), OASIS Standard, 1 de Abril 2006, pp. 1-64, disponível em http://docs.oasis-open.org/wsr/wsr/ws_resource_properties-1.2-spec-os.pdf, visitado a 15-06-2009
- [12] S. Graham, D. Hull, B. Murray, T. Event-driven, "Web Services Base Notification 1.3 (WS-BaseNotification)," 1 October 2006, disponível em http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf, visitado a 15-06-2009
- [13] Dieter Koenig, "WS-BPEL Technical Overview for Developers and Architects", 2007, disponível em <http://bpel.xml.org/presentation-tech-overview>, OASIS Presentation, visitado a 25-11-2009
- [14] Prof. Dr. Frank Leymann, "The Business Value of WS-BPEL for Business Analysts and Managers", OASIS BPEL Webinar, March 12th, 2007
- [15] Workflow Management Coalition (WfMC), disponível em <http://www.wfmc.org/>, visitado a 20-05-2009
- [16] David Hollingsworth, The Workflow Management Coalition Specification, "The Workflow Reference Model", 19-01-1995, disponível em <http://www.wfmc.org/standards/docs/tc003v11.pdf>, visitado a 20-05-2009

- [17] Workflow Management Coalition, Terminology & Glossary, Document Number WFMC-TC-1011, Feb 99
- [18] Object Management Group (OMG), disponível em <http://www.omg.org/>, visitado a 26-06-2009
- [19] OMG BPM Consortium, "Business Process Management with OMG specifications", Fact Sheet, disponível em http://www.bpm-consortium.org/101807_BPM_Fact_Sheet.pdf, visitado a 18-08-2009
- [20] Windows Workflow Foundation (WF), disponível em <http://msdn.microsoft.com/en-us/netframework/aa663328.aspx>, visitado a 25-07-2009
- [21] Windows Workflow Foundation, "The Workflow Way: Understanding Windows Workflow Foundation", disponível em <http://msdn.microsoft.com/en-us/library/dd851337.aspx>, visitado a 22-07-2009
- [22] Distributed Management Task Force (DMTF), disponível em <http://www.dmtf.org/home>, visitado a 22-07-2009
- [23] Business Process Management Initiative (BPMI), disponível em <http://www.bpmi.org/>, visitado a 22-07-2009
- [24] Java, "Java Management Extensions (JMX) Specification, version 1.4," *Management*, 2006, pp. 1-290, disponível em http://java.sun.com/javase/6/docs/technotes/guides/jmx/JMX_1_4_specification.pdf, visitado a 26-07-2009
- [25] Sun JMX, "Getting Started with Java Management Extensions (JMX): Developing and Monitoring Solutions", disponível em <http://java.sun.com/developer/technicalArticles/J2SE/jmx.html>, visitado a 19-07-2009
- [26] Roy Thomas Fielding, "Architectural Styles and the Design of Network-based Software Architectures", Representational State Transfer, disponível em http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf, visitado a 20-06-2009

- [27] The Apache Software Foundation, *homepage*, disponível em <http://www.apache.org/>, visitado a 16-06-2009
- [28] Apache Tomcat, disponível em <http://tomcat.apache.org/>, visitado a 16-06-2009
- [29] Apache ODE, disponível em: <http://ode.apache.org/>, visitado a 18-06-2009
- [30] Apache ODE Architectural Overview, disponível em <http://ode.apache.org/architectural-overview.html>, visitado a 17-06-2009
- [31] Apache ODE Management API, disponível em <http://ode.apache.org/user-guide.html#UserGuide-ManagementAPI16>, visitado a 16-06-2009
- [32] Apache ODE Javadoc, disponível em <http://ode.apache.org/javadoc/>, visitado a 04 -07-2009
- [33] Axis2, disponível em <http://ws.apache.org/axis2/>, visitado a 18-06-2009
- [34] ASF JIRA, Apache's JIRA issue tracker, disponível em <http://issues.apache.org/jira/>, visitado a 20 Julho 2009
- [35] Simon Moser e Tammo van Lessen , “Developing, Deploying and Running a Hello World BPEL Process with Eclipse BPEL Designer and Apache ODE”, disponível em <http://people.apache.org/~vanto>HelloWorld-BPELDesignerAndODE.pdf>, visitado a 14 -08-2009
- [36] Eclipse BPEL Designer Plugin, disponível em <http://www.eclipse.org/bpel/>, visitado a 15-06-2009
- [37] IBM, “WebSphere Business Integration Server”, disponível em <http://www-01.ibm.com/software/integration/wbiserver/>, visitado a 15-06-2009
- [38] Active Endpoints, “ActiveBPEL”, disponível em: <http://www.activevos.com/community-open-source.php>, visitado a 17-06-2009

- [39] Open Enterprise Service Bus, Sun Microsystems/Java.Net, disponível em <https://open-esb.dev.java.net/>, visitado a 17-05-2009
- [40] jBoss, jBPM, disponível em: <http://www.jboss.com/products/jbpm/>, visitado a 19-06-2009
- [41] Java.Net, Swing Application Framework, disponível em <https://appframework.dev.java.net/>, visitado a 28-07-2009
- [42] OMG, “Unified Modeling Language (OMG UML)”, Infrastructure, disponível em <http://www.omg.org/technology/documents/formal/uml.htm>, visitado a 03-12-2009
- [43] OMG, “MDA Guide Version 1.0.1”, 12-06-2003, disponível em <http://www.omg.org/mda/>, visitado a 03-12-2009
- [44] Workflow Management Coalition, “Process Definition Interface, XML Process Definition Language”, Workflow Standard, Document Number WFMC-TC-1025, October 10, 2008, Version 2.1a, disponível em <http://www.wfmc.org/View-document-details/WFMC-TC-1025-Oct-10-08-A-Final-XPDL-2.1-Specification.html>, visitado a 15-12-2009
- [45] D. Trowbridge, U. Roxburgh, G. Hohpe, D. Manolescu, E.G. Nadhan, Microsoft Integration Patterns, patterns and practices, 2004
- [46] Linthicum David S., Enterprise Application Integration, Addison-Wesley InformationTechnology Series, 2000, ISBN 0-201-61583-5
- [47] B. Michael, B. Henning, B. Dave, D. Jean-Jacques, C. Adrian, E. Mike, F. Don, F. Bill, G. Mike, K. Dan, M. Jim, M. Jeff, N. Martin, P. Greg, R. Mike, T. Ken, T. Carl, Service Component Architecture – Building Systems using a Service Oriented Architecture, Joint Whitepaper by BEA, IBM, Interface21, IONA, Oracle, SAP, Siebel, Sybase, Novembro 2005, disponível em http://www.oracle.com/technology/tech/webservices/standards/sca/pdf/SCA_White_Paper1_09.pdf, visitado a 07-09-2009
- [48] S. Shazia, I. Marta, B. Wasana, C. Sandy, Major Issues in Business Process Management: A Vendor Perspective, disponível em <http://www.pacis-net.org/file/2007/1336.pdf>, visitado a 03-02-2010

- [49] B. David, H. Hugo, M. Francis, N. Eric, C. Michael, F. Chris, O. David, Web Services Architecture, W3C Working Group Note, 11-02-2004, disponível em <http://www.w3.org/TR/ws-arch/>, visitado a 14-12-2009
- [50] Accenture, Ariba, Inc., Commerce One, Inc., Fujitsu Limited, Hewlett-Packard Company, i2 Technologies, Inc, Intel Corporation, International Business Machines Corporation, Microsoft Corporation, OracleCorporation, SAP AG, Sun Microsystems, Inc. e VeriSign, Inc, UDDI Executive White Paper, uddi.org, 14-11-2001, disponível em http://www.uddi.org/pubs/UDDI_Executive_White_Paper.pdf, visitado a 16-12-2009
- [51] F. Donald, S. Marcia, Enterprise Business Process Management – Architecture, Technology and Standards, OSOA WhitePaper, disponível em <http://www.osoa.org/download/attachments/250/Enterprise+Business+Process+Management.pdf>, visitado a 17-11-2009
- [52] J. Case, M. Fedor, M. Schoffstall, J. Darvin, A Simple Network Management Protocol (SNMP), RFC 1157 Network Working Group, disponível em <http://www.javvin.com/protocol/rfc1157.pdf>, visitado a 26-07-2009
- [53] OMG, Business Process Model and Notation, OMG Specification, versão 1.2, 03-01-2009, disponível em <http://www.omg.org/spec/BPMN/1.2/PDF/>, visitado a 30 Junho 2009
- [54] Apache ServiceMix, The Apache Software Foundation, disponível em <http://servicemix.apache.org/home.html>, visitado a 12-04-2009

8. Apêndices

8.1. Apêndice A – Ficheiro de definição de um recurso

Ficheiro XML *Schema* constituinte do Sistema Gerador de Recursos onde são definidas as propriedades de um recurso representativo de um processo WS-BPEL.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://isel.deetc.meic/schema/BpelProcessRepresentat
ion"
xmlns:tns="http://isel.deetc.meic/schema/BpelProcessRepresentation"
elementFormDefault="qualified">

  <xsd:element name="bpelProcess">
    <xsd:complexType>
      <xsd:all>
        <xsd:element name="bpelInstanceInfo">
          <xsd:complexType>
            <xsd:attribute default="empty" name="resourcePropertyId"
              type="xsd:string"/>
            <xsd:attribute default="empty" name="instanceId"
              type="xsd:string"/>
            <xsd:attribute default="empty" name="processName"
              type="xsd:string"/>
            <xsd:attribute default="empty" name="status"
              type="xsd:string"/>
            <xsd:attribute default="empty" name="dateStarted"
              type="xsd:string"/>
            <xsd:attribute default="empty" name="dateLastActive"
              type="xsd:string"/>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="bpelVariables">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="bpelVariable" maxOccurs="unbounded">
                <xsd:complexType>
                  <xsd:attribute default="empty"
                    name="resourcePropertyId" type="xsd:string"/>
                  <xsd:attribute default="empty"
                    name="name" type="xsd:string"/>
                  <xsd:attribute default="empty"
                    name="type" type="xsd:string"/>
                  <xsd:attribute default="uninitialized"
                    name="state" type="xsd:string"/>
                  <xsd:attribute default="empty"
                    name="value" type="xsd:string"/>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:all>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

<xsd:element name="bpelSequence">
  <xsd:complexType>
    <xsd:attribute default="empty"
      name="name" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="dateStarted" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="dateEnded" type="xsd:string"/>
    <xsd:attribute default="inactive"
      name="status" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="resourcePropertyId" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="bpelReceive">
  <xsd:complexType>
    <xsd:attribute default="empty"
      name="resourcePropertyId" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="name" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="dateStarted" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="dateEnded" type="xsd:string"/>
    <xsd:attribute default="inactive"
      name="status" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="createInstance" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="operation" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="partnerLink" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="portType" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="variable" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="bpelAssign">
  <xsd:complexType>
    <xsd:attribute default="empty"
      name="resourcePropertyId" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="name" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="dateStarted" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="dateEnded" type="xsd:string"/>
    <xsd:attribute default="inactive"
      name="status" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="validate" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

```

```

<xsd:element name="bpelReply">
  <xsd:complexType>
    <xsd:attribute default="empty"
      name="resourcePropertyId" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="name" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="dateStarted" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="dateEnded" type="xsd:string"/>
    <xsd:attribute default="inactive"
      name="status" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="operation" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="partnerLink" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="portType" type="xsd:string"/>
    <xsd:attribute default="empty"
      name="variable" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
</xsd:all>
<xsd:attribute default="empty"
  name="resourcePropertyId" type="xsd:string"/>
<xsd:attribute default="empty"
  name="name" type="xsd:string"/>
<xsd:attribute default="empty"
  name="processId" type="xsd:string"/>
<xsd:attribute default="empty"
  name="displayName" type="xsd:string"/>
<xsd:attribute default="empty"
  name="targetNamespace" type="xsd:string"/>
<xsd:attribute default="empty"
  name="exitOnStandardFault" type="xsd:string"/>
<xsd:attribute default="empty"
  name="executionMode" type="xsd:string"/>
<xsd:attribute default="empty"
  name="continueOnError" type="xsd:string"/>
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

Listagem 16 – Ficheiro de definição de um recurso (*BpelProcessRepresentation*)

8.2. Apêndice B – Ficheiro de configuração do Sistema Gerador de Recursos

Ficheiro “config.xml”, constituinte do Sistema Gerador de Recursos, onde são definidas as configurações de execução da aplicação resultante deste sistema.

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <generic>
    <dirWatcherPath>
      C:\\BpelProcessDeployFolder\\
    </dirWatcherPath>
    <fileWatcherExtension>.bpel</fileWatcherExtension>
    <bpelResourceSchemaFilePath>
      C:\\BpelResourceGenerator\\BpelProcessRepresentation.xsd
    </bpelResourceSchemaFilePath>
    <bpelResourceFileDirectoryPath>
      C:\\BpelResourceGenerator\\GeneratedResourceFiles\\
    </bpelResourceFileDirectoryPath>
  </generic>

  <bpelResourceNameAssociation>
    <bpws:process>bpelProcess</bpws:process>
    <bpws:variables>bpelVariables</bpws:variables>
    <bpws:variable>bpelVariable</bpws:variable>
    <bpws:sequence>bpelSequence</bpws:sequence>
    <bpws:receive>bpelReceive</bpws:receive>
    <bpws:assign>bpelAssign</bpws:assign>
    <bpws:reply>bpelReply</bpws:reply>
  </bpelResourceNameAssociation>

  <bpelResourceElementXPathExpr>
    <bpelProcess>
      //element[@name='bpelProcess']/complexType/attribute
    </bpelProcess>
    <bpelInstanceInfo>
      //element[@name='bpelInstanceInfo']/complexType/attribute
    </bpelInstanceInfo>
    <bpelVariables>//bpelVariables/@*</bpelVariables>
    <bpelVariable>
      //element[@name='bpelVariable']/complexType/attribute
    </bpelVariable>
    <bpelSequence>
      //element[@name='bpelSequence']/complexType/attribute
    </bpelSequence>
    <bpelReceive>
      //element[@name='bpelReceive']/complexType/attribute
    </bpelReceive>
    <bpelAssign>
      //element[@name='bpelAssign']/complexType/attribute
    </bpelAssign>
    <bpelReply>
      //element[@name='bpelReply']/complexType/attribute
    </bpelReply>
  </bpelResourceElementXPathExpr>
</config>
```

Listagem 17 – Ficheiro de configuração do Sistema Gerador de Recursos

8.3. Apêndice C – Exemplo de ficheiro recurso (original)

Ficheiro XML gerado pelo Sistema Gerador de Recursos, que constitui o recurso representativo do processo WS-BPEL *Loan Request*.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<bpelProcess continueOnError="empty" displayName="empty"
  executionMode="empty" exitOnStandardFault="yes"
  name="LoanRequestProcess" processId="empty"
  resourcePropertyId="1"
  targetNamespace="http://loanrequest">
  <bpelInstanceInfo dateLastActive="empty"
    dateStarted="empty" instanceId="empty"
    processName="empty" resourcePropertyId="2"
    status="empty"/>
  <bpelVariables>
    <bpelVariable name="input" resourcePropertyId="12"
      state="uninitialized" type="empty" value="empty"/>
    <bpelVariable name="output" resourcePropertyId="13"
      state="uninitialized" type="empty" value="empty"/>
    <bpelVariable name="RiskDeciderResponse"
      resourcePropertyId="14" state="uninitialized"
      type="empty" value="empty"/>
    <bpelVariable name="RiskDeciderRequest"
      resourcePropertyId="15" state="uninitialized"
      type="empty" value="empty"/>
    <bpelVariable name="LoanMakerResponse"
      resourcePropertyId="16" state="uninitialized"
      type="empty" value="empty"/>
    <bpelVariable name="LoanMakerRequest"
      resourcePropertyId="17" state="uninitialized"
      type="empty" value="empty"/>
  </bpelVariables>
  <bpelSequence dateEnded="empty" dateStarted="empty"
    name="main" resourcePropertyId="18" status="inactive">
    <bpelReceive createInstance="yes" dateEnded="empty"
      dateStarted="empty" name="receiveInput"
      operation="process" partnerLink="client"
      portType="tns:LoanRequestProcess"
      resourcePropertyId="19" status="inactive"
      variable="input"/>
    <bpelAssign dateEnded="empty" dateStarted="empty"
      name="Assign" resourcePropertyId="21"
      status="inactive" validate="no"/>
    <bpelSequence dateEnded="empty" dateStarted="empty"
      name="Sequence" resourcePropertyId="35"
      status="inactive">
      <bpelAssign dateEnded="empty" dateStarted="empty"
        name="Assign1" resourcePropertyId="36"
        status="inactive" validate="no"/>
      <bpelAssign dateEnded="empty" dateStarted="empty"
        name="Assign2" resourcePropertyId="49"
        status="inactive" validate="no"/>
    </bpelSequence>
  </bpelSequence>
</bpelProcess>
```

```

<bpelSequence dateEnded="empty" dateStarted="empty"
  name="Sequence1" resourcePropertyId="62"
  status="inactive">
  <bpelAssign dateEnded="empty" dateStarted="empty"
    name="Assign5" resourcePropertyId="63"
    status="inactive" validate="no"/>
  <bpelAssign dateEnded="empty" dateStarted="empty"
    name="Assign3" resourcePropertyId="76"
    status="inactive" validate="no"/>
</bpelSequence>
<bpelAssign dateEnded="empty" dateStarted="empty"
  name="Assign4" resourcePropertyId="90"
  status="inactive" validate="no"/>
<bpelReply dateEnded="empty" dateStarted="empty"
  name="replyOutput" operation="process"
  partnerLink="client" portType="tns:LoanRequestProcess"
  resourcePropertyId="102" status="inactive"
  variable="output"/>
</bpelSequence>
</bpelProcess>

```

Listagem 18 – Ficheiro recurso do processo WS-BPEL *LoanRequest*

8.4. Apêndice D – Exemplo de ficheiro recurso (após execução)

Ficheiro XML resultante do Sistema Gerador de Recursos, que define o recurso representativo do processo WS-BPEL *Loan Request*, actualizado pelo Sistema de Actualização de Recursos após a execução do respectivo processo.

```
<?xml version="1.0" encoding="UTF-8"?>
<bpelProcess continueOnError="empty" displayName="empty"
  executionMode="empty" exitOnStandardFault="yes"
  name="LoanRequestProcess" processId="empty"
  resourcePropertyId="1"
  targetNamespace="http://loanrequest">
  <bpelInstanceInfo dateLastActive="2009-09-06T19:20:09.406+01:00"
    dateStarted="2009-09-06T19:20:09.343+01:00"
    instanceId="59402" processName="LoanRequestProcess"
    resourcePropertyId="2" status="ACTIVE"/>
  <bpelVariables>
    <bpelVariable name="input" resourcePropertyId="12"
      state="initialized" type="empty" value="5000"/>
    <bpelVariable name="output" resourcePropertyId="13"
      state="initialized" type="empty"
      value="Emprestimo realizado"/>
    <bpelVariable name="RiskDeciderResponse"
      resourcePropertyId="14" state="uninitialized"
      type="empty" value="sem risco"/>
    <bpelVariable name="RiskDeciderRequest" resourcePropertyId="15"
      state="uninitialized" type="empty" value="5000"/>
    <bpelVariable name="LoanMakerResponse" resourcePropertyId="16"
      state="uninitialized" type="empty"
      value="Emprestimo realizado"/>
    <bpelVariable name="LoanMakerRequest" resourcePropertyId="17"
      state="initialized" type="empty" value="5000"/>
  </bpelVariables>
  <bpelSequence dateEnded="2009-09-06 23:59:48"
    dateStarted="2009-09-06 23:59:47" name="main"
    resourcePropertyId="18" status="COMPLETED">
    <bpelReceive createInstance="yes" dateEnded="2009-09-06 23:59:47"
      dateStarted="2009-09-06 23:59:47" name="receiveInput"
      operation="process" partnerLink="client"
      portType="tns:LoanRequestProcess" resourcePropertyId="19"
      status="COMPLETED" variable="input"/>
    <bpelAssign dateEnded="2009-09-06 23:58:49"
      dateStarted="2009-09-06 23:58:49" name="Assign"
      resourcePropertyId="21" status="COMPLETED" validate="no"/>
    <bpelSequence dateEnded="2009-09-06 23:59:47"
      dateStarted="2009-09-06 23:59:47" name="Sequence"
      resourcePropertyId="35" status="COMPLETED">
      <bpelAssign dateEnded="2009-09-06 23:59:47"
        dateStarted="2009-09-06 23:59:47" name="Assign1"
        resourcePropertyId="36" status="COMPLETED" validate="no"/>
      <bpelAssign dateEnded="2009-09-06 23:59:47"
        dateStarted="2009-09-06 23:59:47" name="Assign2"
        resourcePropertyId="49" status="COMPLETED" validate="no"/>
    </bpelSequence>
  </bpelSequence>
</bpelProcess>
```

```

<bpelSequence dateEnded="2009-09-06 23:59:22"
  dateStarted="2009-09-06 23:59:22" name="Sequence1"
  resourcePropertyId="62" status="COMPLETED">
  <bpelAssign dateEnded="2009-09-06 23:59:22"
    dateStarted="2009-09-06 23:59:22" name="Assign5"
    resourcePropertyId="63" status="COMPLETED" validate="no"/>
  <bpelAssign dateEnded="2009-09-06 23:59:22"
    dateStarted="2009-09-06 23:59:22" name="Assign3"
    resourcePropertyId="76" status="COMPLETED" validate="no"/>
</bpelSequence>
<bpelAssign dateEnded="2009-09-06 23:59:47"
  dateStarted="2009-09-06 23:59:47" name="Assign4"
  resourcePropertyId="90" status="COMPLETED" validate="no"/>
<bpelReply dateEnded="2009-09-06 23:59:48"
  dateStarted="2009-09-06 23:59:48" name="replyOutput"
  operation="process" partnerLink="client"
  portType="tns:LoanRequestProcess" resourcePropertyId="102"
  status="COMPLETED" variable="output"/>
</bpelSequence>
</bpelProcess>

```

Listagem 19 – Ficheiro recurso do processo WS-BPEL *Loan Request* (após actualização)

8.5. Apêndice E – Ficheiro de configuração do Sistema de Actualização de Recursos

Ficheiro “config.xml”, constituinte do Sistema de Actualização de Recursos, onde são definidas as configurações para as funcionalidades deste sistema.

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <logFilePath>
    C:\\ResourceRepository\\Log\\log.txt
  </logFilePath>
  <generatedResourceFilesPath>
    C:\\BpelResourceGenerator\\GeneratedResourceFiles\\
  </generatedResourceFilesPath>
  <resourceRepositoryPath>
    C:\\ResourceRepository\\
  </resourceRepositoryPath>

  <resourceXPathUpdateExpr>
    <updateInstanceInfo>
      //bpelInstanceInfo
    </updateInstanceInfo>
    <updateVariableInfo>
      //bpelVariable[@name="varName"]
    </updateVariableInfo>
    <updateActivityInfo>
      //*[@name="activityName"]
    </updateActivityInfo>
  </resourceXPathUpdateExpr>
</config>
```

Listagem 20 – Ficheiro de configuração do Sistema de Actualização de Recursos

8.6. Apêndice F – Ficheiro WS-BPEL do processo de negócio *Loan Request*

Ficheiro "LoanRequestProcess.bpel" com a definição WS-BPEL do processo *Loan Request*. Este processo foi criado para demonstração da solução.

```
<?xml version="1.0" encoding="UTF-8"?>
<bpws:process exitOnStandardFault="yes"
  name="LoanRequestProcess"
  suppressJoinFailure="yes"
  targetNamespace="http://loanrequest"
  xmlns:bpws="http://docs.oasis-
open.org/wsbpel/2.0/process/executable"
  xmlns:ns="http://loanrequestArtifacts"
  xmlns:ns0="http://riskdecider" xmlns:ns1="http://loanmaker"
  xmlns:tns="http://loanrequest">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
    location="LoanRequestProcess.wsdl"
    namespace="http://loanrequest"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
    location="RiskDeciderWS.wsdl" namespace="http://riskdecider"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
    location="LoanRequestProcessArtifacts.wsdl"
    namespace="http://loanrequestArtifacts"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
    location="LoanMakerWS.wsdl" namespace="http://loanmaker"/>
  <bpws:partnerLinks>
    <bpws:partnerLink myRole="LoanRequestProcessProvider"
      name="client" partnerLinkType="tns:LoanRequestProcess"/>
    <bpws:partnerLink name="RiskDecider"
      partnerLinkType="ns:RiskDeciderWSLT"
      partnerRole="RiskDeciderWSPort"/>
    <bpws:partnerLink name="LoanMaker"
      partnerLinkType="ns:LoanMakerWSLT"
      partnerRole="LoanMakerWSPort"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable
      messageType="tns:LoanRequestProcessRequestMessage"
      name="input"/>
    <bpws:variable
      messageType="tns:LoanRequestProcessResponseMessage"
      name="output"/>
    <bpws:variable messageType="ns0:AvaliateRiskResponse"
      name="RiskDeciderResponse"/>
    <bpws:variable messageType="ns0:AvaliateRiskRequest"
      name="RiskDeciderRequest"/>
    <bpws:variable messageType="ns1:MakeLoanResponse"
      name="LoanMakerResponse"/>
    <bpws:variable messageType="ns1:MakeLoanRequest"
      name="LoanMakerRequest"/>
  </bpws:variables>
  <bpws:sequence name="main">
    <bpws:receive createInstance="yes" name="receiveInput"
      operation="process" partnerLink="client"
      portType="tns:LoanRequestProcess" variable="input"/>
    <bpws:if name="DecideRisk">
```

```

<bpws:assign name="Assign" validate="no">
  <bpws:copy>
    <bpws:from>
      <bpws:literal>
        <tns:LoanRequestProcessResponse
          xmlns:tns="http://loanrequest">
          <tns:result/>
        </tns:LoanRequestProcessResponse>
      </bpws:literal>
    </bpws:from>
    <bpws:to part="payload" variable="output"/>
  </bpws:copy>
  <bpws:copy>
    <bpws:from>
      <bpws:literal>sem_risco</bpws:literal>
    </bpws:from>
    <bpws:to part="payload" variable="output">
      <bpws:query queryLanguage="urn:oasis:names:tc:
        wsbpel:2.0:sublang:xpath1.0">
        <![CDATA[/tns:result]]></bpws:query>
      </bpws:to>
    </bpws:copy>
  </bpws:assign>
  <bpws:condition>
    <![CDATA[($input.payload/tns:input) < 10000]]>
  </bpws:condition>
  <bpws:else>
    <bpws:sequence name="Sequence">
      <bpws:assign name="Assign1" validate="no">
        <bpws:copy>
          <bpws:from>
            <bpws:literal>
              <tns:AvaliateRisk xmlns:tns="http://riskdecider">
                <tns:amount/>
              </tns:AvaliateRisk>
            </bpws:literal>
          </bpws:from>
          <bpws:to part="parameters"
            variable="RiskDeciderRequest"/>
        </bpws:copy>
        <bpws:copy>
          <bpws:from part="payload" variable="input">
            <bpws:query queryLanguage="urn:oasis:names:tc:
              wsbpel:2.0:sublang:xpath1.0">
              <![CDATA[/tns:input]]></bpws:query>
            </bpws:from>
            <bpws:to part="parameters"
              variable="RiskDeciderRequest">
              <bpws:query queryLanguage="urn:oasis:names:tc:
                wsbpel:2.0:sublang:xpath1.0">
                <![CDATA[/ns0:amount]]></bpws:query>
            </bpws:to>
          </bpws:copy>
        </bpws:assign>

```

```

<bpws:invoke inputVariable="RiskDeciderRequest"
  name="InvokeRiskDecider" operation="AvaliateRisk"
  outputVariable="RiskDeciderResponse"
  partnerLink="RiskDecider"
  portType="ns0:RiskDeciderWSPortType"/>
<bpws:assign name="Assign2" validate="no">
  <bpws:copy>
    <bpws:from>
      <bpws:literal>
        <tns:LoanRequestProcessResponse
          xmlns:tns="http://loanrequest">
          <tns:result/>
        </tns:LoanRequestProcessResponse>
      </bpws:literal>
    </bpws:from>
    <bpws:to part="payload" variable="output"/>
  </bpws:copy>
  <bpws:copy>
    <bpws:from part="parameters"
      variable="RiskDeciderResponse">
      <bpws:query queryLanguage="urn:oasis:names:tc:
        wsbpel:2.0:sublang:xpath1.0">
        <![CDATA[/ns0:return]]></bpws:query>
      </bpws:from>
      <bpws:to part="payload" variable="output">
      <bpws:query queryLanguage="urn:oasis:names:tc:
        wsbpel:2.0:sublang:xpath1.0">
        <![CDATA[/tns:result]]></bpws:query>
      </bpws:to>
    </bpws:copy>
  </bpws:assign>
</bpws:sequence>
</bpws:else>
</bpws:if>
<bpws:if name="AnalyseRisk">
  <bpws:sequence name="Sequence1">
    <bpws:assign name="Assign5" validate="no">
      <bpws:copy>
        <bpws:from>
          <bpws:literal>
            <nsl:MakeLoan xmlns:nsl="http://loanmaker">
              <nsl:amount/>
            </nsl:MakeLoan>
          </bpws:literal>
        </bpws:from>
        <bpws:to part="parameters"
          variable="LoanMakerRequest"/>
      </bpws:copy>
      <bpws:copy>
        <bpws:from part="payload" variable="input">
          <bpws:query queryLanguage="urn:oasis:names:tc:
            wsbpel:2.0:sublang:xpath1.0">
            <![CDATA[/tns:input]]></bpws:query>
          </bpws:from>

```

```

        <bpws:to part="parameters"
            variable="LoanMakerRequest">
            <bpws:query queryLanguage="urn:oasis:names:tc:
                wsbpel:2.0:sublang:xpath1.0">
                <![CDATA[/ns1:amount]]></bpws:query>
            </bpws:to>
        </bpws:copy>
    </bpws:assign>
    <bpws:invoke inputVariable="LoanMakerRequest"
        name="InvokeLoanMaker" operation="MakeLoan"
        outputVariable="LoanMakerResponse"
        partnerLink="LoanMaker"
        portType="ns1:LoanMakerWSPortType"/>
    <bpws:assign name="Assign3" validate="no">
        <bpws:copy>
            <bpws:from>
                <bpws:literal>
                    <tns:LoanRequestProcessResponse
                        xmlns:tns="http://loanrequest">
                        <tns:result/>
                    </tns:LoanRequestProcessResponse>
                </bpws:literal>
            </bpws:from>
            <bpws:to part="payload" variable="output"/>
        </bpws:copy>
        <bpws:copy>
            <bpws:from part="parameters"
                variable="LoanMakerResponse">
                <bpws:query queryLanguage="urn:oasis:names:tc:
                    wsbpel:2.0:sublang:xpath1.0">
                    <![CDATA[/ns1:return]]></bpws:query>
            </bpws:from>
            <bpws:to part="payload" variable="output">
                <bpws:query queryLanguage="urn:oasis:names:tc:
                    wsbpel:2.0:sublang:xpath1.0">
                    <![CDATA[/tns:result]]></bpws:query>
            </bpws:to>
        </bpws:copy>
    </bpws:assign>
</bpws:sequence>
<bpws:condition>
    <![CDATA[( $output.payload/tns:result) != "arriscado"]]>
</bpws:condition>
<bpws:else>
    <bpws:assign name="Assign4" validate="no">
        <bpws:copy>
            <bpws:from>
                <bpws:literal>
                    <tns:LoanRequestProcessResponse
                        xmlns:tns="http://loanrequest">
                        <tns:result/>
                    </tns:LoanRequestProcessResponse>
                </bpws:literal>
            </bpws:from>
        </bpws:copy>
    </bpws:assign>
</bpws:else>

```

```

<bpws:assign name="Assign4" validate="no">
  <bpws:copy>
    <bpws:from>
      <bpws:literal>
        <tns:LoanRequestProcessResponse
xmlns:tns="http://loanrequest">
          <tns:result/>
        </tns:LoanRequestProcessResponse>
      </bpws:literal>
    </bpws:from>
    <bpws:to part="payload" variable="output"/>
  </bpws:copy>
  <bpws:copy>
    <bpws:from>
      <bpws:literal>Empréstimo não realizado
    </bpws:literal>
    </bpws:from>
    <bpws:to part="payload" variable="output">
      <bpws:query queryLanguage="urn:oasis:names:tc:
wsbpel:2.0:sublang:xpath1.0">
        <![CDATA[/tns:result]]></bpws:query>
      </bpws:to>
    </bpws:copy>
  </bpws:assign>
</bpws:else>
</bpws:if>
<bpws:reply name="replyOutput" operation="process"
partnerLink="client" portType="tns:LoanRequestProcess"
variable="output"/>
</bpws:sequence>
</bpws:process>

```

Listagem 21 – Ficheiro WS-BPEL de definição do processo *Loan Request*

8.7. Apêndice G – Casos de utilização do Sistema Gerador de Recursos

Diagrama de casos de utilização:

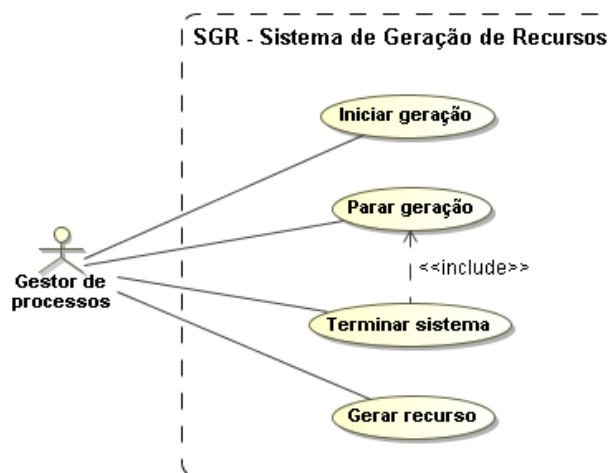


Figura 69 – Diagrama de casos de utilização do Sistema Gerador de Recursos

Descrição dos casos de utilização:

Descrição de caso de utilização			
Designação	Iniciar geração		
Resumo	Iniciar a geração automática de recursos (criar receptor de processos existentes no sistema de execução)		
Pré-condições	Nenhuma		
Actores	Gestor de processos		
Cenário Principal			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	O caso de utilização inicia-se com o actor a pressionar o botão "Iniciar"	
	2		O sistema cria um receptor sobre a pasta de inserção de processos WS-BPEL no sistema de execução.
	3		O sistema apresenta a mensagem de sistema iniciado com sucesso.
	4	O caso de utilização termina	
Cenário Alternativo 1			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1		No passo 2 do cenário principal, ocorre um erro na criação do receptor e o sistema não pode ser iniciado.
	2		O sistema apresenta uma mensagem de erro
	3	O caso de utilização termina	

Tabela 9 – Descrição do caso de utilização *Iniciar geração* (SGR)

Descrição de caso de utilização			
Designação	Parar geração		
Resumo	Suspender a geração automática de recursos (remoção do receptor de processos existentes no sistema de execução)		
Pré-condições	A geração automática de recursos está iniciada		
Actores	Gestor de processos		
Cenário Principal			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	O caso de utilização inicia-se com o actor a pressionar o botão “Parar”	
	2		O sistema remove o receptor criado sobre a pasta de inserção de processos
	3		O sistema apresenta a mensagem de sistema terminado com sucesso.
	4	O caso de utilização termina	
Cenário Alternativo 1			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1		No passo 2 do cenário principal, ocorre um erro na remoção do receptor e a geração automática permanece iniciada.
	2		O sistema apresenta uma mensagem de erro
	3	O caso de utilização termina	

Tabela 10 – Descrição do caso de utilização *Parar geração*

Descrição de caso de utilização			
Designação	Gerar recurso		
Resumo	Geração de um recurso a partir da definição de um processo existente no sistema de execução		
Pré-condições	A geração automática de recursos está iniciada		
Actores	Gestor de processos		
Cenário Principal			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	O caso de utilização inicia-se com o actor a inserir o ficheiro com a definição do processo na pasta de inserção do sistema de execução	
	2		O sistema detecta o novo ficheiro de processo inserido
	3		O sistema analisa a definição WS-BPEL do processo e gera a sua representação em recurso
	4		O sistema apresenta a mensagem de recurso gerado com sucesso

	5		O sistema apresenta informação adicional relativa ao nome do recurso gerado
	6	O caso de utilização termina	
Cenário Alternativo 1			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1		No passo 3 do cenário principal, ocorre um erro na geração do recurso
	2		O sistema apresenta uma mensagem de erro
	3	O caso de utilização termina	

Tabela 11 – Descrição do caso de utilização *Gerar recurso*

Descrição de caso de utilização			
Designação	Terminar sistema		
Resumo	Terminar o sistema SGR		
Pré-condições	Nenhuma		
Actores	Gestor de processos		
Cenário Principal			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	O caso de utilização inicia-se com o actor a pressionar o botão “Sair” no Menu ou no botão de fecho da janela	
	2		O sistema apresenta uma mensagem de confirmação de saída
	3	O actor pressiona o botão “Yes”	
	4		O sistema termina a execução
	5	O caso de utilização termina	
Cenário Alternativo 1			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	No passo 3 do cenário principal, o actor pressiona o botão “No”	
	2		O sistema mantém-se em execução
	3	O caso de utilização termina	
Cenário Alternativo 2			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1		No passo 4 do cenário principal, o sistema de geração está iniciado.
	2		Incluir Parar Geração , a partir do passo 1 do cenário principal
	3	O caso de utilização termina	

Tabela 12 – Descrição do caso de utilização *Terminar sistema*

8.8. Apêndice H – Casos de utilização da interface de gestão

O diagrama de casos de utilização e a respectiva descrição de cada um aplicam-se aos sistemas descritos nos capítulos “4.3.3 Serviço de Gestão” e “4.3.4 Aplicação de Gestão”, com excepção dos casos de utilização *Seleccionar processo* e *Seleccionar instância*, com relevância apenas na Aplicação de Gestão.

Diagrama de casos de utilização:

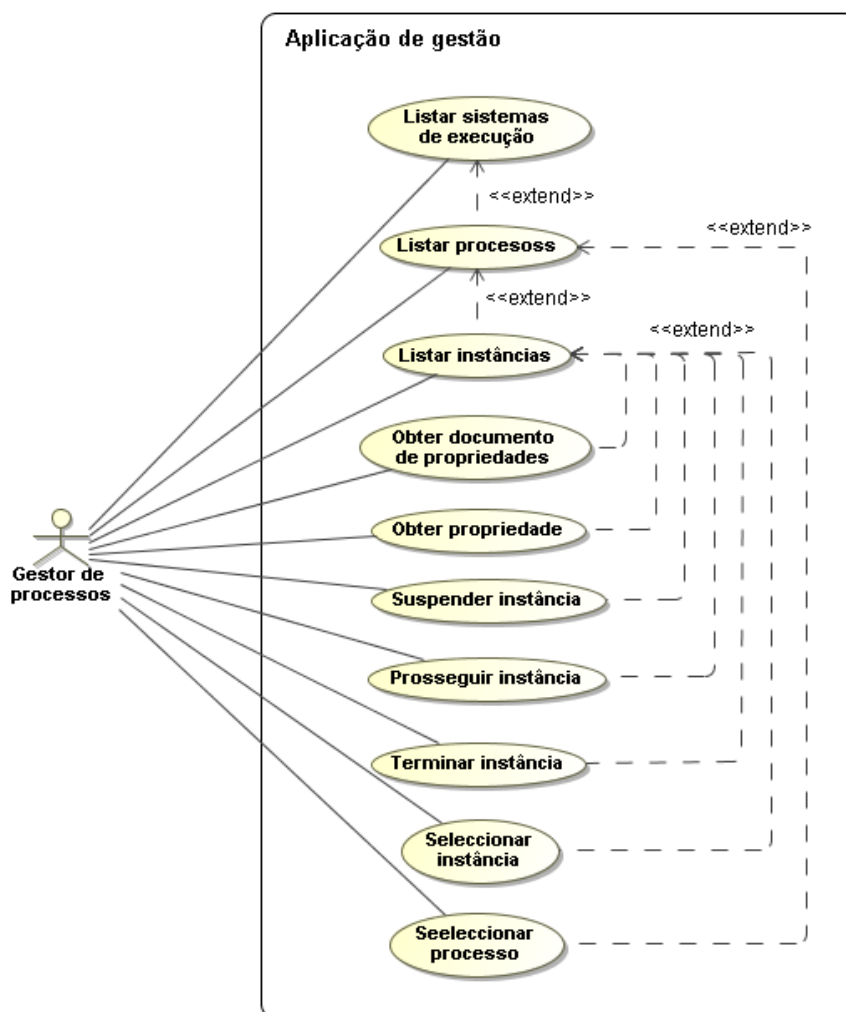


Figura 70 – Diagrama de casos de utilização da interface de gestão

Descrição dos casos de utilização:

Descrição de caso de utilização	
Designação	Listar sistemas de execução
Resumo	Obter a lista de sistemas de execução cujos processos podem ser geridos/monitorizados
Pré-condições	Nenhuma
Actores	Gestor de processos

Cenário Principal			
	#	Input do Actor	Comportamento do sistema
Sequência de acções	1	O caso de utilização inicia-se com o actor a pressionar o botão “Listar sistemas de execução”	
	2		O sistema obtém a lista de sistemas de execução (utiliza o Serviço de Gestão)
	3		O sistema apresenta a listagem dos sistemas de execução
	4	O caso de utilização termina	
Cenário Alternativo 1			
	#	Input do Actor	Comportamento do sistema
Sequência de acções	1		No passo 2 do cenário principal, ocorre um erro na obtenção dos sistemas de execução
	2		O sistema apresenta uma mensagem de erro
	3	O caso de utilização termina	

Tabela 13 – Descrição do caso de utilização *Listar sistemas de execução*

Descrição de caso de utilização			
Designação	Listar processos		
Resumo	Obter a lista de processos existentes no sistema de execução		
Pré-condições	Está seleccionado um sistema de execução		
Actores	Gestor de processos		
Cenário Principal			
	#	Input do Actor	Comportamento do sistema
Sequência de acções	1	O caso de utilização inicia-se com o actor a pressionar o botão “Listar processos”	
	2		O sistema obtém a lista de processos (utiliza o Serviço de Gestão)
	3		O sistema apresenta a listagem dos processos
	4	O caso de utilização termina	
Cenário Alternativo 1			
	#	Input do Actor	Comportamento do sistema
Sequência de acções	1		No passo 2 do cenário principal, ocorre um erro na obtenção dos processos
	2		O sistema apresenta uma mensagem de erro
	3	O caso de utilização termina	

Tabela 14 – Descrição do caso de utilização *Listar processos*

Descrição de caso de utilização			
Designação	Listar instâncias		
Resumo	Obter a lista de instâncias do processo no sistema de execução		
Pré-condições	Está seleccionado um processo		
Actores	Gestor de processos		
Cenário Principal			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	O caso de utilização inicia-se com o actor a pressionar o botão “Listar instâncias”	
	2		O sistema obtém a lista de instâncias do processo (utiliza o Serviço de Gestão)
	3		O sistema apresenta a listagem das instâncias
	4	O caso de utilização termina	
Cenário Alternativo 1			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1		No passo 2 do cenário principal, ocorre um erro na obtenção das instâncias
	2		O sistema apresenta uma mensagem de erro
	3	O caso de utilização termina	

Tabela 15 – Descrição do caso de utilização *Listar instâncias*

Descrição de caso de utilização			
Designação	Obter documento de propriedades		
Resumo	Obter o documento de propriedades de um recurso		
Pré-condições	Está seleccionada uma instância		
Actores	Gestor de processos		
Cenário Principal			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	O caso de utilização inicia-se com o actor a pressionar o botão “Obter documento de propriedades”	
	2		O sistema obtém o documento de propriedades do recurso (utiliza o Serviço de Gestão)
	3		O sistema apresenta o conteúdo do documento
	4	O caso de utilização termina	
Cenário Alternativo 1			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1		No passo 2 do cenário principal, ocorre um erro na obtenção do documento de propriedades do recurso

	2		O sistema apresenta uma mensagem de erro
	3	O caso de utilização termina	

Tabela 16 – Descrição do caso de utilização *Obter documento de propriedades*

Descrição de caso de utilização			
Designação	Obter propriedade		
Resumo	Obter uma propriedade de um recurso		
Pré-condições	Está seleccionada uma instância		
Actores	Gestor de processos		
Cenário Principal			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	O caso de utilização inicia-se com o actor a introduzir o identificador da propriedade (consultar requisito R1)	
	2	O actor pressiona o botão “Obter propriedade”	
	3		O sistema obtém a propriedade do recurso (utiliza o Serviço de Gestão)
	4		O sistema apresenta a informação da propriedade
	5	O caso de utilização termina	
Cenário Alternativo 1			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1		No passo 3 do cenário principal, ocorre um erro na obtenção da propriedade do recurso
	2		O sistema apresenta uma mensagem de erro
	3	O caso de utilização termina	

Tabela 17 – Descrição do caso de utilização *Obter propriedade*

Descrição de caso de utilização			
Designação	Suspende instância		
Resumo	Suspende a execução de uma instância no sistema de execução		
Pré-condições	Está seleccionada uma instância		
Actores	Gestor de processos		
Cenário Principal			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	O caso de utilização inicia-se com o actor a pressionar o botão “Suspende instância”	
	2		O sistema suspende a execução da instância (utiliza o Serviço de Gestão)
	3		O sistema apresenta mensagem de sucesso

	4	O caso de utilização termina	
Cenário Alternativo 1			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1		No passo 2 do cenário principal, ocorre um erro ao suspender a instância
	2		O sistema apresenta uma mensagem de erro e a instância não é suspensa
	3	O caso de utilização termina	
Cenário Alternativo 2			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1		No passo 2 do cenário principal, a instância não altera o seu estado, no sistema de execução, para "suspensa"
	2		O sistema apresenta uma mensagem de impossibilidade de suspender a instância
	3	O caso de utilização termina	

Tabela 18 – Descrição do caso de utilização Suspende instância

Descrição de caso de utilização			
Designação	Prosseguir instância		
Resumo	Prosseguir a execução de uma instância no sistema de execução		
Pré-condições	Está seleccionada uma instância		
Actores	Gestor de processos		
Cenário Principal			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	O caso de utilização inicia-se com o actor a pressionar o botão "Continuar instância"	
	2		O sistema prossegue a execução da instância (utiliza o Serviço de Gestão)
	3		O sistema apresenta mensagem de sucesso
	4	O caso de utilização termina	
Cenário Alternativo 1			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1		No passo 2 do cenário principal, ocorre um erro ao prosseguir a instância
	2		O sistema apresenta uma mensagem de erro e a instância não é prosseguida
	3	O caso de utilização termina	
Cenário Alternativo 2			
Sequência de	#	Input do Actor	Comportamento do sistema

acções	1		No passo 2 do cenário principal, a instância não altera o seu estado, no sistema de execução, para “activa”
	2		O sistema apresenta uma mensagem de impossibilidade de prosseguir a instância
	3	O caso de utilização termina	

Tabela 19 – Descrição do caso de utilização *Prosseguir instância*

Descrição de caso de utilização			
Designação	Terminar instância		
Resumo	Terminar a execução de uma instância no sistema de execução		
Pré-condições	Está seleccionada uma instância		
Actores	Gestor de processos		
Cenário Principal			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	O caso de utilização inicia-se com o actor a pressionar o botão “Terminar instância”	
	2		O sistema termina a execução da instância (utiliza o Serviço de Gestão)
	3		O sistema apresenta mensagem de sucesso
	4	O caso de utilização termina	
Cenário Alternativo 1			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1		No passo 2 do cenário principal, ocorre um erro ao terminar a instância
	2		O sistema apresenta uma mensagem de erro e a instância não é terminada
	3	O caso de utilização termina	
Cenário Alternativo 2			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1		No passo 2 do cenário principal, a instância não altera o seu estado, no sistema de execução, para “terminada”
	2		O sistema apresenta uma mensagem de impossibilidade de terminar a instância
	3	O caso de utilização termina	

Tabela 20 – Descrição do caso de utilização *Terminar instância*

Descrição de caso de utilização			
Designação	Seleccionar processo		
Resumo	Seleccionar um processo da lista de processos		
Pré-condições	Foi obtida a lista de processos		
Actores	Gestor de processos		
Cenário Principal			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	O caso de utilização inicia-se com o actor a seleccionar um processo da lista	
	2		O sistema apresenta a informação disponível acerca do processo seleccionado (consultar requisito R2)
	3	O caso de utilização termina	

Tabela 21 – Descrição do caso de utilização *Seleccionar processo*

Descrição de caso de utilização			
Designação	Seleccionar instância		
Resumo	Seleccionar uma instância da lista de instâncias		
Pré-condições	Foi obtida a lista de instâncias		
Actores	Gestor de processos		
Cenário Principal			
Sequência de acções	#	Input do Actor	Comportamento do sistema
	1	O caso de utilização inicia-se com o actor a seleccionar uma instância da lista	
	2		O sistema apresenta a informação disponível acerca da instância seleccionada (consultar requisito R3)
	3	O caso de utilização termina	

Tabela 22 – Descrição do caso de utilização *Seleccionar instância*

Especificação suplementar

R1 O identificador da propriedade do recurso é do tipo *string*.

R2 A informação do processo contempla o seu identificador, nome, data de inserção, estado e versão.

R3 A informação da instância contempla o seu identificador, o identificador da propriedade do recurso, o nome do processo, o estado, a data em que foi iniciada e a última data em que esteve activa.