Instituto de Pesquisas Tecnológicas do Estado de São Paulo

Antonio Fernando Gaspar Santos

Um método baseado em RM-ODP para verificação de especificações de padrões de mensagem confiável em *Web services*

São Paulo

2007

Antonio Fernando Gaspar Santos

Um método baseado em RM-ODP para verificação de especificações de padrões de mensagem confiável em *Web services*

Dissertação apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT, para obtenção do título de Mestre em Engenharia da Computação.

Área de Concentração: Redes de computadores

Orientador: Dr. Jorge L. Risco Becerra

São Paulo

2007

Ficha Catalográfica
Elaborada pelo Departamento de Acervo e Informação Tecnológica – DAIT
do Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT

S237m Santos, Antonio Fernando Gaspar
Um método baseado em RM-ODP para verificação de especificações de padrões de mensagem confiável em Web services. / Antonio Fernando Gaspar dos Santos. São Paulo, 2007.
172p.

Dissertação (Mestrado em Engenharia de Computação) - Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Área de concentração: Redes de Computadores

Orientador: Prof. Dr. Jorge L. Risco Becerra

1. RM-ODP (Reference Model for Open Distributed Processing) 2. EPMC (Especificação de Padrão de Mensagem Confiável) 3. WEB services 4. Internet (redes de computadores) 5. Entrega confiável de mensagem 5. Tese I. Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Centro de Aperfeiçoamento Tecnológico II.Título

07-140 CDU 004.773(043)

Aos meus pais, Nelson e Inês
e ao eterno “Seu Gaspar”.

Agradecimentos

Ao Professor Doutor Jorge Risco, pela amizade, oportunidade e especialmente pela orientação conduzida, fator fundamental para a elaboração e conclusão do presente trabalho.

À minha esposa Gisele e aos meus familiares, por todo o incentivo e compreensão ao longo desta jornada.

Aos colegas do IPT, aos meus amigos e a todos aqueles que, direta ou indiretamente, colaboraram para a conclusão desta etapa desafiante em minha carreira.

RESUMO

O conceito de SOA (*Service Oriented Architectures*) ou Arquiteturas Orientadas a Serviços tem sido apontado pela indústria de tecnologia da informação como passo evolutivo em arquitetura de software visando, principalmente, a minimizar os problemas da complexidade de integração de sistemas corporativos.

Os *Web services* são vistos pela comunidade de TI como uma prova de conceito da viabilidade de Arquiteturas Orientadas a Serviços, suportados por alguns padrões tecnológicos já estabelecidos. Apesar destes padrões atenderem aos requisitos de muitas aplicações, ainda restam algumas expectativas alusivas ao seu uso em integração de sistemas, pois requerimentos não funcionais, tais como gerenciamento e segurança, ainda carecem de padrões. Além disso, muitos desses padrões dependem de um padrão que garanta a entrega de mensagens em *Web services*.

Este trabalho propõe um método para verificação de especificações de padrões atuais de mensagem confiável em *Web services*. Ele é composto por um modelo de referência para entrega confiável de mensagens, com base no RM-ODP (*Reference Model for Open Distributed Processing*), e por um processo de verificação. Seu objetivo é permitir a verificação de requisitos sobre as duas propostas atuais (*WS-Reliability* e *WS-ReliableMessaging*) que são reconhecidas pelo W3C (*World Wide Web Consortium*) como as duas opções para o futuro padrão a ser adotado pela indústria, para implementar mensagem confiável em *Web services*. Uma vez obtidos os resultados de verificação, respectivos a cada uma dessas especificações de padrão, estabelece-se uma comparação qualitativa entre elas, provendo uma visão imparcial sobre como cada uma delas atende aos requisitos de entrega confiável de mensagens em *Web services*.

Palavras-chaves: entrega confiável de mensagens, protocolos de transporte confiável de dados, RM-ODP, verificação de software, *Web services*, *WS-Reliability*, *WS-ReliableMessaging*.

ABSTRACT

The concept of SOA (Service Oriented Architectures) has been claimed by the information technology industry as the next step in the software architecture area aiming to minimize issues in the complexity of corporate systems integration.

The technology information community claims Web services as proof of concept for the viability of service oriented architectures implementation, provided that Web services are supported by a set of standards already in place. In spite of current Web services standards are able to accomplish many application requirements, there are still some expectations regarding to the use of Web services for systems integration. Some non-functional requirements such as management and security still need standards to support them. Besides, many missing standards in Web services depend on a standard that assures reliable messaging delivery.

This work proposes a method to the verification of the current standard specifications for reliable messaging in Web services. It comprises a RM-ODP (*Reference Model for Open Distributed Processing*) based reliable messaging reference model and a verification process. The main goal of the proposed method is to allow proceeding requirements verification over the two current standard specifications (WS-Reliability and WS-ReliableMessaging) that are recognized by the World Wide Web Consortium – W3C as the options for the standard to be adopted in the future by the industry as the reliable messaging standard for Web services. Once obtained the verification results over each one of these two proposed standard specifications, a qualitative comparison is set. The requirements based comparison provides an independent point of view of how each one of these proposals is able to accomplish the Web services reliable messaging requirements.

Key words: reliable messaging delivery, reliable data transport protocols, RM-ODP, software verification, Web services, WS-Reliability, WS-ReliableMessaging.

Lista de Ilustrações

Figura 1.1 Pilha da arquitetura *Web sevices* 19

Figura 2.1 Os pontos de vista da arquitetura de um sistema, segundo o RM-ODP..... 24

Figura 2.2 Elementos que compõem o modelo de canal..... 27

Figura 2.3 Diagrama de seqüência de pacotes no protocolo *go-back-n*..... 31

Figura 2.4 Diagrama de seqüência de perdas de pacotes no protocolo *go-back-n*..... 32

Figura 2.5 Diagrama de seqüência do protocolo repetição seletiva..... 33

Figura 2.6 Envio de mensagem entre dois *endpoints*, por múltiplos protocolos de transporte..... 37

Figura 3.1 Elementos da estrutura do Método de Verificação..... 51

Figura 3.2 O Processo de Verificação..... 52

Figura 3.3 Níveis de abstração: os pontos de vista do Modelo de Referência..... 55

Figura 3.4 Diagrama UML de pacotes para os requisitos do ponto de vista da empresa..... 56

Figura 3.5 Diagrama de classes do ponto de vista da empresa para entrega confiável de mensagens..... 59

Figura 3.6 Diagrama UML de classes de estruturas agregadas de uma mensagem..... 60

Figura 3.7 Diagrama UML de classes dos elementos agregados e generalizados de uma mensagem..... 63

Figura 3.8 Diagrama UML de atividades do esquema dinâmico..... 64

Figura 3.9 Diagrama de contexto do ponto de vista da computação para entrega confiável de mensagens..... 66

Figura 3.10 Diagrama UML de seqüência para o modo básico de interação..... 67

Figura 3.11 Diagrama UML de seqüência para o modo de interação *go-back-n*..... 68

Figura 3.12 Diagrama UML de seqüência para o modo de interação por repetição seletiva..... 69

Figura 3.13 Diagrama UML de seqüência para conexão explícita..... 71

Figura 3.14 Diagrama UML de seqüência para desconexão explícita..... 72

Figura 3.15 Diagrama UML de MEF para conexão explícita..... 75

Figura 3.16 Diagrama UML de MEF para o modo básico de interação..... 77

Lista de Tabelas

Tabela 3.1 Estrutura de Matriz de Mapeamento de Requisitos..... 79

Tabela 4.1 Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao escopo, comunidade, federação, funções empresariais e procedimentos..... 81

Tabela 4.2 Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao contrato..... 83

Tabela 4.3 Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto às políticas..... 85

Tabela 4.4 Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao esquema invariante..... 86

Tabela 4.5 Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao esquema estático..... 89

Tabela 4.6 Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao esquema dinâmico..... 90

Tabela 4.7 Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao tipo de interface..... 91

Tabela 4.8 Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao modo de interação..... 92

Tabela 4.9 Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto aos modos de conexão e desconexão..... 93

Tabela 4.10 Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto aos objetos do canal..... 94

Tabela 4.11 Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao escopo, comunidade, federação, funções empresariais e procedimentos..... 96

Tabela 4.12 Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao contrato..... 98

Tabela 4.13 Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto às políticas..... 100

Tabela 4.14 Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao esquema invariante..... 101

Tabela 4.15 Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao esquema estático..... 103

Tabela 4.16 Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao esquema dinâmico..... 104

Tabela 4.17 Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao tipo de interface..... 106

Tabela 4.18 Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao modo de interação..... 107

Tabela 4.19 Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto aos modos de conexão e desconexão..... 108

Tabela 4.20 Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto aos objetos do canal..... 110

Tabela 4.21 Comparação entre as EPMCs quanto ao escopo, comunidade, federação, funções empresariais e procedimentos..... 114

Tabela 4.22 Comparação entre as EPMCs quanto ao contrato..... 115

Tabela 4.23 Comparação entre as EPMCs quanto às políticas..... 118

Tabela 4.24 Comparação entre as EPMCs quanto ao esquema invariante..... 121

Tabela 4.25 Comparação entre as EPMCs quanto ao esquema estático..... 128

Tabela 4.26 Comparação entre as EPMCs quanto ao esquema dinâmico..... 129

Tabela 4.27 Comparação entre as EPMCs quanto ao tipo de interface..... 134

Tabela 4.28 Comparação entre as EPMCs quanto ao modo de interação..... 135

Tabela 4.29 Comparação entre as EPMCs quanto aos modos de conexão e desconexão..... 137

Tabela 4.30 Comparação entre as EPMCs quanto à estrutura de canal..... 140

Tabela 4.31 Síntese dos resultados de verificação e comparativo entre as EPMCs..... 146

Lista de Abreviaturas e Siglas

ACK	Positive Acknowledgement
ACM	Association for Computing Machinery
ADR	Active Design Review
APDU	Application Protocol Data Unit
ARQ	Automatic Response Request
ATAM	Architecture Trade-off Analysis Method
AWG	Architecture Working Group (do departamento de defesa dos Estados Unidos da América)
C4ISR	Computer, Communications, Command and Control for Information Surveilance and Reconnaissance
CIO	Chief Information Officers
COA	Confirmation Of Arrival
CORBA	Common Object Request Broker Architecture
DoD	United States Department of Defense
EAI	Enterprise Application Integration
ECM	Entrega Confiável de Mensagens
EDA	Event Driven Architecture
EDC	Error Detecting Code
EDOC	Enterprise Distributed Object Computing
EPMC	Especificação de Padrão de Mensagem Confiável
FEAF	Federal Enterprise Architecture *Framework*
FTP	File Transfer Protocol
FTR	Formal Technical Review
HTTP	HyperText Transfer Protocol
HTTPS	(vide SHTTP)
IDL	Interface Definition Language
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ISI	Informantion Science Institute
J2EE	Java 2 Enterprise Edition
JMS	Java Messaging Service

MCA	Message Channel Agent
MDA	Model Driven Architecture
MEF	Máquina de Estados Finitos
MEP	Message Exchange Pattern
MIC	Message Integrity Code
MIT	Massachusetts Institute of Technology
MOM	Message Oriented Middleware
MOWS	Management Of Web Services
MUWS	Management Using Web Services
NACK	Negative Acknowledgement
NETBLT	Network Block Transfer
NIST	National Institute of Standards and Technology
OASIS	Organization for the Advancement of Structured Information Standards
ODP	Open Distributed Processing
OSE	Open Systems Environment
OMG	Object Management Group
PACK	(vide ACK)
PDU	Protocol Data Unit
PIM	Plartform Independent Model
PSM	Platform Specific Model
RDT	Reliable Data Transfer
RM	Reliable Messaging
RM-ODP	Reference Model for Open Distributed Processing
RMP	Reliable Messaging Processor
RPC	Remote Procedure Call
RTO	Retransmition Time Out
RTT	Round Trip Time
RTTM	Round Trip Time measurement
SAAM	Scenario-based Architecture Analysis Method
SHTTP	Secure Hypertext Transfer Protocol
SMTP	Single Mail Transfer Protocol
SOA	Service Oriented Architecture

SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
T/TCP	TCP Extensions for Transactions
TCP	Transport Control Protocol
TI	Tecnologia da Informação
TINA	Telecommunication Information Network Architecture
TPDU	Transport Protocol Data Unit
UDDI	Universal Description Discovery and Integration
UDT	Unreliable Data Transfer
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Universal Time Coordinated
VMTP	Versatile Message Transaction Protocol
XML	Extensible Markup Language
W3C	World Wide Web Consortium
WS-I	Web Services Interoperability Organization
WS-RM	Web Services Reliability Messaging
WS-TX	Web Services Transaction
WSA	Web Services Architecture
WSDL	Web Services Description Language
WSDM	Web Services Distributed Management
WSN	Web Services Notification
WSRM	Web Services Reliable Messaging
WSS	Web Services Security

Sumário

1 INTRODUÇÃO..... 16

1.1 Objetivos..... 17

1.2 Abrangência..... 18

1.3 Justificativa..... 19

1.4 Metodologia..... 20

1.5 Estrutura do trabalho..... 22

2 RM-ODP, TRANSPORTE CONFIÁVEL, *WEB SERVICES* E VERIFICAÇÃO.... 23

2.1 O Modelo de Referência ODP..... 23

2.1.1 Os pontos de vista do RM-ODP..... 24

2.1.2 O Modelo de Referência ODP e interoperabilidade..... 28

2.1.3 Arquiteturas de referência e os pontos de vista do RM-ODP..... 28

2.2 Conceitos de transporte confiável de dados..... 29

2.3 Os *Web services.*..... 33

2.3.1 Órgãos de padronização *Web services.* 34

2.3.2 Confiabilidade em *Web services*..... 35

2.3.3 Independência de protocolo de transporte..... 36

2.3.4 Mensagem confiável em *Web Services*..... 38

2.3.5 Padrões dependentes de mensagem confiável..... 38

2.4 As opções de padrão de mensagem confiável em *Web services*..... 41

2.4.1 A EPMC *WS-Reliability*..... 41

2.4.2 A EPMC *WS-ReliableMessaging* 43

2.5 Verificação e avaliação de especificações de arquiteturas..... 45

2.5.1 Verificação de software..... 45

2.5.2 Características de processo de verificação..... 47

3 O MÉTODO DE VERIFICAÇÃO..... 50

3.1 A estrutura do método..... 50

3.2 O Processo de Verificação..... 52

3.3 O Modelo de Referência..... 53

3.3.1 O ponto de vista da empresa..... 55

3.3.2 O ponto de vista da informação..... 59

3.3.2.1 O esquema invariante..... 59

3.3.2.2 O esquema estático..... 62
3.3.2.3 O esquema dinâmico..... 63
3.3.3 O ponto de vista da computação 65
3.3.3.1 Tipo de interface..... 66
3.3.3.2 Modos de interação..... 67
3.3.3.3 Modos de conexão e desconexão..... 70
3.3.4 O ponto de vista da engenharia..... 72
3.3.4.1 Objetos adaptadores..... 73
3.3.4.2 Objetos conectores..... 73
3.3.4.3 Objetos protocolos..... 76
3.4 Matriz de Mapeamento de Requisitos..... 79
4 VERIFICAÇÃO E COMPARAÇÃO ENTRE AS EPMCs..... 81
4.1 Mapeamento de Requisitos da EPMC *WS-Reliability*..... 81
4.1.1 Matrizes para o ponto de vista da empresa..... 81
4.1.2 Matrizes para o ponto de vista da informação..... 86
4.1.2.1 Quanto ao esquema invariante..... 86
4.1.2.2 Quanto ao esquema estático..... 89
4.1.2.3 Quanto ao esquema dinâmico..... 89
4.1.3 Matrizes para o ponto de vista da computação..... 91
4.1.3.1 Quanto ao tipo de interface..... 91
4.1.3.2 Quanto ao modo de interação..... 91
4.1.3.3 Quanto aos modos de conexão e desconexão..... 93
4.1.4 Matriz para o ponto de vista da engenharia..... 94
4.2 Mapeamento de Requisitos da EPMC *WS-ReliableMessaging*..... 96
4.2.1 Matrizes para o ponto de vista da empresa..... 96
4.2.2 Matrizes para o ponto de vista da informação..... 101
4.2.2.1 Quanto ao esquema invariante..... 101
4.2.2.2 Quanto ao esquema estático..... 103
4.2.2.3 Quanto ao esquema dinâmico..... 103
4.2.3 Matrizes para o ponto de vista da computação..... 106
4.2.3.1 Quanto ao tipo de interface..... 106
4.2.3.2 Quanto ao modo de interação..... 106
4.2.3.3 Quanto aos modos de conexão e desconexão..... 108

4.2.4 Matriz para o ponto de vista da engenharia..... 110
4.3 Comparação entre as EPMCs *WS-Realibility* e *WS-ReliableMessaging*..... 112
4.3.1 Comparação quanto ao ponto de vista da empresa..... 113
4.3.2 Comparação quanto ao ponto de vista da informação..... 121
4.3.2.1 Comparação quanto ao esquema invariante..... 121
4.3.2.2 Comparação quanto ao esquema estático..... 127
4.3.2.3 Comparação quanto ao esquema dinâmico..... 129
4.3.3 Comparação quanto ao ponto de vista da computação..... 133
4.3.3.1 Comparação quanto ao tipo de interface..... 133
4.3.3.2 Comparação quanto ao modo de interação..... 135
4.3.3.4 Comparação quanto aos modos de conexão e desconexão..... 137
4.3.4 Comparação quanto ao ponto de vista da engenharia..... 140
4.4 Conclusões gerais sobre os resultados..... 141
5 CONSIDERAÇÕES FINAIS..... 151
5.1 Conclusões sobre o Método de Verificação..... 151
5.2 Continuidade da pesquisa..... 152
Referências..... 154
Glossário..... 165
Anexos..... 169

1 INTRODUÇÃO

O conceito de SOA (*Service Oriented Architecture*), ou Arquiteturas Orientadas a Serviços, tem sido apontado pela indústria de TI como o próximo passo evolutivo em arquitetura de *software* visando a minimizar os problemas de complexidade, especialmente relacionados com integração de empresas (HOLLEY et al. 2003).

Atualmente, há um consenso de visão dos *Web services* como uma prova de conceito de implementação de SOA (ENDREI et al., 2004; HOLLEY et al. 2003; KAYE, 2004; SLEEPER, 2004).

Uma vez suportados por padrões difundidos, como o SOAP (*Simple Object Access Protocol*), o UDDI (*Universal Description Discovery and Integration*), o WSDL (*Web Services Description Language*) e o XML (*Extensible Markup Language*), os *Web services* estão ganhando grande popularidade como tecnologia de integração em organizações. Trata-se de uma nova abordagem tecnológica em integração de aplicações em ambientes heterogêneos (MUKHI et al., 2004a).

Não obstante que os padrões tecnológicos citados acima sejam suficientes para muitas aplicações de *Web services*, ainda há outras expectativas alusivas ao seu uso em integração de sistemas de missão crítica, aqueles cuja falha pode causar impactos em segurança ou grandes perdas financeiras. Os *Web services* ainda carecem de padrões visando a atender requisitos não-funcionais, tais como segurança, confiabilidade e gerenciamento (BIRMAN; RENESSE; VOGELS, 2004a; BIRMAN, 2004b; ERRADI; MAHESHWARI, 2005; KAYE, 2004; MUKHI et al., 2004a; MUKHI; KONURU; CORBERA, 2004b; PALLICKARA; FOX, 2005; SLEEPER, 2004).

O conceito de confiabilidade em *Web services* refere-se à mensagem confiável. Esta, por sua vez, corresponde à garantia de que uma mensagem será entregue e que tanto seu remetente quanto seu destinatário terão a mesma compreensão do *status* da entrega. Entrega garantida, ordenada, íntegra e não duplicada de mensagem são os principais atributos esperados em mensagem confiável (ERRADI; MAHESHWARI, 2005; TAI; MIKALSEN; ROUVELLOU, 2003; W3C, 2004a, 2004b, 2004c, 2004e).

Um padrão de mensagem confiável amplia a eficiência dos demais padrões dos *Web services*, tais como segurança, transações e processos de negócios. Essas melhorias são possíveis somente se a entrega confiável de mensagens for um padrão e não estiver implementada na lógica de negócios. Para tanto, duas especificações de padrão de mensagem confiável em *Web services* foram propostas aos comitês técnicos de entidades de padronização *Web services*, para possível adoção pela indústria de software. Qual das duas será adotada e qual delas tem melhor aderência às necessidades de mensagem confiável ainda é objeto de fóruns de discussão em vários âmbitos.

1.1 Objetivos

O objetivo principal deste trabalho é propor um Método de Verificação visando a avaliar qualitativamente os requisitos contemplados em cada uma das propostas de padrão de mensagem confiável, submetidas às entidades de padronização. Cada proposta (doravante denominadas neste pela sigla EPMC - Especificação de Padrão de Mensagem Confiável) é citada pelo W3C (*World Wide Web Consortium*). Segundo o W3C, as duas opções atuais que visam a implementar mensagem confiável em *Web services*, a partir da extensibilidade do protocolo de aplicação padrão SOAP, são: *WS-Reliability* e *WS-ReliableMessaging* (OASIS, 2004b, 2006c; W3C, 2004a).

Na proposição do Método de Verificação são definidos elementos que fazem parte de sua estrutura. Um Modelo de Referência para entrega confiável de mensagens é o elemento que provê a especificação de uma arquitetura de referência baseada em RM-ODP (*Reference Model for Open Distributed Processing*), definindo requisitos almejados em entrega confiável de mensagens, quanto às partes envolvidas e seus relacionamentos, restrições, objeto informação, interfaces e interações.

Para realizar a verificação, tem-se o elemento Processo de Verificação com uma abordagem de avaliação de especificação de arquitetura. O Processo de Verificação visa a estabelecer um mapeamento, inerente à verificação, entre os requisitos almejados e os requisitos contemplados na EPMC verificada (IEEE, 2004; RAKITIN, 2001).

Para o registro dos resultados de verificação, tem-se outro elemento do Método de Verificação, a Matriz de Mapeamento de Requisitos. Trata-se de uma estrutura tabular que registra o mapeamento entre os requisitos objetivados - esses definidos pelo Modelo de Referência - e os requisitos contemplados na EPMC. Além disso, resultados de verificações individuais permitem a comparação qualitativa (DOBRICA; NIEMELA, 2002).

Posto isto, a partir das Matrizes de Mapeamento de Requisitos - respectivas à verificação de cada uma das EPMCs - viabiliza-se uma comparação qualitativa entre elas. Por resultado, obtém-se uma visão independente de como cada uma das duas EPMCs em *Web services* atende aos requisitos de entrega confiável de mensagens, bem como suas diferenças. Esses resultados podem contribuir para os contextos de fóruns de discussão pela escolha entre uma ou outra proposta de padrão, bem como prover subsídios de verificação independente dessas à comunidade de profissionais envolvidos em iniciativas de avaliação e utilização de tecnologias baseadas em *Web services*.

1.2 Abrangência

O contexto macro deste trabalho situa-se na arquitetura dos *Web services*. Ela foi definida pelo grupo de trabalho WSA (*Web Services Architecture*) do W3C, cuja missão é desenvolver e manter uma arquitetura de referência padrão para *Web services* (W3C, 2004b).

Dentro desse contexto macro, seu foco é específico na camada de mensagens da pilha de arquitetura *Web services*. Ou seja, abaixo da camada de descrição de serviços e acima da camada de comunicação. A figura 1.1 situa a camada de mensagens na pilha de arquitetura *Web services* do W3C.

Ainda na camada de mensagens, o presente trabalho restringe-se ao aspecto denominado pelo W3C como mensagem confiável (*reliable messaging*) e propõe um Método de Verificação para as atuais propostas de padrão de mensagem confiável em *Web services*.

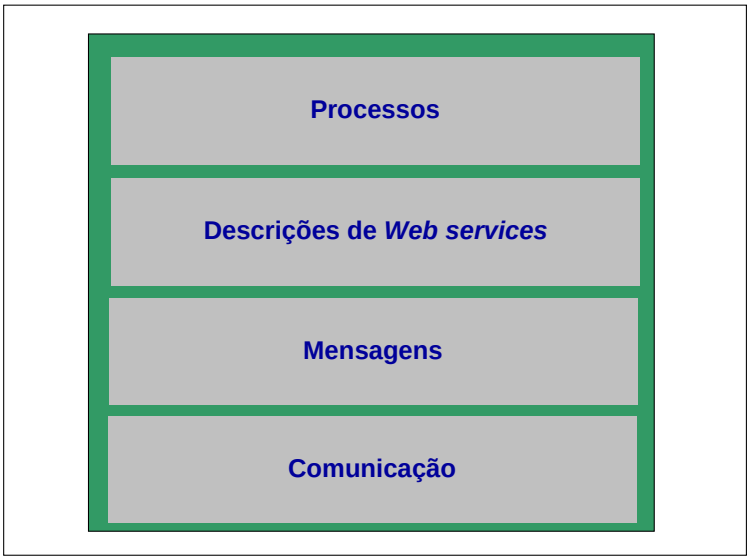


Figura 1.1. Pilha da arquitetura *Web sevices* (W3C, 2004a).

Demais aspectos não-funcionais em *Web services* (ex: segurança e gerenciamento) não são aqui abordados, uma vez que esses constituem, nos órgãos padronizadores de *Web services*, frentes distintas de estudos. Essas frentes de estudos são paralelas ao escopo de mensagem confiável e têm por objetivo propor os padrões para demais aspectos não-funcionais, ainda pendentes em *Web services*.

1.3 Justificativa

Com o crescimento de implementações baseadas em *Web services*, o âmbito de entrega confiável de mensagens desperta interesse da comunidade científica, além de órgãos de padronização focados em sistemas distribuídos.

Uma vez que o padrão de arquitetura dos *Web services* ainda carece de complementações em aspectos não-funcionais (especialmente em mensagem confiável), existem atualmente duas especificações que propõem implementar mensagem confiável em *Web services* e que se encontram sob análise dos comitês técnicos de entidades de padronização *Web services*. Apesar da disponibilidade de informações na Internet a respeito de cada uma dessas duas propostas (*WS-Reliabiltiy* e *WS-ReliableMessaging*), eventuais análises individuais e/ou comparativas disponíveis tendem à parcialidade, uma vez que elas foram propostas por consórcios formados por empresas fabricantes de software.

Por meio de verificação independente, objetivo do presente trabalho, é possível uma avaliação qualitativa e imparcial de como cada uma dessas especificações atende aos requisitos de entrega confiável de mensagens em *Web services* (BALCI, 1998; IEEE, 2004; NIST, 1996; RAKITIN, 2001).

Como trabalhos correlatos a este, destacam-se:

- Em (PARNAS; WEISS, 1985), artigo também referenciado em (ALBIN, 2003; BASS; CLEMENS; KAZMAN, 2003; DOBRICA; NIEMELA, 2002) propõe-se que a avaliação de especificação de arquiteturas utilizando-se inspeção e questionamento constitui revisão ativa de projeto e permite realizar verificação. Em (BARCELOS, 2006) disserta-se que inspeção e questionamento são a melhor forma de realizar avaliação de especificação de arquiteturas;

- Em (SINDEREN, 1995) tem-se que a melhor forma para a especificação de um protocolo de aplicação ocorre por meio de arquitetura de referência. Outrossim, em (ALMEIDA; SINDEREN; PIRES, 2004; PUTMAN, 2001) aborda-se a utilização do RM-ODP na especificação de arquiteturas de referência;

- Em (PALLICKARA; FOX, 2005) tem-se uma primeira tentativa de análise das duas EPMCs verificadas neste trabalho. Durante a fase de pesquisa e elaboração do presente trabalho, esta foi a única referência encontrada, de autoria não diretamente ligada à indústria, que buscava estabelecer uma comparação generalizada dessas EPMCs. Entretanto, trata-se de análise sobre versões muito preliminares das respectivas EPMCs *WS-Reliability* e *WS-ReliableMessaging*, portanto, desatualizada. Outro aspecto do artigo é a ausência de explanação quanto ao método utilizado para a análise comparativa bem como finalização pouco conclusiva.

1.4 Metodologia

O desenvolvimento desta dissertação adota a seguinte metodologia:

Pesquisa - fase que enfoca a busca, seleção e classificação de informações para base de conhecimento e fundamentação deste trabalho. As fontes de pesquisa

comportam livros, artigos publicados em entidades científicas, trabalhos de pós-graduação e publicações de entidades de padronização, focando principalmente nos temas relativos aos *Web services*, requisitos de mensagem confiável em *Web services*, transporte confiável de dados, sistemas de processamento distribuído e verificação de software.

Proposição do Método de Verificação - etapa onde se definem os elementos que compõem o método aqui proposto. Nele define-se um Modelo de Referência, baseado em níveis discretos de abstrações. Para estruturar o Modelo de Referência, adota-se o RM-ODP (*Reference Model for Open Distributed Processing*) da ISO cujo, objetivo principal é prover um modo consistente e sistemático para a especificação de uma arquitetura de referência contemplando os requisitos de entrega confiável de mensagens.

Para operacionalizar a verificação um Processo de Verificação é proposto, com abordagem de avaliação de especificação de arquitetura, por inspeção e questionamento (BARCELOS, 2006; DOBRICA; NIEMELA, 2002; DoD, 2001; PARNAS; WEISS, 1985).

O Processo de Verificação visa a estabelecer o mapeamento entre os requisitos almejados (definidos no Modelo de Referência) e os requisitos contemplados na EPMC verificada. O Método de Verificação aplica o Processo de Verificação sobre as EPMCs *WS-Reliability* e *WS-ReliableMessaging*, que são as duas propostas correntes de extensão do padrão SOAP de mensagens em *Web services* e que visam a implementar entrega confiável de mensagens. Para registro dos resultados da verificação, o último elemento do Método de Verificação é a Matriz de Mapeamento de Requisitos. Trata-se de uma estrutura tabular que registra o mapeamento entre os requisitos objetivados e os requisitos contemplados na especificação verificada.

Verificação das EPMCs e análise comparativa - engloba a operacionalização do Método de Verificação sobre cada EPMC. A partir dos resultados de verificação de cada uma dessas, viabiliza-se um comparativo qualitativo entre elas, tomando-se os

requisitos dos pontos de vista estabelecidos no Modelo de Referência como parâmetros de comparação.

1.5 Estrutura do Trabalho

Esta dissertação é composta de cinco capítulos, a seguir descritos:

O Capítulo 1, Introdução - apresenta a introdução, os objetivos, a abrangência, a justificativa, a metodologia e a estrutura do trabalho. Este capítulo provê um preâmbulo geral e sucinto de todo o contexto desenvolvido nos capítulos subseqüentes.

O Capítulo 2, RM-ODP, transporte confiável, *Web Services* e verificação - introduz os conceitos tecnológicos básicos que sustentam e/ou permeiam a área e tema focos deste trabalho.

O Capítulo 3, O Método de Verificação - apresenta os elementos que estruturam o método a ser aplicado para verificação das EPMCs com relação ao contexto de mensagem confiável em *Web services*. Neste capítulo são definidos o Modelo de Referência, o Processo de Verificação para entrega confiável de mensagens e a Matriz de Mapeamento de Requisitos.

O Capítulo 4, Verificação e comparação entre as EPMCs - Neste, apresentam-se as Matrizes de Mapeamento de Requisitos, resultantes da verificação das EPMCs *WS-Reliability* e *WS-ReliableMessaging*. A partir dessas matrizes, apresenta-se um comparativo qualitativo entre as EPMCs, tomando-se os requisitos dos pontos de vista estabelecidos no Modelo de Referência como parâmetros de comparação.

O Capítulo 5, Considerações finais - apresenta as conclusões deste trabalho, contribuições, comentários gerais sobre os resultados e pontos passíveis de continuidade de pesquisa em futuros trabalhos acadêmicos.

2 RM-ODP, TRANSPORTE CONFIÁVEL, *WEB SERVICES* E VERIFICAÇÃO

Este capítulo apresenta conceitos tecnológicos que sustentam e/ou permeiam os objetivos deste trabalho.

2.1 O Modelo de Referência ODP

O modelo de referência para processamento distribuído aberto RM-ODP é um padrão para especificação de arquiteturas de sistemas de processamento distribuído que separa os aspectos e simplifica a especificação dessas. Ele propõe um metapadrão que descreve uma estrutura para a especificação de arquiteturas de sistemas distribuídos abertos (ISO, 1996a, 1996b, 1998a, 1998b; PUTMAN, 2001; WEGMANN; NAUMENKO, 2001).

O RM-ODP usa um processo de modelagem baseada em múltiplos níveis de abstração. Esses níveis de abstração, representados na figura 2.1, são compostos por cinco pontos de vista, sendo que cada um é associado com uma linguagem própria para abordagem de seu contexto.

No RM-ODP, os conceitos de interação entre objetos, funções, tipos e ações podem ser modelados com uso de UML (*Unified Modeling Language*). A UML é uma linguagem de especificação, pois endereça as decisões importantes de análise e projeto e não está limitada à modelagem de software (OMG, 2005; PUTMAN, 2001; RUMBAUGH; BOOCH; JACKOBSON, 2001).

O RM-ODP tem sido utilizado por muitas instituições privadas e governamentais. Na área de telecomunicações, companhias como *Lucent Technologies*, *AT&T* e *Nortel Networks* utilizaram o RM-ODP para especificação de seus sistemas. O consórcio TINA (*Telecommunications Information Network Architecture*) iniciou seus trabalhos baseando-se no RM-ODP. Na área financeira, destacam-se empresas como *Merrill Lynch*, *Morgan Stanley* e *United Bank of Switzerland* (PUTMAN, 2001).

O próprio OMG (*Object Management Group*) propõe a utilização do RM-ODP como estrutura de especificação de arquiteturas de sistemas distribuídos, baseado em pontos de vista, para especificação de metamodelos MDA (*Model Driven Architecture*). É o caso do EDOC (*Enterprise Distributed Object Computing*), publicado pelo OMG, que provê um metamodelo baseado em MDA e decomposto em pontos de vista RM-ODP.

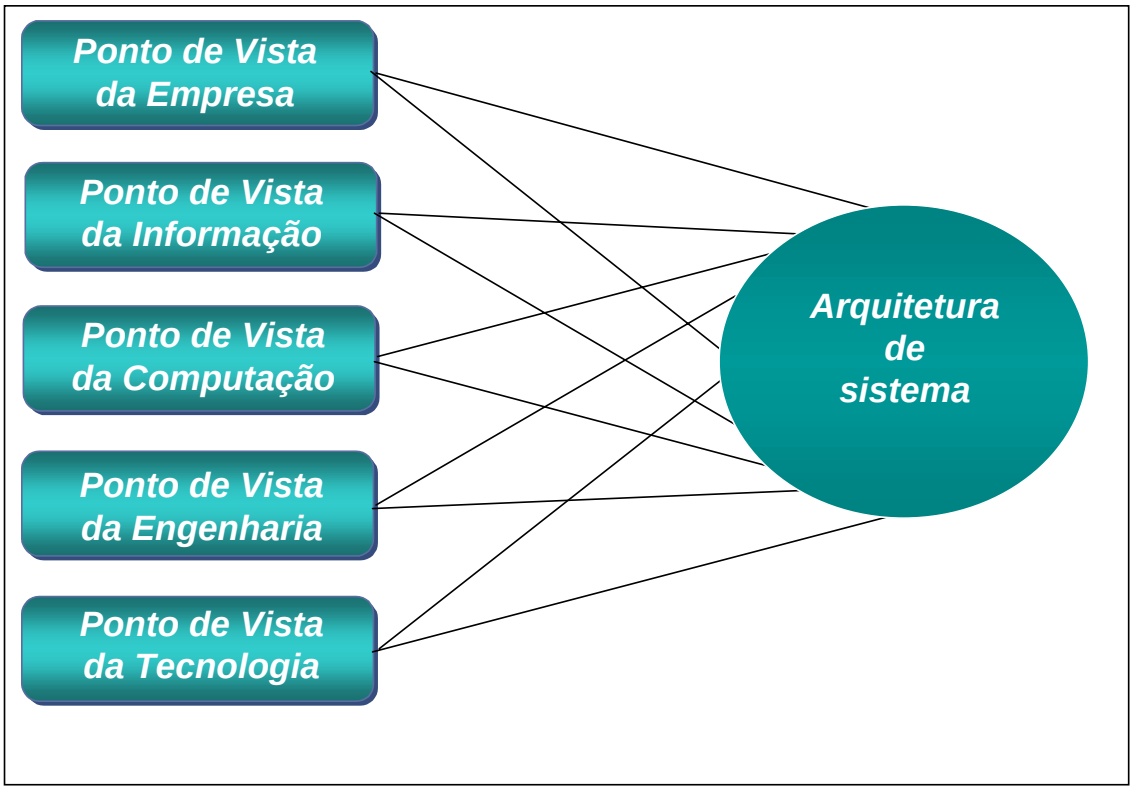


Figura 2.1. Os pontos de vista da arquitetura de um sistema, segundo o RM-ODP.

2.1.1 Os pontos de vista do RM-ODP

Os pontos de vista do modelo de referência ODP provêem um modo de abordagem frente à complexidade envolvida em especificar sistemas distribuídos. Um ponto de vista de um sistema é uma abstração, que produz uma especificação de um sistema completo relativo a um conjunto particular de aspectos. Cinco pontos de vista foram escolhidos para serem simples e completos, convergindo todos os domínios de projetos arquiteturais. Esses cinco pontos de vista são a seguir descritos (ISO, 1996a, 1996b, 1998a; PUTMAN, 2001).

1) Ponto de vista da empresa: provê os requisitos básicos, na visão empresarial. A estrutura do ponto de vista da empresa contém os seguintes aspectos:

   - Comunidades: grupos de objetos com propósito comum;
   - Contrato: refere-se a um acordo entre as partes envolvidas;
   - Escopo: define os objetivos da empresa, visualizando essa como uma entidade de negócios;
   - Federação: grupo de objetos pertencentes a diferentes domínios, com propósito comum;
   - Funções empresariais: são as funções realizadas pelos elementos ou objetos dentro da empresa;
   - Políticas: expressam o que é permitido, o que é proibido e o que é obrigatório;
   - Procedimentos: conjunto de ações específicas realizadas pelos objetos empresa.

2) Ponto de vista da informação: define as semânticas da informação e semânticas do processamento de informação no sistema. Isto é feito por meio de três esquemas:

   - Esquema invariante: especifica as condições que serão sempre verdadeiras, aplicadas a um ou mais objetos de informação;
   - Esquema estático: define o estado e a estrutura de um objeto informação em determinado ponto no tempo e está sujeito às restrições do esquema invariante;
   - Esquema dinâmico: define todas as ações que implicam em mudanças de estados do objeto informação.

3) Ponto de vista da computação: descreve como as aplicações e os componentes distribuídos do sistema ODP interagem. Neste ponto de vista existe a preocupação com interoperabilidade, sem se preocupar com a infra-estrutura de comunicação. Para tal, representa-se a decomposição funcional de um sistema em objetos que interagem por meio de interfaces. Este ponto de vista foca na captura dos objetos e suas ligações, em termos de interações, interfaces e conexões.

O RM_ODP define três tipos de interface possíveis no ponto de vista da computação:

- Interface de operação: suporta interações chamadas “operações”, que são semelhantes aos procedimentos ou rotinas utilizadas em programação de sistemas, caracterizando uma relação cliente-servidor;
- Interface de fluxo: suporta as interações chamadas de “fluxo”, que representam fluxos contínuos de informação, tais como áudio e vídeo em multimídia;
- Interface de sinal: para interações chamadas de “sinais”, que representam eventos e/ou interrupções.

Para conexões, o RM-ODP define dois tipos distintos:

- Conexão explícita: utilizada por interfaces do tipo operação;
- Conexão implícita: utilizada por interfaces do tipo operação, sinal e fluxo, especialmente quando não é necessário explicitar o estabelecimento de persistência entre as partes.

4) Ponto de vista da engenharia: nele são especificadas funções e estruturas que viabilizam a comunicação entre objetos de computação distribuídos num sistema ODP. O ponto de vista da engenharia é composto por três elementos: o objeto básico de engenharia, a estrutura gerencial e a estrutura de comunicação.

A estrutura de gerenciamento tem por objetivo organizar a infra-estrutura do sistema para otimizar a forma de gerenciamento dos objetos distribuídos. Nela surgem os conceitos de grupos de objetos (*clusters*), cápsulas, núcleos e nós.

Existe uma correspondência direta entre o ponto de vista da computação e o ponto de vista da engenharia. No ponto de vista da engenharia, o objeto computação, definido no modelo de computação, transforma-se em objeto básico de engenharia. Por outro lado, as conexões computacionais nele são vistas como canais (ISO, 1996a).

Na estrutura de comunicação, quando os objetos pertencem a diferentes cápsulas ou nós, utiliza-se uma conexão distribuída, padronizada no ODP. Para tal, tem-se uma estrutura denominada de canal, representada na figura 2.2, cujo objetivo é oferecer de forma transparente os serviços de comunicação aos objetos que estão interagindo. O canal é constituído dos seguintes objetos de engenharia:

- Adaptador: objeto que interage diretamente com o objeto básico de engenharia, oferecendo os serviços de conversão dos dados que são gerados na interação;
- Conector: objeto encarregado de estabelecer a ligação (*bind*) e gerenciar a integridade ponto-a-ponto do canal;
- Protocolo: objeto encarregado de oferecer as funções de comunicação;
- Interceptor: objeto aplicado quando é necessário realizar a conversão de protocolos.

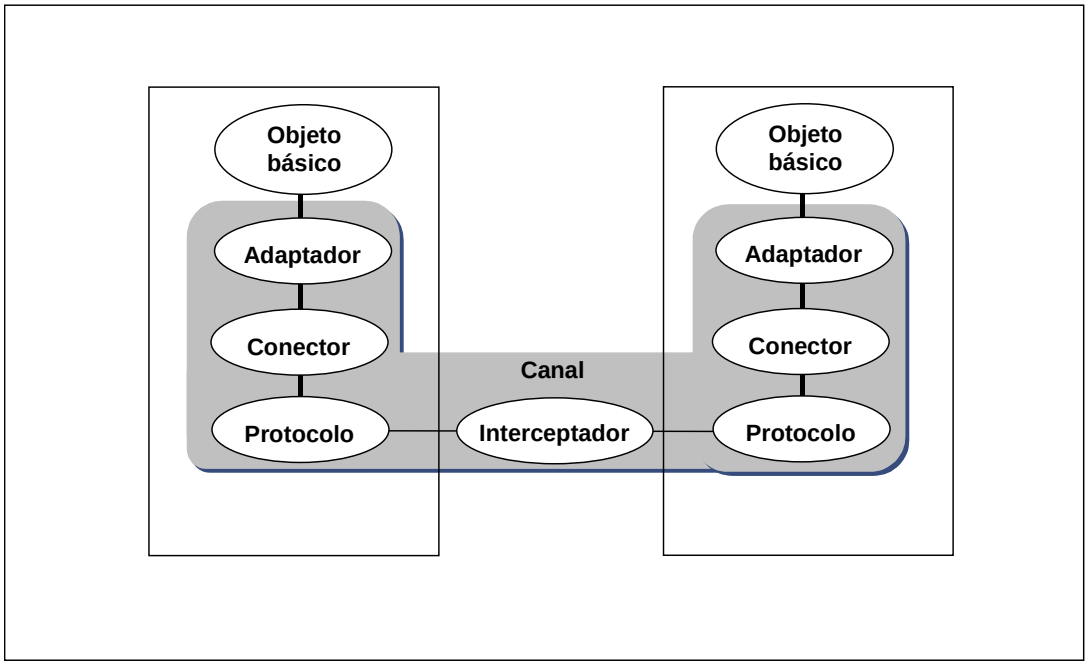


Figura 2.2.Elementos que compõem o modelo de canal.

5) Ponto de vista da tecnologia: trata de aspectos de implementação e objetiva à especificação de configurações de hardware, software e elementos de rede.

2.1.2 O Modelo de Referência ODP e interoperabilidade

Um dos maiores benefícios do RM-ODP é sua habilidade de criar uma arquitetura para somente uma parte de um sistema maior. Ou seja, ele é igualmente aplicado para definir um sistema de sistemas, um sistema ou um componente de software (ISO, 1996a; PUTMAN, 2001).

Por outro lado, o conceito de interoperabilidade está na habilidade de dois ou mais sistemas ou componentes trocarem informação de uma forma mutuamente acordada entre as partes (IEEE, 1990; PUTMAN, 2001; TANENBAUM; STEEN, 2002).

O RM-ODP, sendo um modelo de referência baseado em níveis de abstração, chamados de pontos de vista, tem por objetivo principal o suporte à especificação de arquiteturas de sistemas distribuídos e abertos onde exista interoperabilidade entre diversos sistemas, quer sejam sistemas ODP, sistemas proprietários ou outros sistemas padronizados de mercado (BECERRA, 1998).

A abordagem a interoperabilidade é um dos pontos fortes do RM-ODP, que assume haver níveis de abstração a serem endereçadas para obtê-la. Segundo ele, interoperabilidade é obtida a partir de três aspectos principais: interação que define um relacionamento entre as partes; interface que habilita a interação; conexão que habilita a conexão entre as partes.

Esse contexto, no RM-ODP é chamado de *interaction framework*, que é um padrão de abordagem para descrever como objetos interagem para prover um relacionamento e não como eles são implementados. Esses aspectos são descritos através dos seus cinco pontos de vista, com um enfoque no ponto de vista da computação.

2.1.3 Arquiteturas de referência e os pontos de vista do RM-ODP

Uma especificação de arquitetura pode ser composta de um ou mais pontos de vista. Entretanto, nem todos os pontos de vista são necessários para uma dada especificação, ou seja, o arquiteto não precisa utilizar todos os pontos de vista ou nem mesmo todos os aspectos de um ponto de vista. A escolha é do arquiteto em

decidir o que será utilizado para prover a especificação (ISO, 1996b; PUTMAN, 2001).

Uma arquitetura corresponde aos conceitos e regras que definem a estrutura e relacionamentos entre partes de um sistema. Uma arquitetura de referência pode ser considerada uma especificação de alto nível de um sistema que define sua estrutura de objetivos gerais (componentes e relacionamentos entre elas) de uma forma sistemática e consistente. A diferença essencial entre uma arquitetura de referência e uma arquitetura é que a primeira é uma instância mais detalhadamente especificada que a segunda. Uma arquitetura de referência define a arquitetura visionada, sem endereçar considerações de implementação (ISO, 1996a; PUTMAN, 2001).

O RM-ODP, pode ser aplicado na definição de arquiteturas de referências sendo, portanto, aplicado à definição de protocolos de aplicação. Isto se deve ao fato de que arquiteturas de referências constituem a melhor forma para a especificação de protocolos de aplicação (SINDEREN, 1995).

2.2 Conceitos de transporte confiável de dados

Em geral, o problema de implementar transporte confiável de dados ocorre não somente em camada de transporte, mas também na camada de enlace e na camada de aplicação. Em um cenário realista de transporte de dados, essencialmente três capacitações adicionais de protocolo são exigidas para tratar presença de erros de transmissão (KUROSE; ROSS, 2001):

- Detecção de erros: o destinatário precisa detectar quando ocorreram os erros. Isso pode ser obtido com uso de técnicas de somas de verificação. Portanto, é necessário incluir informações adicionais ao pacote de dados transmitido;
- Realimentação da parte do destinatário: o único modo de o remetente saber se um pacote foi recebido corretamente ou não é o destinatário fornecer realimentação explícita ao remetente. As respostas de reconhecimento positivo ACK (*Positive Acknowledgement*) ou negativo NACK (*Negative Acknowledgement*) são exemplos dessa realimentação. Em principio, esses

pacotes de realimentação retornam o número de um pacote recebido corretamente pelo destinatário;
- Retransmissão: se um pacote é recebido com erro no destinatário ele é retransmitido pelo remetente.

Estas propriedades permitem implementar mecanismos de controle de erro de transmissão. Existem duas categorias de esquemas para controle de erros de transmissão em sistemas de comunicação de dados: o ARQ (*Automatic-Repeat-reQuest*) e o FEC (*Forward-Error-Correction*). O ARQ é amplamente utilizado por sua simplicidade e confiabilidade, em relação ao FEC. Protocolos padrões IETF, tais como TCP (*Transport Control Protocol*), NETBLT (*Network Block Transfer*), T/TCP (*TCP Extensions for Transactions*) e VMTP (*Versatile Message Transaction Protocol*), utilizam ARQ.

Existem, basicamente, três modos ARQ que podem ser utilizados para controle de erros de transmissão: pare-e-espere, *go-back-n* e repetição seletiva. Eles são a seguir descritos.

1) Protocolo pare-e-espere. Neste protocolo, quando um remetente envia um pacote ao destinatário ele passa a esperar pelo respectivo reconhecimento (ACK ou NACK) referente a esse pacote transmitido. Assim, o remetente não envia novos pacotes até que tenha certeza de que o destinatário recebeu o pacote em questão. Este protocolo atende aos requisitos funcionais de transporte confiável, mas sua implementação tem uma limitação não funcional: desempenho.

2) Protocolo *go-back-n*. Neste protocolo o remetente é autorizado a transmitir múltiplos pacotes (se disponíveis), sem esperar por um reconhecimento individual a cada pacote enviado. Entretanto, ele fica limitado a não ter mais do que um número “n” permitido de pacotes não reconhecidos.

   Este protocolo permite que o remetente potencialmente “encha a rede” com pacotes, evitando-se assim os problemas de subutilização de canal do protocolo pare-e-espere. Ao receber um pacote, cabe ao destinatário enviar um reconhecimento (ACK), contendo o número de seqüência do pacote reconhecido.

A figura 2.3 ilustra o diagrama de seqüências para a operação do protocolo *go-back-n*.

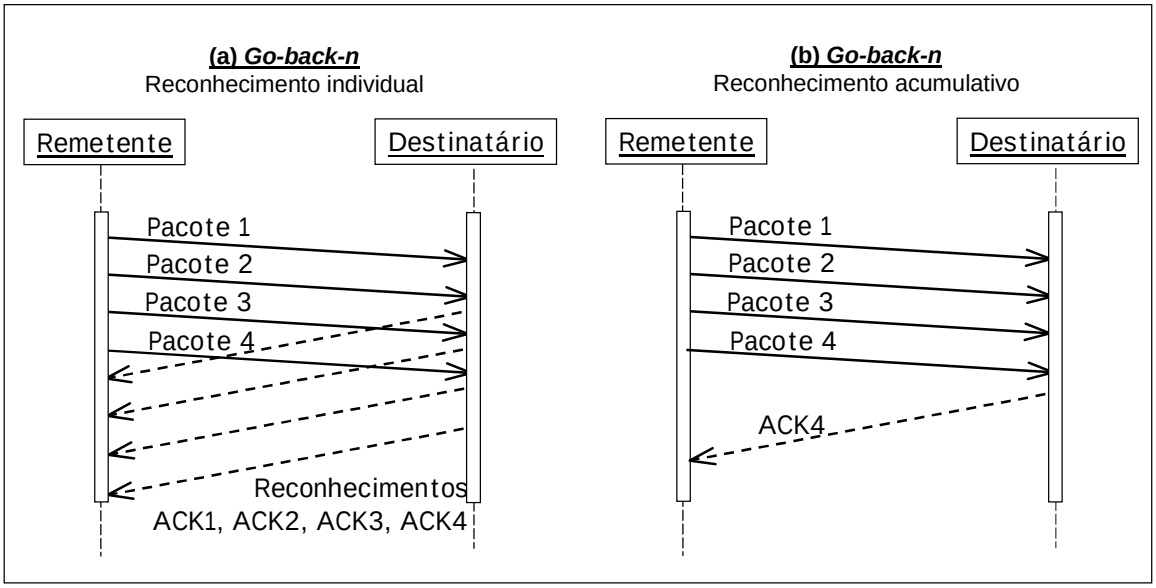


Figura 2.3. Diagrama de seqüência de pacotes no protocolo *go-back-n*.

O protocolo *go-back-n* opera com reconhecimentos individuais para cada pacote, tal como representado na figura 2.3.a. Entretanto, tal como na figura 2.3.b, se o remetente receber um reconhecimento de pacote com número de seqüência ainda não reconhecido (no exemplo, o ACK ao pacote com número de seqüência igual a 4) dentro da janela de reconhecimentos pendentes, esse reconhecimento pode ser interpretado como reconhecimento acumulativo, mostrando que todos os pacotes anteriores ao de número de seqüência igual a 4 (incluindo este) foram corretamente recebidos no destinatário.

Caso um dos pacotes não chegue ao destinatário (na figura 2.4, o pacote com número de seqüência igual a 2), o destinatário deverá enviar reconhecimento referente ao último pacote recebido corretamente (no exemplo, o pacote com número de seqüência igual a 1).

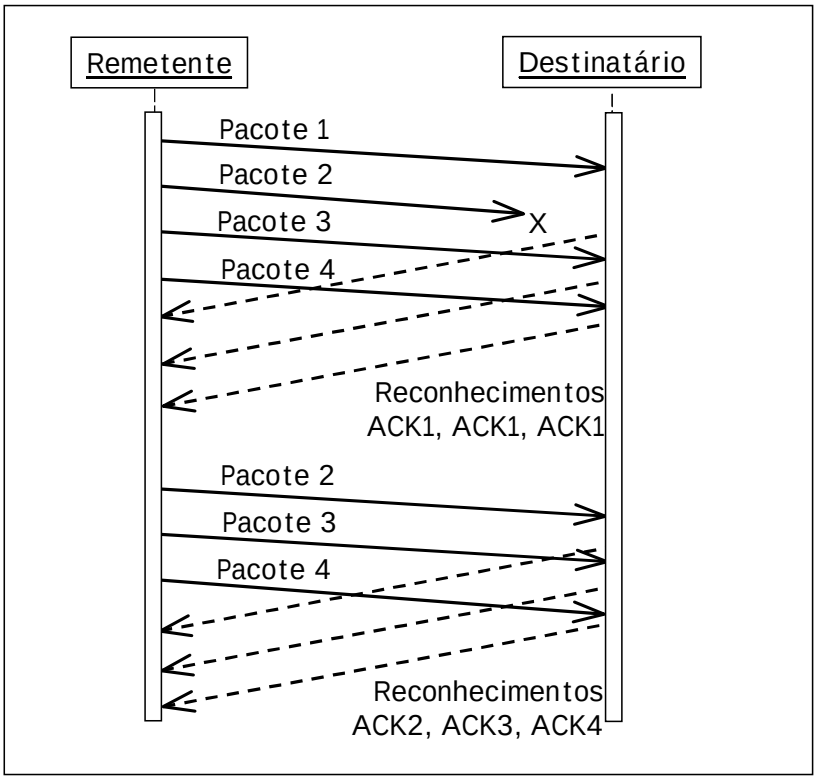


Figura 2.4. Diagrama de seqüência de perdas de pacotes no protocolo *go-back-n*.

Isso faz com que o remetente retransmita automaticamente, após expiração de um temporizador, a seqüência de pacotes posterior ao que foi corretamente recebido (no caso, os pacotes 2, 3 e 4). No protocolo *go-back-n*, o destinatário descarta os pacotes que chegam fora de ordem. Ou seja, ao receber um pacote fora de ordem, o destinatário o descarta e envia um reconhecimento referente ao último pacote corretamente recebido.

3) Protocolo repetição seletiva. O protocolo do tipo *go-back-n* sofre com problemas de desempenho, pois um único erro de pacote pode fazer com que um grande número de pacotes seja retransmitido. Para minimizar esses problemas, há o protocolo baseado em repetição seletiva. Os protocolos de repetição seletiva evitam retransmissões desnecessárias por fazerem com que o remetente retransmita somente os pacotes suspeitos de terem sido recebidos com erro, tal como representado na figura 2.5.

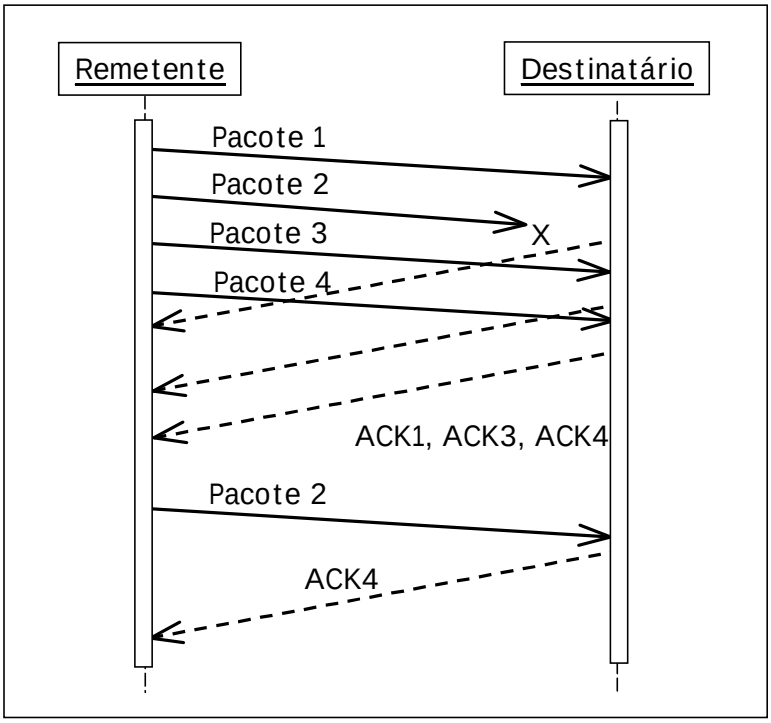


Figura 2.5. Diagrama de seqüência do protocolo repetição seletiva.

O protocolo repetição seletiva opera com reconhecimentos individuais para cada pacote e também suporta reconhecimentos acumulativos. Caso um dos pacotes não chegue ao destinatário (na figura 2.5, o pacote com número de seqüência igual a 2), este continua a receber, aceitar e reconhecer os demais pacotes subseqüentes. O destinatário deve responder com o respectivo reconhecimento a cada pacote recebido dentro dessa janela.

Após a expiração de um temporizador, para um pacote enviado mas ainda pendente de reconhecimento, o remetente retransmite somente esse pacote (no caso, o pacote 2). O protocolo repetição seletiva implementa um mecanismo de controle de erros de transmissão ARQ mais eficiente que os anteriores. Ele é utilizado em protocolos IETF ditos de entrega confiável de dados, como o TCP.

2.3 Os *Web services*

Os *Web services* proporcionam um modo padrão de interoperação entre diferentes aplicações de software, executadas em uma variedade de plataformas e/ou estruturas. Um *Web service* é um sistema de software projetado para suportar

interação máquina-a-máquina em uma rede que tem uma interface descrita em um formato processável por máquina (WSDL – *Web Services Description Language*). Outros sistemas interagem com o *Web service*, em um modo descrito no WSDL, usando mensagens XML, via SOAP (*Simple Object Access Protocol*), tipicamente transportadas via HTTP (W3C, 2004a).

O XML é a escolha natural para o modo de representação dos dados. Muitas especificações utilizam o XML para representação dos dados, assim como os *XML Schemas*, para descrever os tipos dos dados. O SOAP é um protocolo leve para troca de informações.

O WSDL é uma linguagem baseada em XML, com a finalidade de documentar as mensagens que os *Web services* aceitam e geram. Também é necessária uma forma de localização dos *Web services*. O protocolo *Disco* (*Discovery Protocol*) define um formato para o documento *discovery* e um protocolo para devolver esse documento, possibilitando a localização dos serviços em um *web site* conhecido. No entanto, é comum que não se saiba as URLs (*Uniform Resource Locator*) onde os serviços podem ser encontrados. O UDDI (*Universal Description, Discovery, and Integration*) é um mecanismo para os fornecedores *Web services* anunciarem a existência de seus serviços e para os consumidores *Web services* localizarem os serviços de seu interesse.

A WSA (*Web Services Architecture*), publicada pelo W3C, provê a especificação de uma arquitetura de referência de *Web services*. Um *Web service* é implementado por um agente. Ele é a peça concreta de hardware ou software que recebe e envia mensagens. Entretanto, o W3C se exime de direcionar aspectos da implementação e de distribuição de agentes *Web services*. O W3C está somente focado em especificar a interoperabilidade, ou seja, a habilidade de dois ou mais componentes em trocar informação entre si (W3C, 2004a, 2004b).

2.3.1 Órgãos de padronização Web services

As principais entidades científicas que participam dos processos de padronização em *Web services* são a IETF (*Internet Engineering Task Force*), a OASIS/UDDI

(*Organization for the Advancement of Structured Information Standards & Universal Description Discovery and Integration*), o W3C (*World Wide Web Consortium*) e a WS-I (*Web Services Interoperability Organization*).

A IETF é responsável por alguns dos protocolos suportados pelos *Web services* para transporte de mensagens SOAP. A OASIS/UDDI possui comitês técnicos para avaliação de novos padrões em diversas áreas de *Web services* (ex: segurança e mensagem confiável). O W3C atua na padronização da arquitetura *Web services* e do protocolo SOAP. Por fim, a WS-I tem um papel de moderador entre as entidades científico-acadêmicas e a indústria.

A WS-I também objetiva a promover a interoperabilidade entre as implementações *Web services* da indústria, pela publicação dos *basic profiles*. Os *basic profiles* são descrições de convenções e práticas para o uso de combinações específicas de padrões *Web services*, por meio das quais os sistemas poderão se interagir. Desta forma, a indústria de software e seus diferentes fornecedores podem então prover implementações compatíveis entre si (WSI, 2004).

2.3.2 Confiabilidade em *Web Services*

A confiabilidade de software é vista como “a habilidade de um sistema ou componente em executar suas funções requeridas sobre certas condições por um período específico de tempo” (IEEE, 1990).

O foco em confiabilidade dos *Web Services* não está exatamente no contexto de erros de sintaxe ou erros por má qualidade de desenvolvimento de aplicações. Há um conjunto suficiente de causa de erros a ser tratado, no nível de perda de conexões de rede e queda de servidores, durante a execução de transações. Em *Web services* é possível endereçar os aspectos de confiabilidade em alguns níveis distintos: a entrega confiável e previsível de serviços de infra-estrutura (tal como entrega confiável de mensagem); as interações confiáveis e previsíveis entre serviços; o comportamento confiável e previsível de agentes solicitantes e agentes provedores de serviços (W3C, 2004a).

Em alusão ao primeiro desses aspectos (entrega confiável de mensagem), a indústria de tecnologia de informação, que utiliza os *Web services* há mais de cinco anos, aponta essa como área-chave ainda pendente de uma solução eficiente. “Entrega confiável de mensagens” ou “mensagem confiável”, em *Web services*, advém de “*reliable messaging*”. O conceito de mensagem confiável, em *Web services*, refere-se à garantia de que uma mensagem será entregue e que tanto o remetente quanto o destinatário terão a mesma compreensão do status da entrega (W3C, 2004a, 2004b).

Neste contexto, não é possível utilizar eficazmente *Web services* se seus participantes não têm a certeza da troca confiável de mensagens. Basicamente, é preciso garantir a entrega ordenada, única e íntegra de mensagem. Sem um padrão para esta questão, as aplicações precisam tratar isso dentro de sua lógica. Isso, por sua vez, representa um ônus aos desenvolvedores de lógica de negócios e, não menos importante, impacta na interoperabilidade (BIRMAN; RENESSE; VOGELS, 2004a; ERRADI; MAHESHWARI, 2005; KAYE, 2004; MUKHI et al., 2004a; MUKHI; KONURU; CORBERA, 2004b; PALLICKARA; FOX, 2005; SLEEPER, 2004).

2.3.3 Independência de protocolo de transporte

Uma das alternativas em obter entrega confiável de mensagens em *Web services* é a adoção de um protocolo confiável de transporte (por exemplo, utilizando transporte baseado em filas de mensagens), sobre o qual mensagens SOAP possam ser trocadas entre *endpoints*. Entretanto, a WSA não restringe a infra-estrutura de transporte somente aos protocolos confiáveis. A WSA se define como independente do protocolo de transporte.

Há muitos protocolos de comunicação para transporte de mensagens, como o FTP (*File Transfer Protocol*), o HTTP (*HyperText Transport Protocol*), o SMTP (*Simple Mail Transport Protocol*) e o JMS (*Java Messaging Service*) nos *Web services*. O protocolo HTTP, devido a sua difusão na Internet, é o mais utilizado por implementações *Web services* para transporte de mensagens. Entretanto, esse protocolo não é considerado confiável, sob o aspecto de entrega confiável de

mensagens para *Web services* (ENDREI et al., 2004; OASIS, 2004b; TAI; MIKALSEN; ROUVELLOU, 2003).

O fato de a arquitetura dos *Web services* ser independente de protocolo de transporte para mensagens SOAP dá margem às falhas. Ou seja, a entrega confiável de mensagens só ocorre quando se adota um protocolo de transporte confiável entre o remetente e o destinatário. Por outro lado, considerando-se o aumento de implementações de *Web services*, é imprevisível a um *endpoint* assumir que um protocolo confiável de transporte esteja sendo usado por todo o caminho a ser percorrido por uma mensagem.

A figura 2.6 representa um exemplo genérico e factível, onde uma mensagem transmitida por um *endpoint* remetente "A" atravessa múltiplas infra-estruturas de transporte (com diferentes protocolos), até atingir seu *endpoint* destinatário "F". Dependendo das características de conectividade entre os dois *endpoints*, é possível que haja trechos que utilizem protocolos não confiáveis.

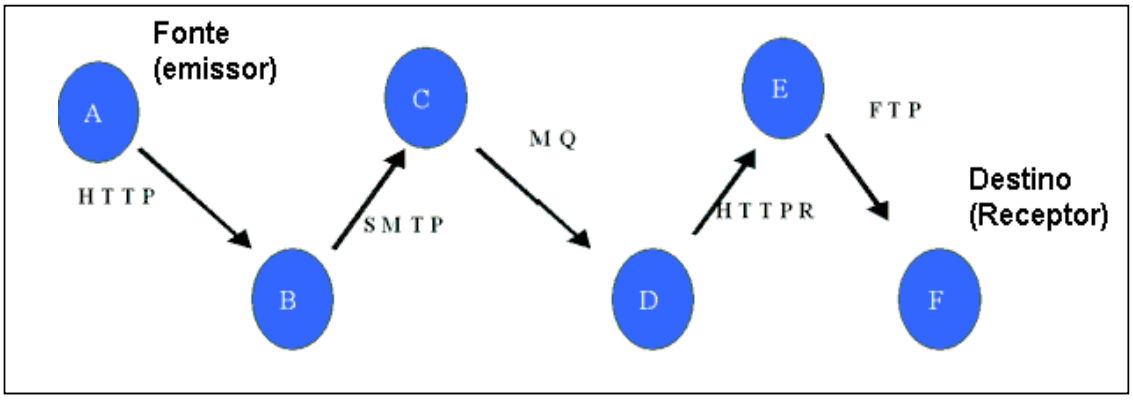


Figura 2.6. Envio de mensagem entre dois *endpoints*, por múltiplos protocolos de transporte.

Este exemplo é um caso extremo. Entretanto, dada a característica de independência de protocolo de transporte, bem como o presente aumento de implementações de *Web services* utilizando infra-estruturas de transporte baseadas em vários protocolos, entre eles o HTTP, torna-se imprevisível para um remetente assumir que um único protocolo confiável esteja sendo usado por todo o caminho a ser percorrido por uma mensagem por ele enviada até o respectivo destino.

2.3.4 Mensagem confiável em *Web Services*

A WSA define um modelo orientado a mensagens que foca em aspectos de arquitetura relacionado às mensagens e seu processamento. Neste modelo, não se consideram semânticas relacionadas ao conteúdo da mensagem, bem como sua relação com outras mensagens. Entretanto, o modelo foca na estrutura da mensagem, no relacionamento entre remetentes e destinatários de mensagens e como as mensagens são transmitidas (W3C, 2004a).

As características mínimas esperadas em um padrão de entrega confiável de mensagens para os *Web services* são: entrega garantida, entrega íntegra, entrega ordenada e entrega não duplicada (OASIS, 2004a; W3C, 2004a).

A WSA não especifica como implementar o suporte específico à entrega confiável de mensagens. Ela apenas indica que isso pode ser obtido estendendo-se as estruturas de cabeçalhos e corpo de mensagens no SOAP, a fim de suportar informações adicionais de controle de mensagens. Assim, uma infra-estrutura de entrega confiável de mensagens pode ser implementada de forma independente das aplicações, dos serviços e dos protocolos de transporte.

A WSA menciona *WS-Reliability* e *WS-ReliabilityMessaging* como as duas especificações de padrões de mensagem confiável (neste trabalho, chamadas de EPMCs), submetidas e atualmente em avaliação pelas entidades de padronização em *Web services*. Esses novos padrões permitirão implementar um *middleware*[1] aberto orientado a mensagem, implementando entrega confiável de mensagens em *Web services* (OASIS, 2004c, W3C, 2004a, 2007).

2.3.5 Padrões dependentes de mensagem confiável

A importância de adoção e implementação de um padrão de mensagem confiável em *Web services* não visa apenas a suportar interoperabilidade, mas também a viabilizar outros padrões não funcionais em *Web services*. Esses outros padrões são

[1] Camada de *software* que “une” e permite que sistemas diferentes (que podem estar em computadores diferentes) executem uma tarefa juntos. Sua principal função é permitir comunicação entre componentes diferentes de *software*.

conduzidos por comitês técnicos da OASIS e W3C, nas áreas de gerenciamento, segurança, transações e coreografia. Tais comitês são abordados a seguir:

1) O comitê técnico WSDM (*Web Services Distributed Management*) está definindo padrões para gerenciamento de recursos distribuídos usando *Web services*. Esses padrões dividem-se em duas áreas primárias: MOWS (*Management Of Web Services*) que corresponde ao gerenciamento de *Web services* e MUWS (*Management Using Web Services*), que corresponde ao uso de *Web services* para realizar funções de gerenciamento em outros recursos computacionais. A versão 1.0 dessas especificações foi publicada em 2004. Em 2005 iniciou-se uma revisão dessas, gerando a atual versão 1.1, classificada como padrão pela OASIS. Embora não haja uma menção explícita sobre dependência do padrão WSDM com mensagem confiável, o WSDM tem como pré-requisitos as especificações OASIS *WS-Security* (segurança em *Web services*) e *WS-BaseNotification* (notificação e eventos em *Web services*). Essas especificações, por sua vez, têm sua dependência com o contexto de mensagem confiável (OASIS, 2005a).

2) O comitê técnico OASIS chamado *Web services Notification* visa a definir uma coleção de especificações (chamada *WS-BaseNotification*) que padronizam o modo de interação entre *Web services*, utilizando “notificações” ou “eventos”. Essas especificações também formam os fundamentos para a implementação de arquiteturas orientadas a eventos baseadas em *Web services*. As especificações do *WS-BaseNotification*, atualmente em versão 1.3 de padrão OASIS, apontam a necessidade de entrega confiável de mensagem, pois espera-se que uma notificação enviada por um *Web services* seja entregue ao respectivo destinatário. Como requisitos, a *WS-BaseNotification* deve ser composta com outras especificações *Web services*, tais como *WS-Security* (segurança em *Web services*), *WS-Coordination* (coordenação de transações em *Web services*) e *WS-ReliableMessaging* (mensagem confiável em *Web services*). Portanto, este padrão depende intrinsecamente de um padrão de mensagem confiável (OASIS, 2006b).

3) Coreografia é o modelo de seqüência e condições em que múltiplos agentes *Web services* cooperativos e independentes trocam informações para realizar em conjunto determinada função. O grupo de trabalho para a padronização de coreografia em *Web services* publicou em 2004 a versão preliminar (*draft*) do modelo de coreografia em *Web services*. Essa publicação, no tópico que define interações, indica que a troca de informações deve ser suportada por protocolos baseados em mensagens que implementem entrega confiável. Tem-se, portanto, que o futuro padrão de coreografia em *Web services* pressupõe, como requisito, a adoção de um padrão de mensagem confiável (W3C, 2004d, 2004f).

4) O comitê técnico OASIS WSS[2] (*OASIS Web Services Security*) tem por objetivo prover base técnica para especificar funções de segurança em mensagens aplicadas a *Web services*. Em 2006 a OASIS publicou especificações de extensão do protocolo SOAP para suporte a funções de segurança, que correspondem não a um padrão isolado e independente, mas um padrão a ser combinado com outros padrões para sua complementação. Embora o contexto de mensagem confiável esteja fora do escopo deste padrão, a própria OASIS ratifica que o uso de um padrão de mensagem confiável integra e complementa outras especificações *Web services*, tais como a *WS-Security* (OASIS, 2006a, 2006d).

5) O propósito do comitê técnico OASIS WS-TX (*OASIS Web Services Transaction*) é definir um conjunto de especificações para coordenar ações de aplicações distribuídas. Em 2006 esse comitê técnico publicou as versões preliminares (*draft*) 1.1 das chamadas especificações de transações *Web services*. Essas especificações visam definir mecanismos para interoperabilidade transacional entre *Web services*. Elas descrevem uma estrutura (*framework*) de coordenação chamada *WS-Coordination*, que atua em conjunto com outras duas especificações do mesmo comitê, sendo uma para transações de curta duração (*WS-AtomicTransaction*) e outra para transações de negócios de longa duração (*WS-BusinessActivity*). As especificações do futuro padrão WS-TX apontam ser extremamente recomendado que transações em *Web services* sejam

[2] WSS também é referenciada como *WS-Security* pela OASIS.

implementadas em conjunto com especificações de segurança e de mensagem confiável em *Web services*. Além disso, uma vez que a OASIS coloca o contexto de mensagem confiável como complementar ao contexto de *WS-Security*, tem-se mais um indício de dependência do padrão de transações a um padrão de mensagem confiável (OASIS 2006e).

2.4 As opções de padrão de mensagem confiável em *Web services*

Uma EPMC (sigla adotada no presente trabalho, referindo-se a Especificação de Padrão de Mensagem Confiável), corresponde a uma especificação de protocolo de aplicação, com abordagem de extensão ao padrão SOAP, proposta às entidades de padronização *Web services*, visando a tornar-se o padrão oficial para implementar entrega confiável de mensagens em *Web services*. Atualmente, *WS-Reliability* e *WS-ReliableMessaging* são as duas EPMCs, citadas pelo W3C como as opções de futuro padrão a ser adotado em *Web services* (W3C, 2004a).

2.4.1 A EPMC *WS-Reliability*

A EPMC *WS-Reliability* é uma proposta de padrão apresentada à OASIS em 2004 por um consórcio formado pelas empresas Fujitsu, Novell, Oracle e Sun. Ela consiste em uma extensão de especificação do padrão SOAP, visando a atender aos requerimentos críticos de entrega confiável de mensagem em arquiteturas *Web services*. Essa EPMC também especifica associações (*bindings*) pertinentes ao protocolo HTTP para implementar trocas de mensagens (OASIS, 2004b).

Com uma estrutura baseada em XML, a EPMC *WS-Reliability* tem seu mecanismo de reconhecimento baseado em “reconhecimento positivo”. Isso implica que todas as detecções de erro, inicializações de correção de erros e retransmissões de mensagens perdidas são realizadas pelo *endpoint* remetente. Além disso, um remetente deve ser pró-ativo, iniciando retransmissões automáticas de mensagem a partir da simples ausência de recebimento de reconhecimento de uma mensagem enviada, após um intervalo de tempo pré-definido.

A EPMC *WS-Reliability* endereça a questão de ordenação e detecção de mensagens duplicadas, por meio do uso de três parâmetros identificadores[3] (IDs) distintos: identificador de numeração de seqüência de mensagem, identificador de grupo e identificador de mensagem.

O conceito de grupo de mensagem, presente nesta EPMC pelo uso de um identificador de grupo, advém da possibilidade de que as mensagens possam pertencer a diferentes categorias de aplicações de negócio.

O identificador de numeração de seqüência de mensagem, nessa EPMC, chama-se *SequenceNumber* e corresponde ao número de seqüência da mensagem dentro de um grupo. Entretanto, esse elemento não tem sua presença obrigatória em todas as mensagens. A prática de se incluir o identificador de numeração de seqüência de mensagem ocorre somente para controlar a ordem de entrega. O *SequenceNumber* inicia-se sempre com zero e, durante uma transmissão de mensagens de um grupo, caso esse elemento exceda seu valor máximo (*rollover*), o *endpoint* remetente gera um novo identificador de grupo e inicia uma nova seqüência de transmissão (após obter o reconhecimento referente à última mensagem enviada no grupo antigo).

Nesta EPMC, um *endpoint* destinatário deve enviar um reconhecimento (*acknowledgement*) ao *endpoint* remetente para cada mensagem recebida. Os reconhecimentos são baseados em identificadores únicos de mensagens recebidas.

As detecções de duplicações e ordenação são realizadas sempre pelo *endpoint* destinatário. Nesta EPMC a numeração das mensagens tem apenas a função de controle de ordenação pelo destinatário. A detecção de duplicidade de mensagens é feita a partir do identificador de mensagens, que é único, associado a cada mensagem.

Esta EPMC também utiliza marcadores de tempo (*timestamps*), onde tanto mensagens quanto grupos de mensagens podem ser considerados inválidos, uma vez expirado o tempo associado a essas. O controle de expiração é baseado no

[3] Identificador é um nome, endereço, etiqueta ou índice de distinção de um objeto em um programa de computador (IEEE, 1990).

UTC (*Universal Time Coordinated*) e permite atribuir a cada mensagem o seu período de expiração, por meio do parâmetro *ExpirityTime*, presente no cabeçalho da mensagem.

Além disso, temporizadores são incorporados para inicialização de retransmissões (devido ao mecanismo de reconhecimento positivo, onde o *endpoint* remetente deve retransmitir uma mensagem, caso não receba o respectivo ACK referente a essa, dentro de um intervalo de tempo).

2.4.2 A EPMC *WS-ReliableMessaging*

A EPMC *WS-ReliableMessaging*, republicada recentemente pela BEA, IBM, Microsoft e TIBCO, define um protocolo que objetiva assegurar que as mensagens sejam entregues de forma confiável entre dois *endpoints*. Essa proposta de padrão foi submetida à avaliação da OASIS em 2005. Em agosto desse mesmo ano foi publicada sua versão 1.1 de especificação, para revisão pública.

O objetivo desta EPMC é prover um protocolo para a troca de mensagens com garantia de entrega, não duplicação e garantia de entrega ordenada de mensagens. Trata-se de uma especificação baseada em SOAP que atende aos requerimentos de entrega confiável de mensagens, críticos a algumas aplicações de *Web services* (OASIS, 2006c).

Tal como a EPMC *WS-Reliability*, a EPMC *WS-ReliableMessaging* também espera que empresas fornecedoras de software incorporem essa como padrão em componentes de software reutilizáveis para suporte a aplicações *Web services*, de forma a eximir os desenvolvedores do esforço da implementação de entrega confiável de mensagem, em escopo de aplicação. Adicionalmente, propõe-se que empresas fornecedoras de *middleware* passem a utilizar a EPMC *WS-ReliableMessaging* como um protocolo de interoperabilidade SOAP, visando a conectar seu ambiente orientado a mensagens com outros ambientes orientados a mensagem de outros fornecedores de software (FERGUSON et al., 2003).

A EPMC *WS-ReliableMessaging* especifica uma estrutura XML introduzindo os elementos necessários para prover entrega confiável de mensagens, através de uma extensão do padrão SOAP. Esta EPMC também faz uso de mecanismo de reconhecimento baseado em reconhecimento positivo (*ACK*). Ou seja, as detecções e retransmissões de mensagens perdidas são realizadas pelo *endpoint* remetente, que também faz retransmissão automática de uma mensagem a partir do não recebimento de reconhecimento desta, após um intervalo pré-definido. Opcionalmente, essa EPMC suporta reconhecimento negativo (NACK).

A EPMC *WS-ReliableMessaging* trata a questão de ordenação e detecção de mensagens duplicadas, por meio do uso de dois IDs (identificadores) distintos: identificador de numeração de seqüência <*MessageNumber*> de mensagem e identificador de grupo <*Identifier*>. A associação desses dois identificadores forma o identificador único da mensagem, visando a detecções de duplicidade. As detecções de duplicações e ordenação são realizadas no lado do *endpoint* destinatário.

O conceito de grupo de mensagem também é suportado por essa EPMC, atendendo a requisitos de classificação de mensagens em diferentes grupos (grupo nesta EPMC é chamado de seqüência), visando a prover níveis de serviço diferenciados, de acordo com aplicações de negócio. Nesta EPMC, toda mensagem é considerada com integrante de um grupo de mensagens. Mesmo no caso de haver uma única mensagem, esta é considerada como parte de um grupo de mensagens.

Como visto, nesta EPMC o identificador de numeração de seqüência de mensagem é um elemento chamado <*MessageNumber*>, que é associado com cada mensagem dentro de um grupo. Este elemento é utilizado para identificar o inicio e a posição (ordem) das mensagens em um grupo de mensagens. O elemento <*MessageNumber*>, cujo valor começa sempre com 1 (um), é obrigatório em todas as mensagens.

Nesta EPMC, um *endpoint* destinatário não precisa enviar um reconhecimento ao *endpoint* remetente para cada mensagem recebida. Esta EPMC, suporta reconhecimento acumulativo.

As retransmissões são sempre iniciadas pelo *endpoint* remetente, devido à característica de reconhecimento positivo. Esta EPMC estabelece o intervalo de retransmissão, baseado no padrão RTTM (*Round Trip Time Measurement*), RFC 1323 da IETF, para uma seqüência de mensagens.

O conceito de expiração está presente somente no contexto de grupo. Não existe controle de expiração por mensagem e, no processo de iniciação de uma seqüência, um dos elementos que compõem a sintaxe de pedido de criação é o <*wsrm:expires*>. Este elemento é do tipo *xs:duration*[4] e especifica a duração requerida pelo *RM Source* para uma seqüência.

2.5 Verificação e avaliação de especificações de arquiteturas

Uma das maiores questões em desenvolvimento de software hoje é qualidade. A ISO/IEC 9126-1:2001 define que confiabilidade faz parte das categorias de atributos de qualidade de software. A idéia de prever a qualidade de um produto de software a partir de uma descrição de alto nível não é recente (DOBRICA; NIEMELA, 2002).

A busca e a resolução de problemas por meio de verificação aumenta a qualidade de software e pode reduzir significativamente os custos de projeto, desenvolvimento e testes de implementação. Sendo assim, testes baseados em requisitos devem ser realizados antes da implementação por meio de verificação. Eles são considerados partes integrantes do processo de desenvolvimento e têm o objetivo de verificar requisitos utilizando-se uma abordagem por pontos de vista, obtendo-se uma matriz de testes de requisitos (CHEW; SULLIVAN, 2000; DOBRICA; NIEMELA, 2002; RAKITIN, 2001; RAMA, 1996).

2.5.1 Verificação de software

Antes que uma única linha de código de software seja escrita ou que um hardware seja fabricado, é necessário rever seu projeto para assegurar que este esteja em conformidade com seus requisitos. A verificação é a confirmação, por exame e

[4] *Duration* é um tipo de dados primitivo de XML (W3C, 2001) que representa uma duração de tempo, com seis dimensões (ano, mês, dia, hora, minutos e segundos), de acordo com a norma ISO 8601, de 1988.

provisionamento de evidência ou indício, de que requerimentos especificados foram contemplados, ou seja, o software está em conformidade com os requisitos (CHEW; SULLIVAN, 2000; IEEE, 1990, 1996, 2004; PRESSMAN, 2001; RAKITIN, 2001).

Verificação lida com o conceito de “construindo o modelo certo” e serve como um previsor de qualidade, antes de se implementar um sistema. Ela está relacionada com a especificação da arquitetura, ou seja, a estrutura organizacional de um sistema, que, por sua vez, é composto por componentes de software, suas propriedades e relacionamentos (ALBIN, 2003; BALCI, 1998; BASS; CLEMENS; KAZMAN, 2003; IEEE, 1990).

A verificação de duas ou mais arquiteturas viabiliza a comparação entre elas a partir dos resultados das verificações individuais. Portanto, verificação também permite avaliar a precisão com que duas arquiteturas são equivalentes (BALCI, 1998; CHEW; SULLIVAN, 2000; DOBRICA; NIEMELA, 2002; PUTMANN, 2001).

Verificação envolve dois contextos de arquiteturas de software: o contexto conceitual e o contexto de projeto. O contexto conceitual engloba a especificação da arquitetura do sistema almejado. Trata-se de especificação referencial que compreende os componentes, requisitos, interações e comportamentos definidos para uma arquitetura. O contexto de projeto é composto pela especificação da arquitetura do software que será implementado - a documentação do software a ser verificado, antes de sua implementação. Esse contexto dita quais são os atributos de qualidade que são contemplados no sistema (IEEE, 2004; NIST, 1996).

A verificação consiste em avaliar o atendimento aos requisitos funcionais, determinando seu correto mapeamento entre os dois contextos, de projeto e conceitual. Este processo de mapeamento[5], dá origem a uma matriz com o mapeamento dos requisitos almejados e os contemplados na especificação da arquitetura (BARCELOS, 2006; CHEW; SULLIVAN, 2000).

O processo de verificação é uma atividade estática (avalia a precisão do modelo projetado e não requer sua execução). Seu objetivo é estabelecer o mapeamento

[5] Também chamado de análise de rastreabilidade.

entre elementos individuais de um modelo com outro: requisitos de software, do contexto de projeto, com requisitos de sistema, do contexto conceitual (DoD, 2001; IEEE, 2004; NIST, 1996).

A partir da verificação obtém-se uma matriz de mapeamento dos requisitos. Esta, por sua vez, registra o mapeamento entre os requisitos contemplados (contexto de projeto) e os requisitos objetivados (contexto conceitual) em um dado componente de software.

Por meio de avaliação de especificação de arquitetura de software é feita verificação, que, por sua vez, aplica técnicas de questionamento e inspeção [6] sobre a especificação da arquitetura de software, tomando-se por base uma especificação de referência (ALBIN, 2003; BARCELOS, 2006; BASS; CLEMENS; KAZMAN, 2003; DOBRICA; NIEMELA, 2002; PARNAS; WEISS, 1985).

2.5.2 Características de processo de verificação

Um processo de verificação tem características intrínsecas a seguir listadas (BALCI, 1998; IEEE, 2004; RAKITIN, 2001):

- Os resultados de um processo de verificação não devem ser considerados como binários, ou seja, não se pode considerar que apenas a especificação de uma arquitetura de software verificada esteja completamente correta ou incorreta;
- O processo de verificação deve ser imparcial, ou seja, conduzido por entidade externa à equipe de desenvolvimento;
- Verificação é um processo difícil, requer criatividade e conhecimento sobre o contexto;
- Num processo de verificação não se esperam testes e simulações;
- Um processo de verificação deve ter resultados documentados.

A definição da arquitetura de software é considerada como o primeiro produto em um processo de desenvolvimento e, a partir desse ponto de vista, a análise a esse

[6] Inspeção é uma técnica de análise estática, baseada em um exame visual a produtos de desenvolvimento para detectar erros, violações de padrões de desenvolvimento e outros problemas. Inspeção é uma técnica que pode atingir significantes resultados em melhoria de qualidade de software (IEEE, 1990; RAKITIN, 2001).

nível deveria revelar conflitos de requisitos e descrições incompletas de projeto (DOBRICA; NIEMELA, 2002; IEEE, 1990).

Segundo o padrão IEEE 1061-1992, qualidade de software representa o grau com que um software possui desejada combinação de atributos. Um dos propósitos de avaliar a especificação de arquitetura de um sistema de software é verificar se os requisitos de qualidade foram endereçados no projeto para identificar riscos potenciais, antes dele ser implementado (ALBIN, 2003; DOBRICA; NIEMELA, 2002; PARNAS; WEISS, 1985).

Não se trata de avaliar o produto de software em si (por exemplo, o código executável ou o código fonte), mas sim a descrição de projeto arquitetural do sistema. O projeto arquitetural é o resultado do processo de definição dos relacionamentos entre componentes e de suas interfaces, para estabelecer uma estrutura que permita o desenvolvimento de um sistema. Ele é representado pela descrição de arquitetura enquanto ainda não é parte de um produto final, e é um ponto chave no projeto e implementação do produto (ALBIN, 2003; IEEE, 1990).

Avaliações de arquitetura [7] constituem verificação e servem como previsor de qualidade antes da construção do sistema. Dos métodos de avaliação arquitetural atuais, destacam-se o SAAM (*Scenario-based Architecture Analysis Method*) e o ATAM (*Architecture Trade-off Analysis Method*), por servirem como base para a criação de grande parte dos demais métodos. Esses dois métodos buscam avaliar a arquitetura em relação a determinados requisitos de qualidade e utilizam técnica de avaliação baseada em execução de cenários que representam o comportamento esperado do software em relação a uma determinada característica de qualidade (ALBIN, 2003; BASS; CLEMENS; KAZMAN, 2003).

Entretanto, esses métodos apresentam alguns problemas: grande subjetividade, elevado custo de aplicação, dificuldades para avaliar simultaneamente o atendimento a vários requisitos de qualidade e contexto limitado para a aplicação de alguns dos métodos (BABAR et al., 2004; BARCELOS, 2006; DOBRICA; NIEMELA, 2002).

[7] Também conhecida por avaliação arquitetural, ela verifica principalmente se as informações descritas no documento do software estão consistentes e se a arquitetura nele representada atende aos requisitos especificados para o produto (BARCELOS, 2006).

Uma outra forma de avaliação é a abordagem por inspeção, utilizando-se técnicas de *checklist* (também chamadas de técnicas de questionamentos). Elas têm sido consideradas como mais adequadas para serem aplicadas durante a inspeção de uma especificação de arquitetura de software (BARCELOS, 2006; CONRADI et al., 2003; SHULL, et al., 2000).

Portanto, questionamentos são aplicáveis a processos de verificação de software em avaliações de especificação de arquitetura. O processo de responder aos questionamentos de revisão de arquitetura envolve a análise e inspeção da especificação da arquitetura, sendo esta a melhor forma de fazê-lo (NASA, 1993; RAKITIN, 2001; RAMA, 1996; PARNAS; WEISS, 1985).

O principio básico do processo de inspeção está em examinar com atenção uma especificação, com relação a uma outra de referência. A inspeção é uma análise estática, baseada em um exame visual em produtos de desenvolvimento para detectar erros, violações de padrões de desenvolvimento e outros problemas, auxiliando na verificação. Ela é considerada uma técnica formal de revisão (FTR - *Formal Technical Review*), aplicada para verificar se o software especificado atende aos seus requisitos (IEEE, 2004; NASA, 1993; NIST, 1996; PRESSMAN, 2001; RAKITIN, 2001).

3 O MÉTODO DE VERIFICAÇÃO

Este capítulo apresenta o Método de Verificação a ser aplicado às duas EPMCs (Especificações de Padrão de Mensagem Confiável) *WS-Reliability* e *WS-ReliableMessaging*, viabilizando-se obter suas respectivas Matrizes de Mapeamento de Requisitos.

3.1 A estrutura do método

A verificação aqui deve alcançar os seguintes objetivos: identificar os critérios de verificação e realizar a verificação (IEEE, 1996).

Neste sentido, o Método de Verificação proposto no presente trabalho possui elementos relativos a esses dois objetivos citados, que são aqui chamados de Modelo de Referência e Processo de Verificação, respectivamente. Além desses, existem mais dois elementos que são a Matriz de Mapeamento de Requisitos e a Especificação de Padrão de Mensagem Confiável. Todos esses elementos são representados na figura 3.1.

O contexto de projeto é composto pela Especificação de Padrão de Mensagem Confiável. Uma EPMC é a especificação da arquitetura a ser verificada antes de sua implementação. No presente trabalho são verificadas as duas EPMCs citadas pelo W3C: *WS-Reliability* e *WS-ReliableMessaging*.

O contexto conceitual é composto pelo Modelo de Referência. Ele provê a especificação de uma arquitetura de referência baseada em RM-ODP, definindo as partes envolvidas, regras, contratos, os relacionamentos entre componentes e suas interfaces, focados em entrega confiável de mensagens.

O Processo de Verificação é uma atividade estática, pois verifica os requisitos presentes nas EPMCs e não requer a execução destas. O objetivo do Processo de Verificação é estabelecer o mapeamento entre requisitos individuais de um contexto de projeto (as EPMCs) e os requisitos do contexto conceitual (definidos no Modelo de Referência para entrega confiável de mensagens).

Tanto o Modelo de Referência quanto a Especificação de Padrão de Mensagem Confiável são utilizadas pelo Processo de Verificação. Por esta razão, representam-se setas indicativas que convergem desses para o elemento Processo de Verificação.

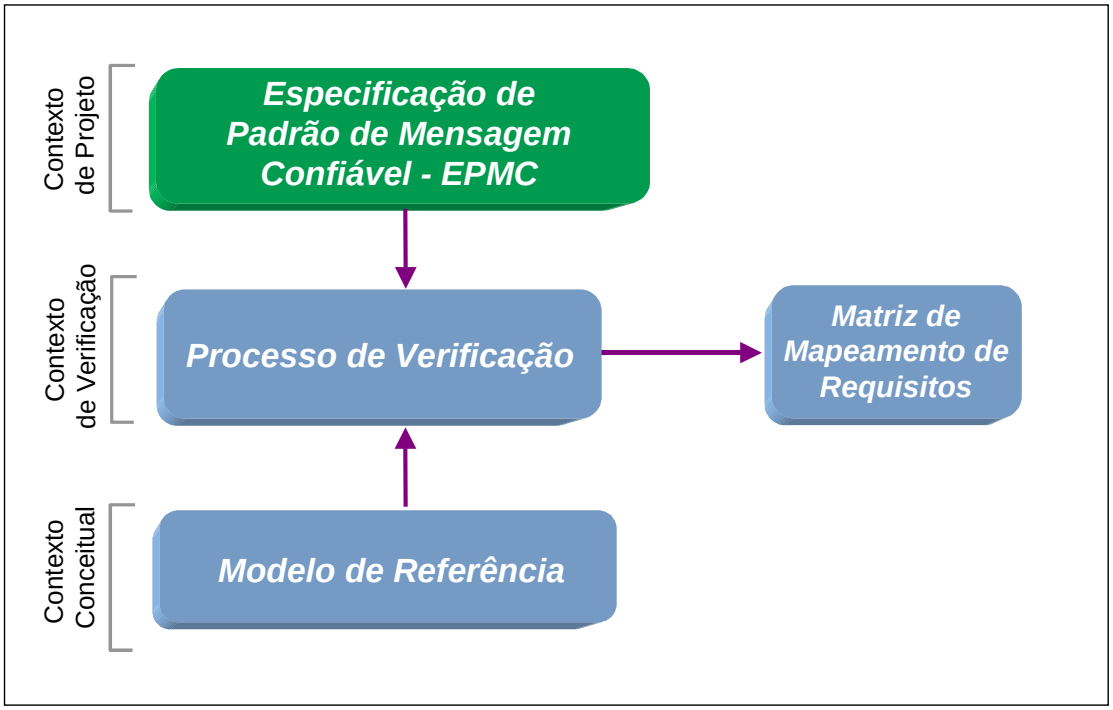


Figura 3.1. Elementos da estrutura do Método de Verificação.

Como resultados de verificação - saída do Processo de Verificação - tem-se a Matriz de Mapeamento de Requisitos. Trata-se de uma estrutura tabular que registra um mapeamento entre os requisitos objetivados para entrega confiável de mensagens, definidos no Modelo de Referência, e os requisitos especificados na EPMC.

Dos elementos que compõem o Método de Verificação, o elemento Especificação de Padrão de Mensagem Confiável está definido e disponível em (OASIS, 2004b, 2006c).

Como critério de seleção das EPMCs em *Web services* elegíveis à verificação, o presente trabalho considera as EPMCs citadas e submetidas aos principais órgãos de padronização *Web services*, que são o W3C e a OASIS.

Os elementos Processo de Verificação, Modelo de Referência e Matriz de Mapeamento de Requisitos são descritos nos itens subseqüentes.

3.2 O Processo de Verificação

O Processo de Verificação aqui proposto visa a estabelecer as atividades para verificar as Especificações de Padrão de Mensagem Confiável em *Web services*. Seu escopo está representado na figura 3.2.

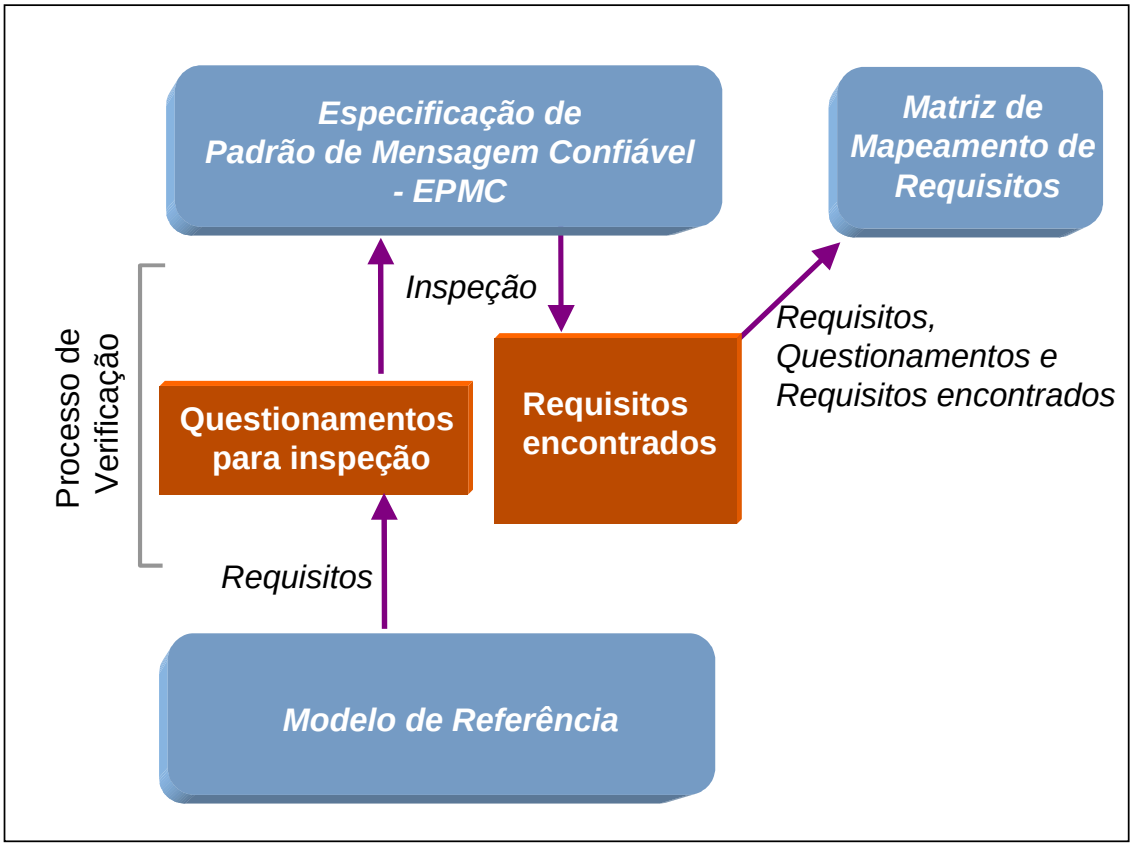


Figura 3.2. O Processo de Verificação.

Para o Processo de Verificação, o Modelo de Referência provê os requisitos que precisam ser verificados na documentação da EPMC. A partir destes requisitos, o

Processo de Verificação realiza avaliação de especificação de arquitetura e esta é feita por técnicas baseadas em questionamento e inspeção (BASS; CLEMENS; KAZMAN, 2003; DOBRICA; NIEMELA, 2002; PARNAS; WEISS, 1985).

Estas técnicas são aplicadas, tomando-se os requisitos do Modelo de Referência e gerando-se questionamentos sobre estes. Um questionamento sobre um requisito visa a direcionar a inspeção sobre como e onde a especificação da arquitetura de software sob verificação (aqui, EPMC), indica atender a este.

Para a elaboração dos questionamentos de inspeção, a partir de requisitos, consideram-se aqui as características (BABAR et al., 2004; BARCELOS, 2006; DOBRICA; NIEMELA, 2002; PARNAS; WEISS, 1985):

- Os questionamentos são qualitativos;
- Eles são criados a partir dos requisitos e respectivos atributos, definidos no Modelo de Referência;
- Os questionamentos são agrupados de acordo com seu nível de abstração;
- Os itens de questionamento objetivam avaliar a abordagem empregada pela especificação (EPMC) para atender aos requisitos e não a forma como eles foram documentados.

O Processo de Verificação, portanto, consiste em fazer avaliações de especificação de arquitetura, aplicando-se questionamento e inspeção referente a cada um dos requisitos definidos no Modelo de Referência sobre a EPMC. Como resultado das inspeções, para cada requisito definido no Modelo de Referência registram-se os respectivos requisitos encontrados[8] na EPMC. O conjunto de todos os requisitos, seus questionamentos e requisitos encontrados na EPMC compõe o resultado final da verificação, que é o elemento Matriz de Mapeamento de Requisitos.

3.3 O Modelo de Referência

O Modelo de Referência para entrega confiável de mensagens é um dos elementos do Método de Verificação proposto no presente trabalho.

[8] Requisitos encontrados, ou resultados de verificação, são chamados de *findings* em verificação de software (CHEW; SULLIVAN, 2000; DoD, 2001; IEEE, 2004; NIST, 1996).

Ele especifica a arquitetura de referência para um protocolo de aplicação de entrega confiável de mensagens. Seu objetivo no Método de Verificação está em estabelecer os pontos de testes de especificação, ou seja, em diferentes níveis de abstração, quais são os requisitos almejados por meio da definição das partes envolvidas, suas interações, políticas e restrições de relacionamentos.

Para a definição do Modelo de Referência utilizam-se a especificação WSA e o RM-ODP.

A especificação WSA (*Web Services Architecture*) foi publicada pelo consórcio W3C de padronização *Web services*. A WSA, na abordagem de interoperabilidade, define-se como um modelo orientado a mensagens. Este, por sua vez, visa à obtenção de um *middleware* aberto orientado a mensagem, implementando entrega confiável de mensagens em *Web services* (OASIS, 2004c; W3C, 2004a, 2007).

Todo o escopo de interoperabilidade por mensagem confiável da WSA está distribuído em um conjunto de referências composto pela descrição de arquitetura, requisitos de arquitetura, cenários de uso e glossário (W3C, 2004a, 2004b, 2004c, 2004d, 2004e, 2004f).

O RM-DOP é utilizado aqui como metamodelo para especificação de arquiteturas. Ele provê uma abordagem sistemática, consistente e estruturada em níveis de abstração, que proporciona a especificação do Modelo de Referência para entrega confiável de mensagens.

O presente Modelo de Referência para entrega confiável de mensagens utiliza o conceito de níveis hierárquicos de abstração com base em quatro pontos de vista do RM-ODP, conforme representado na figura 3.3. São eles: ponto de vista da empresa, ponto de vista da informação, ponto de vista da computação e ponto de vista da engenharia. Pelo fato de a WSA ser independente de implementação, o presente Modelo de Referência não se estende ao ponto de vista da tecnologia do RM-ODP.
A representação com setas na figura 3.3, indica que a especificação do Modelo de Referência segue do nível de abstração maior para o menor. Para a especificação

dos pontos de vista do Modelo de Referência utilizam-se as notações UML (*Unified Modeling Language*) e linguagem natural.

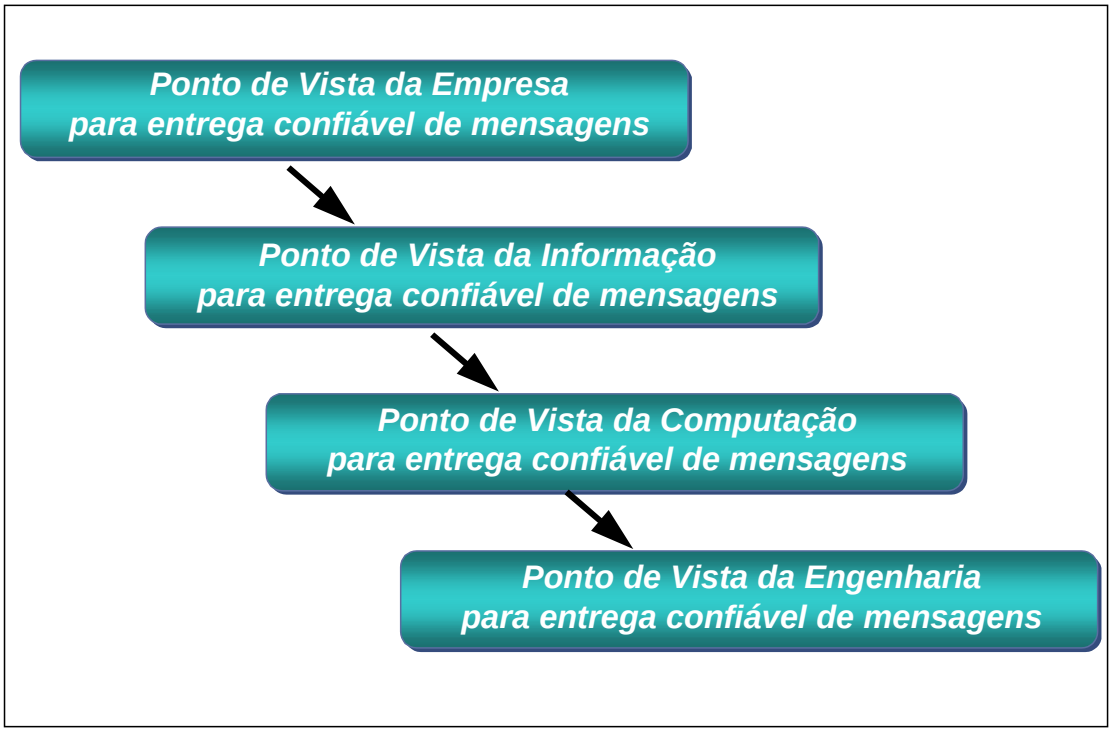


Figura 3.3. Níveis de abstração: os pontos de vista do Modelo de Referência.

Aspectos de *Web services* alheios à troca confiável de mensagens (segurança, processos e aplicações de negócios, gerenciamento de *Web services* e resolução de endereços) não são contemplados no presente Modelo de Referência. Eles podem ser associados, mas não constituem, por si só, um padrão de mensagem confiável em *Web services* (W3C, 2004a).

3.3.1 O ponto de vista da empresa

Este ponto de vista representa a definição dos requisitos básicos na visão empresarial com enfoque em interoperabilidade, visando à entrega confiável de mensagens. A figura 3.4 ilustra os requisitos da empresa, consolidados em notação de diagramas de pacotes UML, obtidos a partir das especificações de modelo orientado a mensagem da WSA (W3C, 2004a, 2004b, 2004c, 2004d, 2004e).

Na relação entre os elementos desse ponto de vista, o escopo define os objetivos de mensagem confiável, com uma visão de empresa. Estes objetivos, por sua vez, são viabilizados por comunidades e/ou federações. As comunidades e federações são compostas por grupos de objetos que constituem os componentes *Web services*, com propósito comum de realizar entrega confiável de mensagens. Estes componentes, chamados de componentes de interoperabilidade, assumem funções empresariais específicas, que, por sua vez, realizam procedimentos distintos. Todas estas funções e procedimentos são regidos pelos contratos e políticas. Os requisitos de empresa são descritos a seguir.

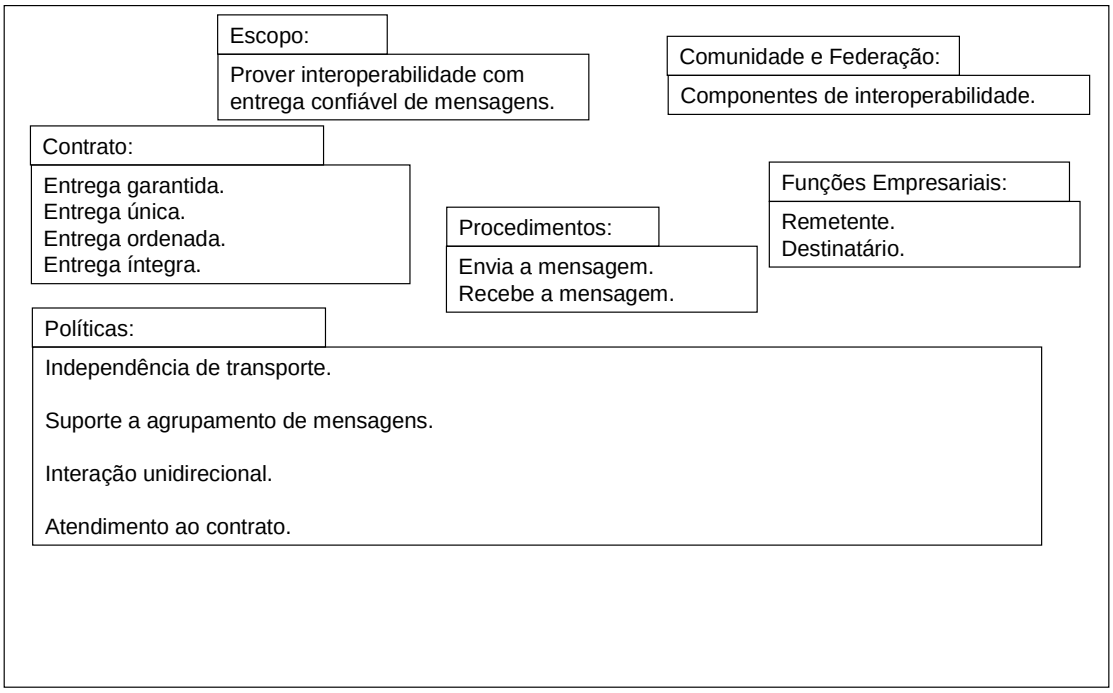


Figura 3.4. Diagrama UML de pacotes para os requisitos do ponto de vista da empresa.

1) Escopo. Segundo a WSA, “mensagem confiável” tem por característica principal realizar interoperabilidade confiável, proporcionando às respectivas aplicações dos agentes *Web services* a mesma percepção de entrega de uma mensagem (W3C, 2004a).

2) Comunidade e Federação. Estes são constituídos por componentes de interoperabilidade que são responsáveis por realizar interação entre agentes

Web services, trocando mensagens de uma forma mutuamente acordada. Os componentes de interoperabilidade *Web services* são originalmente chamados de *SOAP Processors* (W3C, 2000, 2004c, 2004e).

Os grupos de componentes de interoperabilidade que fazem parte de um mesmo domínio empresarial constituem uma comunidade. Os grupos de componentes de interoperabilidade que fazem parte de domínios empresariais diferentes, constituem uma federação.

3) Funções Empresariais. Estas delimitam o conjunto de funções realizadas que identificam comportamentos desempenhados pelos componentes de interoperabilidade. No presente, um componente de interoperabilidade *Web services* realiza duas funções empresariais básicas: remetente e destinatário (W3C, 2004a).

4) Procedimentos. Como conjunto de ações específicas realizadas pelos componentes de interoperabilidade, tem-se enviar mensagens a componentes de interoperabilidade e receber mensagens de componentes de interoperabilidade.

5) Contratos. Em *Web services*, o conceito de “mensagem confiável” é considerado qualidade de serviço. Ao contrato atribuem-se os seguintes requisitos (W3C, 2004a):

- Entrega garantida: a mensagem enviada pelo remetente será entregue ao destinatário;
- Entrega única: não haverá duplicidade de uma mesma mensagem entregue ao destinatário;
- Entrega ordenada: uma seqüência de mensagens será entregue na ordem em que foi enviada;
- Entrega íntegra: a mensagem enviada será preservada até seu destino.

6) Políticas. As políticas expressam o que é permitido, o que é proibido e o que é obrigatório. Em entrega confiável de mensagem têm-se as seguintes políticas:

- Independência de transporte: o transporte é o mecanismo utilizado para entrega de mensagens. A arquitetura de referência *Web services*, definida pelo W3C, diz-se independente da camada de transporte. Portanto, uma mensagem deve ser transmitida do remetente ao destinatário, com transparência sobre qual meio de transporte será utilizado (ex: FTP, HTTP, SHTTP ou SMTP);
- Suporte a grupamento de mensagens: neste caso, as mensagens devem poder ser associadas ao conceito de grupo de mensagens para fins de vinculação a âmbito de negócio ou até para fins de priorização do tratamento da mensagem no destinatário;
- Interação unidirecional: segundo a WSA, uma interação de troca de mensagens é ponto-a-ponto e unidirecional, envolvendo duas partes: o remetente (que envia a mensagem) e o destinatário (que recebe a mensagem);
- Atendimento ao contrato: o contrato estabelece o acordo entre as partes. Numa interação com entrega confiável de mensagens é almejado que os requisitos de contrato (entrega garantida, entrega única, entrega ordenada e entrega íntegra) sejam atendidos.

O presente ponto de vista especificou os requisitos básicos referentes à entrega confiável de mensagens, numa visão mais abstrata que é o da empresa. A representação desses requisitos, no âmbito geral desse ponto de vista, aplicado à mensagem confiável está representada na figura 3.5.

A interoperabilidade confiável focada em entrega confiável de mensagem constitui o escopo. As comunidades e federações são compostas pelos componentes de interoperabilidade que têm o propósito comum de realizar troca de mensagens de uma forma mutuamente acordada. Os componentes de interoperabilidade assumem funções empresariais específicas, que podem ser de remetente ou destinatário, e estas realizam procedimentos de envio e recepção de mensagens através de interfaces. As interações inerentes às funções empresariais e seus procedimentos são restritas aos contratos e políticas específicas para entrega confiável de mensagens.

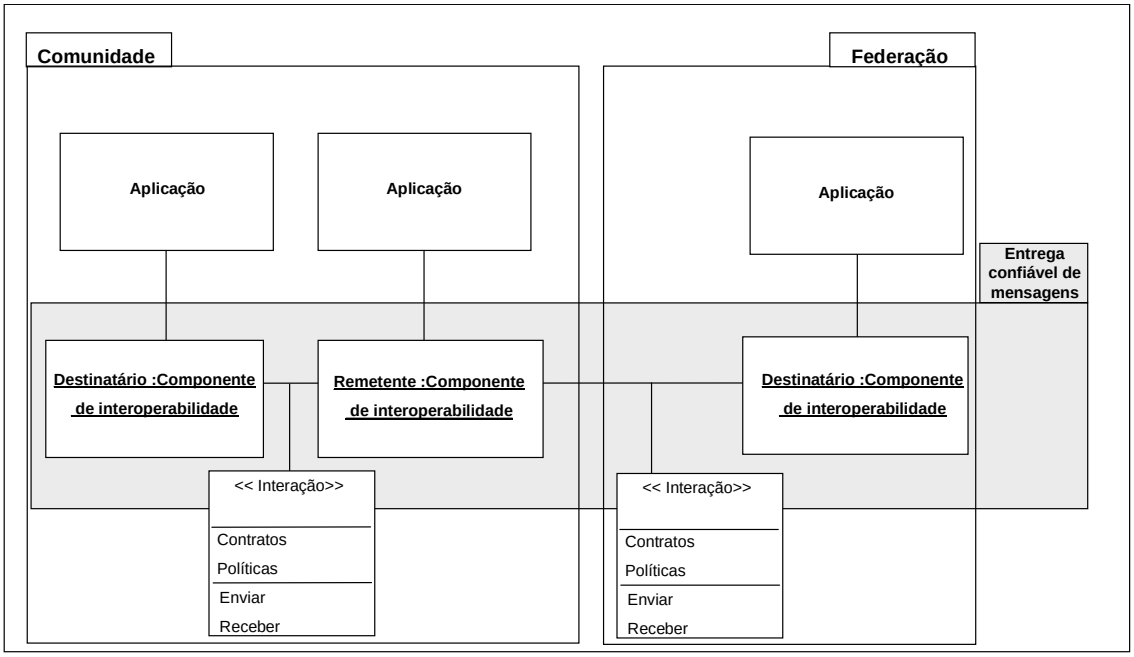


Figura 3.5. Diagrama de classes do ponto de vista da empresa para entrega confiável de mensagem.

3.3.2 O ponto de vista da informação

Este ponto de vista foca a semântica da informação, no Modelo de Referência para entrega confiável de mensagens, aqui proposto. A descrição deste ponto de vista é endereçada por um conjunto de três esquemas: esquema invariante, esquema estático e esquema dinâmico.

3.3.2.1 O esquema invariante

O esquema invariante do presente Modelo de Referência especifica as condições aplicadas à mensagem que serão sempre verdadeiras, na interação entre os componentes de interoperabilidade *Web services*. Para esse esquema, destacam-se os requisitos:

1) A interação é baseada em mensagens. Uma mensagem (também chamada de mensagem SOAP) é a unidade básica de informação para comunicação entre *Web services* (W3C, 2004a, 2004f).

No esquema invariante do ponto de vista da informação, para o presente Modelo de Referência, tem-se que a interação entre *Web services* é feita por meio de mensagens. A estrutura de uma mensagem é composta por duas sub-partes principais, representadas na figura 3.6: um cabeçalho e um corpo.

O corpo de uma mensagem comporta o conteúdo específico de aplicação a ser entregue ao destinatário. O cabeçalho contém os parâmetros que representam informações sobre o conteúdo e sobre a interação (constantes, variáveis ou expressões usadas para passagem de valores entre componentes), como, por exemplo, o número de seqüência da mensagem.

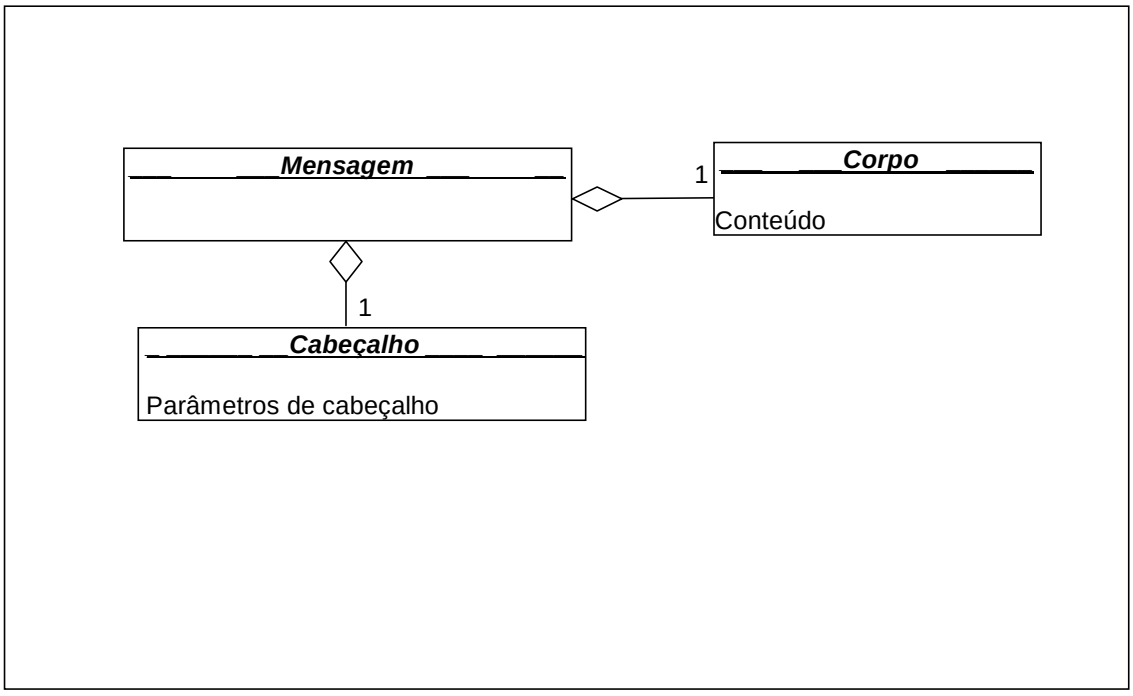


Figura 3.6. Diagrama UML de classes de estruturas agregadas de uma mensagem.

2) Mensagens possuem integridade. A propriedade de validade da mensagem é um dos requisitos a serem atendidos pela implementação de mecanismos de entrega confiável de mensagem. A própria WSA conceitua validade de mensagem como “A mensagem recebida foi a mesma que foi enviada?” (W3C, 2004a, 2004b, 2004c).

O conceito de validade de mensagem está relacionado à integridade da mensagem. Ou seja, se a mensagem recebida pelo destinatário não for a mesma que foi enviada pelo remetente, deve ser possível de se detectar esse erro. Para tal, o Modelo de Referência aponta a necessidade da utilização de parâmetros de controle de integridade no cabeçalho da mensagem.

3) Mensagens possuem numeração de seqüência. Uma mensagem deve ser entregue na ordem em que foi enviada. A inclusão de números de seqüências em mensagens possibilitará ao destinatário, ao receber uma mensagem, detectar se ela está em seqüência (W3C, 2004a, 2004c, 2004d).

   O esquema invariante do ponto de vista da informação, para o Modelo de Referência, considera a presença de parâmetro com numeração de seqüência de mensagem no cabeçalho da mesma.

4) Mensagens possuem identificação única. A comunicação orientada a mensagem em *Web services* deve utilizar identificação única de mensagem para reconhecimento e detecção mais eficaz de duplicidade, provendo uma identidade única para cada mensagem enviada (W3C, 2004a; TAI; MIKALSEN; ROUVELLOU, 2003).

   Posto isso, o esquema invariante do ponto de vista da informação, no presente Modelo de Referência, considera a presença de identificador único de mensagem no cabeçalho dessa.

5) Mensagens possuem identificação de grupo. O esquema invariante do ponto de vista da informação, para o Modelo de Referência, considera o uso de parâmetro no cabeçalho da mensagem, visando a identificar o grupo de mensagem ao qual ela pertence. A necessidade de que as mensagens sejam agrupadas é permitir que possam ser vinculadas a algum aspecto de negócios (lLOWELL; CHEN, 1998; TAI; MIKALSEN; ROUVELLOU, 2003; W3C, 2004c, 2004e).

6) Mensagens possuem marcas de tempo (*timestamps*). Essas marcas associam tempo às mensagens e são úteis na área de segurança para a detecção de alguns tipos de ataques, bem como um dos requisitos para implementar

transações utilizando modelos orientados a mensagens (ERRADI; MAHESHWARI, 2005; TAI; MIKALSEN; ROUVELLOU, 2003; W3C, 2004a, 2004c; 2004e).

O esquema invariante do ponto de vista da informação, para o Modelo de Referência, aponta necessidade de parâmetro de marca de tempo no cabeçalho da mensagem. A associação de marcas de tempo às mensagens permite, no presente Modelo de Referência, não só estabelecer a ordem cronológica destas dentro de um grupo de mensagens, mas também identificar mensagens que estejam expiradas. Além disso, a presença de marca de tempo em mensagens permite ao destinatário detectar e desconsiderar a iniciação de uma conexão implícita ao receber uma mensagem já expirada.

3.3.2.2 O esquema estático

Este esquema define os estados possíveis da mensagem. Uma mensagem confiável pode assumir dois estados: mensagem de interação ou mensagem de reconhecimento. Na figura 3.7 ilustram-se os elementos que compõem uma mensagem *Web services* e seus elementos básicos: o cabeçalho e o corpo da mensagem.

A mensagem de interação é aquela enviada do remetente ao destinatário e que transporta conteúdo específico de aplicação. A mensagem de reconhecimento é aquela pela qual o destinatário informa ao remetente que uma determinada mensagem foi recebida e aceita.

O corpo de uma mensagem, como exposto anteriormente, comporta conteúdo específico de aplicação. O cabeçalho contém os parâmetros que representam informações sobre o conteúdo e sobre a interação, como, por exemplo, o grupo ao qual a mensagem pertence.

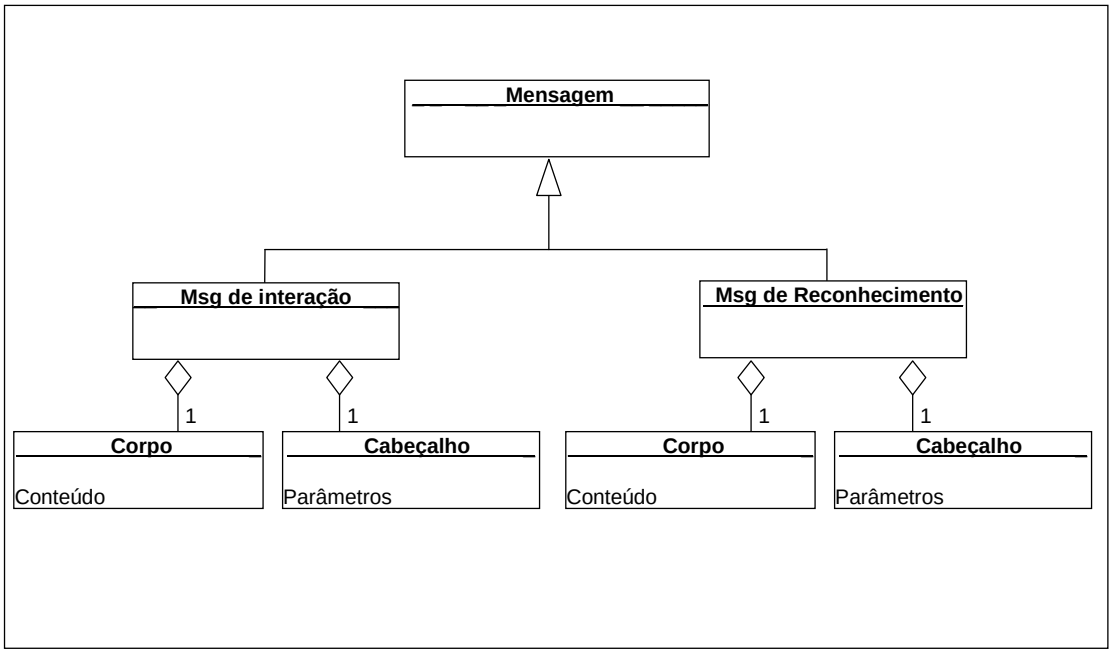


Figura 3.7. Diagrama UML de classes dos elementos agregados e generalizados de uma mensagem.

3.3.2.3 O esquema dinâmico

Este esquema define todas as ações sobre a mensagem que implicam em mudanças de estados na mesma. A figura 3.8 apresenta o diagrama de atividades UML, referente ao esquema dinâmico de informação, para o presente Modelo de Referência.

Neste representam-se as atividades e o fluxo de mensagens (de conteúdo e de reconhecimentos) delineados entre duas partes: remetente e destinatário. O componente de interoperabilidade remetente inicia interação ao receber pedido de envio de mensagem da camada superior de seu agente. A camada superior refere-se a um componente de aplicação *Web services*, que está fora do presente escopo, mas que utiliza o componente de interoperabilidade remetente para o envio de mensagem a outro agente *Web services*.

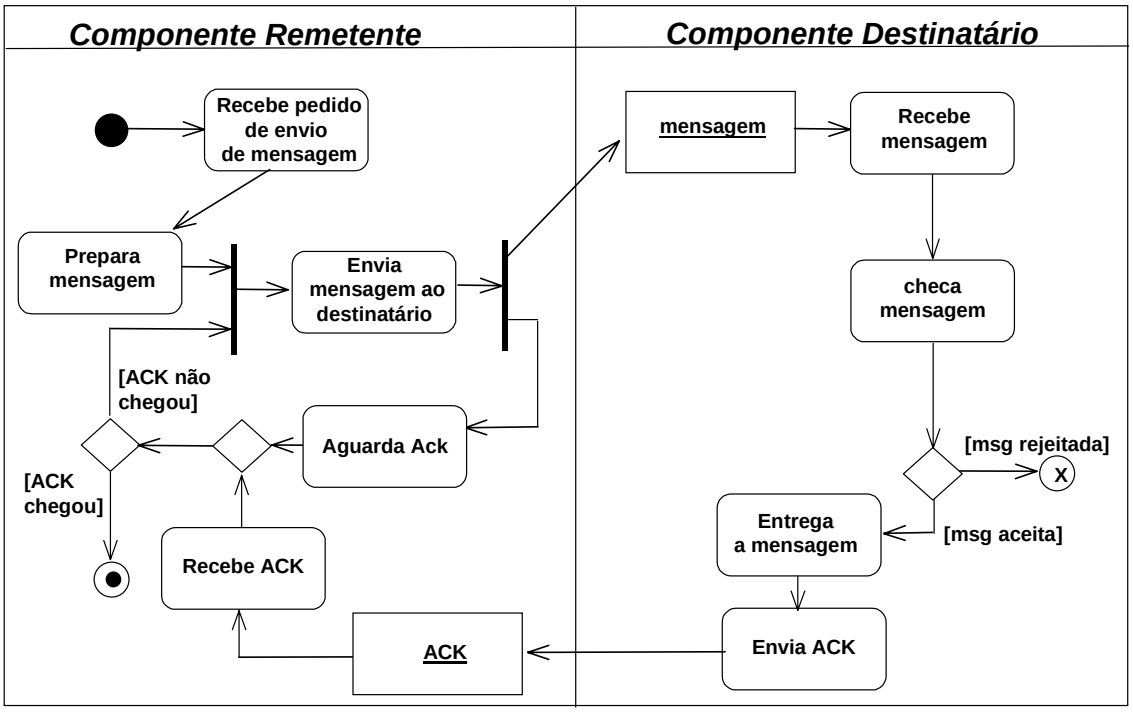


Figura 3.8. Diagrama UML de atividades do esquema dinâmico.

Ao receber esse pedido, juntamente com a mensagem a ser transmitida e o endereço destinatário[9], o componente remetente precisa adequá-la, antes de enviá-la. Esta atividade, aqui denominada “prepara mensagem”, adiciona parâmetros de controle no cabeçalho, tais como número de seqüência de mensagem, código de detecção de erros, identificação única de mensagem, identificação de grupo de mensagem e marca de tempo. Após isso, o componente remetente envia a mensagem ao destinatário e passa à atividade de espera por um reconhecimento (ACK) vindo do destinatário.

Quando o componente destinatário recebe uma mensagem, ele realiza atividades de checagem da mensagem antes de entregá-la às camadas superiores de seu agente *Web services*. Nessa checagem, verifica-se quanto à mensagem recebida estar íntegra, estar em seqüência, não estar expirada, não ser repetida e pertencer a um grupo válido de mensagens.

[9] O processo de resolução de endereços não faz parte do escopo deste trabalho. Essa atividade é realizada pelo “*discovery service*”, que é independente do componente de interoperabilidade (W3C, 2004a).

No caso de alguma irregularidade com relação a essas atividades, a mensagem poderá ser descartada ou aceita e entregue à camada superior (aplicação no agente *Web services* destinatário), sendo que um reconhecimento positivo (ACK) é enviado ao componente remetente. Enquanto isso, o componente remetente, que ficou no aguardo de um reconhecimento, pode disparar uma retransmissão da mesma mensagem, caso o respectivo reconhecimento não chegue dentro de um intervalo de tempo específico.

Em geral, há três possibilidades de causa para o não recebimento de um reconhecimento pelo remetente: a mensagem enviada não chegou ao destinatário; a mensagem enviada chegou ao destinatário, mas foi rejeitada; a mensagem chegou ao destinatário, foi aceita e esse enviou um reconhecimento positivo (que não chegou ao remetente). Quando um reconhecimento chega ao remetente, confirma-se que a mensagem enviada foi recebida e aceita pelo destinatário. Dessa forma, encerra-se o ciclo de entrega confiável de uma mensagem.

Definidos os aspectos pertinentes à semântica da mensagem, passa-se à definição de aspectos de interação, interfaces e conexões entre componentes de interoperabilidade, que realizam a troca de mensagens.

3.3.3 O ponto de vista da computação

A especificação do ponto de vista da computação do presente Modelo de Referência, define como os componentes de interoperabilidade *Web services* interagem. Este ponto de vista foca nesses componentes e suas ligações (em termos de interações, interfaces e conexões), conforme representado na figura 3.9.

A interação com mensagem confiável em *Web services* ocorre entre componentes de interoperabilidade. A interface, por sua vez, determina o comportamento da interação entre componentes de interoperabilidade. A conexão (*binding*) representa a ligação dinâmica entre componentes de interoperabilidade e habilita as capacidades de interações referentes a um tipo de interface.

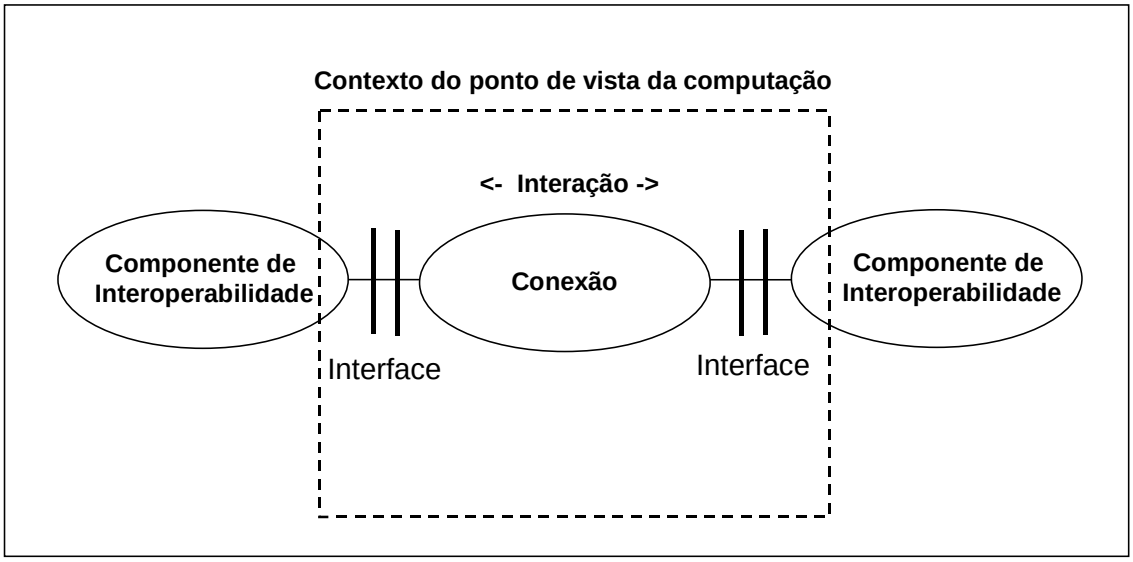


Figura 3.9. Diagrama de contexto do ponto de vista da computação para entrega confiável de mensagens.

3.3.3.1 Tipo de interface

O RM-ODP define, originalmente, três tipos de interface possíveis (operação, fluxo e sinal) para suportar interações. Por outro lado, os *Web services* interagem trocando mensagens segundo MEPs (*Message Exchange Patterns*) suportados na especificação WSA (W3C, 2004c, 2004e).

O MEP tipo solicitação/resposta é tido como a interação fundamental entre *Web services*. Este modo de interação se destaca por viabilizar uma interação cliente-servidor. Neste caso, um solicitador de serviço (“cliente” remetente) envia uma mensagem de solicitação ao provedor de serviço (“servidor” destinatário), sendo que este último retorna alguma resposta (a informação solicitada ou confirmação da ocorrência de alguma ação).

Posto isto, tem-se que o tipo de interface RM-ODP mais adequado a *Web services*, e adotado no presente Modelo de Referência, é a interface do tipo operação. A interface do tipo operação suporta interações entre um objeto cliente e um objeto servidor e representam procedimentos de solicitação ou anúncio (ISO, 1996b, 1998a, PUTMAN, 2001).

3.3.3.2 Modos de interação

Interações confiáveis entre *Web services* necessitam de mecanismos de reconhecimento. Reconhecimento de mensagens em *Web services* deve ser positivo, sendo opcional e complementar o uso de reconhecimento negativo (ERRADI; MAHESHWARI, 2005; TAI; MIKALSEN; ROUVELLOU, 2003; W3C, 2004a).

As interações com reconhecimento implementam esquemas de controle de erros de transmissão - chamados ARQ (*Automatic Response Request*), amplamente utilizados por protocolos ditos de entrega confiável de dados. Portanto, o presente Modelo de Referência propõe, como modelo de interação, o suporte a um dos modos de interação baseados em ARQ, aqui denominados como modo básico, *go-back-n* e repetição seletiva.

1) Modo básico de interação. Este modo de interação, ilustrado na figura 3.10, corresponde ao esquema de controle de erros de transmissão, que implementa interação entre partes do tipo pare-e-espere. Neste modo, para cada mensagem enviada pelo componente de interoperabilidade remetente, haverá espera por um reconhecimento vindo do destinatário. No caso, se o remetente tiver mais de uma mensagem a enviar usando o presente modo de interação, este processo será uma cadência exata de “envio de mensagem”, “aguardo de reconhecimento” e “recepção reconhecimento”, do lado remetente.

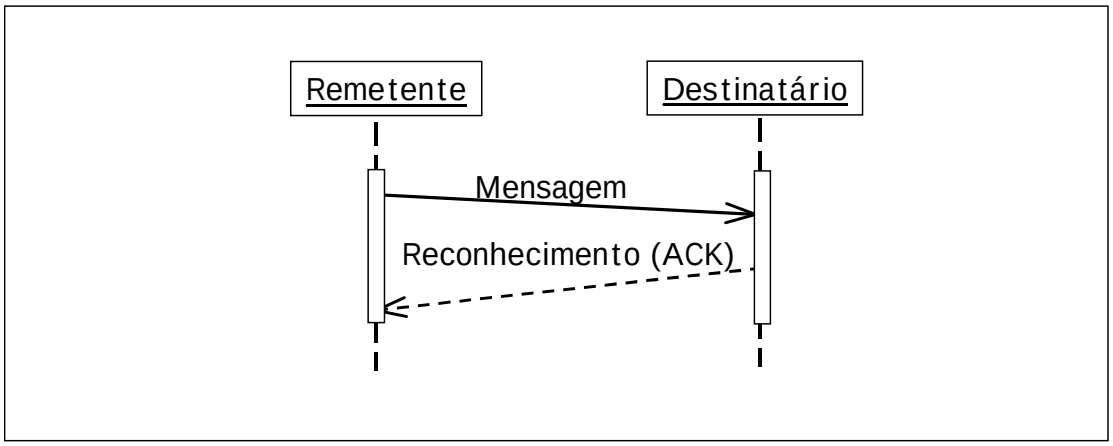


Figura 3.10. Diagrama UML de seqüência para o modo básico de interação.

2) Modo de interação *go-back-n*. Este modo é mais eficiente que o modo básico de interação e ilustrado na figura 3.11.a. Nele, o remetente pode enviar múltiplas mensagens, dentro de uma janela pré-definida, sem esperar por um reconhecimento a cada envio. A janela limita o remetente a não ter mais do que um número J de mensagens não reconhecidas (no exemplo da figura, J=4).

   Ao receber uma mensagem, o destinatário envia um reconhecimento contendo o número de seqüência da mensagem reconhecida. O modo *go-back-n* opera com reconhecimentos individuais para cada mensagem, mas também suporta reconhecimento acumulativo.

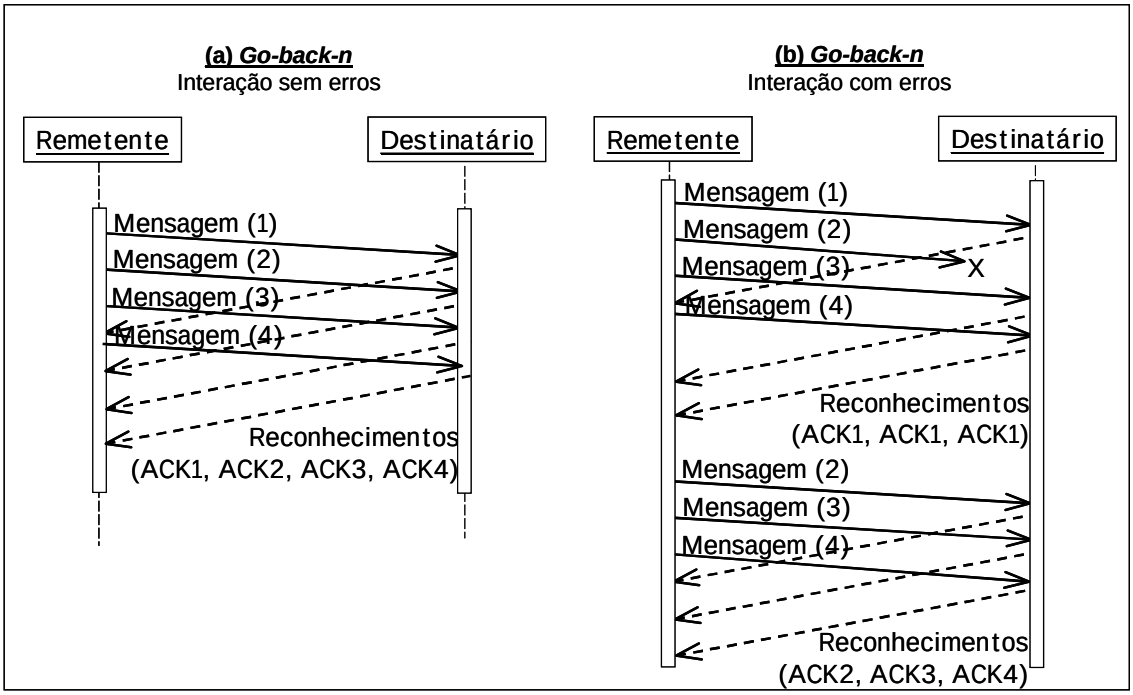


Figura 3.11. Diagrama UML de seqüência para o modo de interação *go-back-n*.

Caso uma das mensagens não chegue ao destinatário ou caso chegue fora de ordem (figura 3.11.b), o destinatário envia reconhecimento referente à última mensagem, de uma seqüência de mensagens recebida corretamente. No exemplo, reconhece-se repetidamente a mensagem número 1, porque o destinatário não recebeu a mensagem número 2. Após a expiração de um temporizador, o remetente retransmite automaticamente a seqüência de

mensagens posterior à última mensagem reconhecida (no caso, as mensagens 2, 3 e 4).

3) Modo de interação por repetição seletiva. Este é um modo de interação mais eficiente que os dois anteriores e ilustrado na figura 3.12. Ele evita retransmissões desnecessárias - tal como no *go-back-n* – por fazer com que o remetente retransmita somente as mensagens que não chegaram ao destinatário.

   Neste modo de interação, a seqüência de transmissão (figura 3.12.a) é similar à do modo *go-back-n*. Ou seja, é permitido que o remetente envie uma quantidade J (chamada de janela de transmissão) de mensagens sem precisar parar e esperar por reconhecimento antes de cada transmissão.

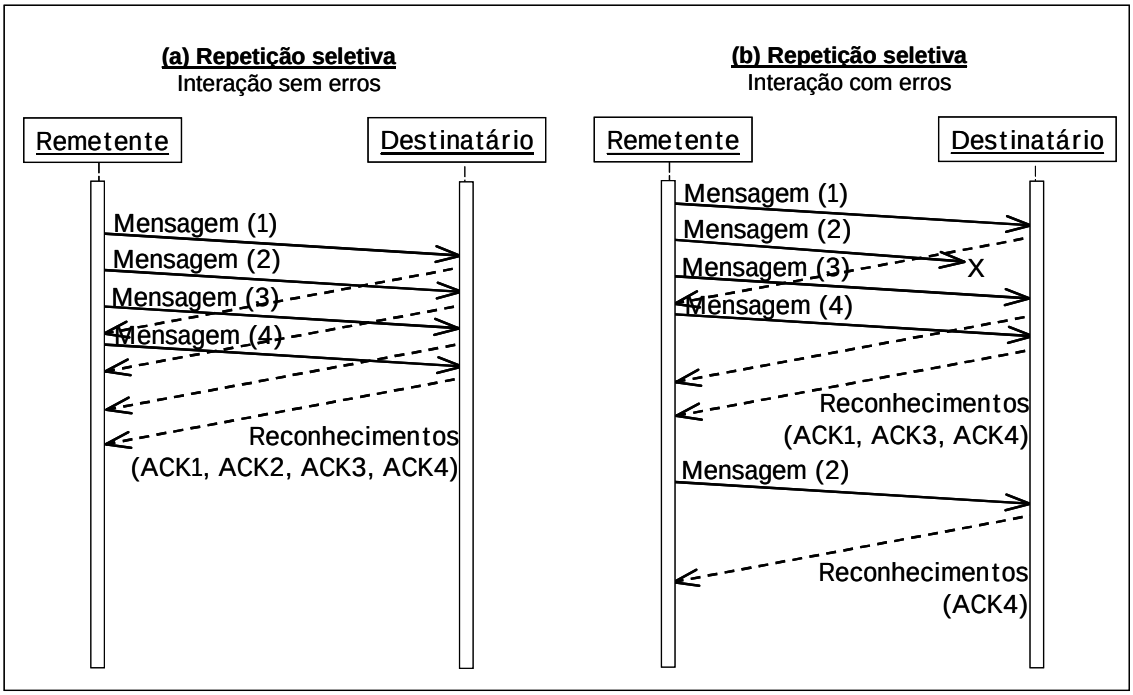


Figura 3.12. Diagrama UML de seqüência para o modo de interação por repetição seletiva.

No processo de recepção, o modo de repetição seletiva diferencia-se do modo de interação *go-back-n* por permitir que mensagens cheguem fora de ordem (o destinatário se encarrega de ordenar as mensagens antes de entregá-las à aplicação). Outra característica deste modo, como ilustrado na figura 3.12.b, caso uma das mensagens não chegue ao destinatário (no exemplo, a mensagem 2), o

mesmo envia reconhecimentos referentes a cada uma das demais mensagens recebidas corretamente (dentro de uma janela de tamanho J), exceto o reconhecimento da mensagem que não chegou ou que foi recebida com erro. Isso faz com que o remetente retransmita automaticamente, após a expiração de um termporizador, a respectiva mensagem não reconhecida.

3.3.3.3 Modos de conexão e desconexão

Como, segundo o Modelo de Referência, os *Web services* utilizam interfaces do tipo operação, é necessário estabelecimento de conexão. Para tal, o ponto de vista da computação do presente Modelo de Referência adota dois tipos de conexões possíveis: conexão implícita e conexão explícita.

1) Modo de conexão implícita. Esta conexão é estabelecida assim que uma mensagem com conteúdo específico de aplicação é enviada do remetente ao destinatário. Ou seja, o remetente simplesmente inicia o envio de mensagens ao destinatário, dispensando qualquer interação preliminar específica com este para o estabelecimento de conexão. Uma interação, tal como a representada na figura 3.10, pode ser considerada como estabelecimento de conexão implícita.

2) Modo de conexão explícita. A interação entre componentes de interoperabilidade, utilizando conexão explícita no presente Modelo de Referência, inicia-se tal como ilustrado na figura 3.13. Neste caso, há uma interação preliminar, específica para o estabelecimento de conexão, que ocorre antes de se dar inicio às interações de mensagens com conteúdo específico de aplicação. Conexões explícitas podem ter procedimentos de conexão do tipo *2-way handshake* ou *3-way handshake* (IREN; AMER; CONRAD, 1999; KUROSE; ROSS, 2001).

   No primeiro caso, tal como na figura 3.13.a, o componente remetente envia um pedido de conexão ao componente destinatário, que responde com uma confirmação de conexão. No segundo caso, tal como na figura 3.13.b, mais complexo, assim que o remetente receber a confirmação de conexão, ele deve enviar um reconhecimento de confirmação de conexão ao destinatário. Em ambos os procedimentos de conexão entre componentes de interoperabilidade

Web services, não se esperam trocas de conteúdo específico de aplicações nesta fase. Trata-se apenas de interação exclusiva para o estabelecimento de conexão.

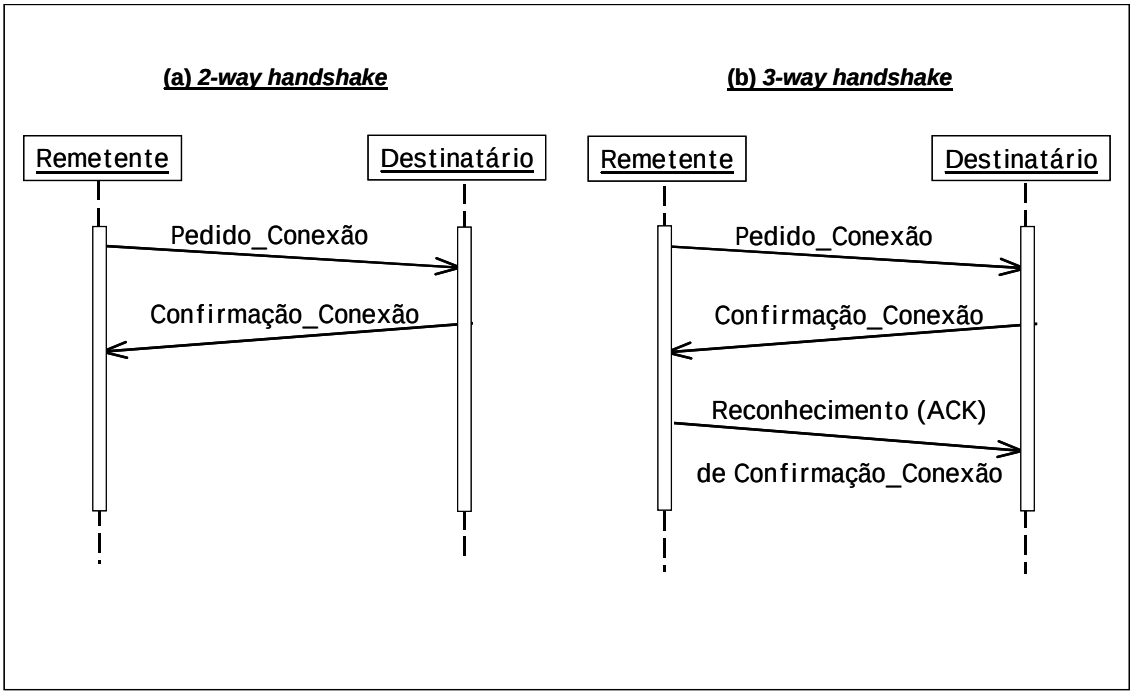


Figura 3.13. Diagrama UML de seqüência para conexão explícita.

Uma vez estabelecida a conexão e realizada a transferência de dados entre dois componentes, o último estágio é o encerramento da conexão. O presente Modelo de Referência suporta os modos de desconexão implícita e explícita, a seguir descritos.

Quando um dos componentes de interoperabilidade não recebe mensagens de seu par por um certo período de tempo, ele simplesmente considera a conexão encerrada. Esse tipo de desconexão é chamado de desconexão implícita e é o utilizado por mecanismos que implementam conexão implícita.

Desconexões explícitas podem adotar procedimentos *2-way handshake* ou *4-way handshake*. Para o caso de uma desconexão em procedimento *2-way-handshake*, ilustrada na figura 3.14.a, o remetente envia um pedido de desconexão ao destinatário e esse responde com uma confirmação da desconexão. Já no caso de uma desconexão em procedimento *4-way-handshake*, ilustrada na figura 3.14.b,

ocorrem duas seqüências de desconexão *2-way-handshake*. A primeira inicia do remetente para o destinatário. A segunda, do destinatário para o remetente.

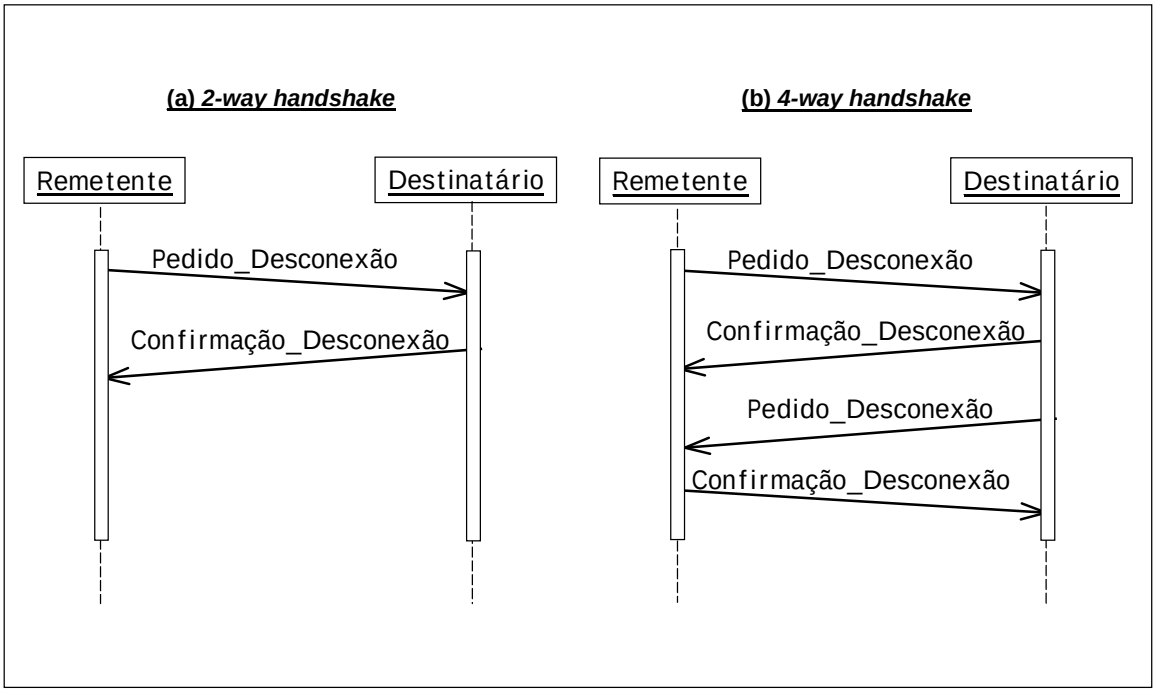


Figura 3.14. Diagrama UML de seqüência para desconexão explícita.

Com o exposto, completa-se a especificação do ponto de vista da computação do presente Modelo de Referência onde são descritos aspectos de interação, tipo de interface e modos de conexão e desconexão aplicados à interoperabilidade com entrega confiável de mensagens. Em seguida, passa-se à definição da infra-estrutura que suporta esses aspectos.

3.3.4 O ponto de vista da engenharia

A linguagem deste ponto de vista foca na descrição dos recursos necessários para os aspectos definidos no ponto de vista da computação. Aqui tem-se a descrição da infra-estrutura necessária para suportar interações entre os componentes de interoperabilidade.

A especificação do ponto de vista da engenharia no Modelo de Referência para entrega confiável de mensagens é centrada na estrutura de intercomunicação, esta que está baseada nos objetos de canal.

Uma vez que a WSA não define conversões e/ou troca do padrão de interoperabilidade entre *Web services*, o Modelo de Referência desconsidera o uso de objetos interceptadores no canal. Portanto, o ponto de vista da engenharia foca em definir o modelo de canal, formado pelos objetos: adaptador, conector e protocolo.

3.3.4.1 Objetos adaptadores

No presente Modelo de Referência, os objetos adaptadores são responsáveis pela adaptação do cabeçalho das mensagens para comportar os parâmetros necessários ao controle de erros: número de seqüência, identificador de mensagem, identificador de grupo de mensagem e código de detecção de erro.

3.3.4.2 Objetos conectores

Um objeto conector estabelece a conexão quando o canal é criado entre dois objetos básicos e o procedimento de estabelecimento de conexão depende diretamente do tipo de interface. Como o tipo de interface do Modelo de Referência é o de operação, passa-se à descrição dos objetos conectores do canal para cada um dos modos possíveis de conexão e desconexão.

Os objetos conectores estabelecem seqüências distintas de mensagens e assumem estados distintos durante o estabelecimento e o encerramento do canal. Embora o presente Modelo de Referência não intencione descrever a implementação desses objetos, adota-se aqui definir seus estados possíveis, bem como ilustrar, em alguns casos, os respectivos diagramas de MEFs (Máquinas de Estados Finitos).

1) Conexão implícita. Este tipo de conexão é estabelecido assim que a primeira mensagem é enviada do remetente ao destinatário. Neste caso, não há procedimentos específicos de estabelecimento de conexão. Os objetos

conectores dos componentes de interoperabilidade podem assumir somente dois estados possíveis: “conectado” ou “desconectado”.

2) Conexão explícita *2-way-handshake*. Neste caso, ocorrem trocas de mensagens específicas para o estabelecimento de conexão, cuja seqüência foi ilustrada na figura 3.13.a. Seus objetos conectores passam a suportar mais estados possíveis que em um estabelecimento de conexão implícita. No caso, os objetos conectores podem assumir até três estados: “desconectado”, “aguardando confirmação de conexão” ou “conectado”.

3) Conexão explícita 3-*way-handshake*. Neste caso, também ocorrem trocas de mensagens específicas para o estabelecimento de conexão, cuja seqüência foi ilustrada na figura 3.13.b. Para tal, os objetos conectores podem assumir até quatro estados: “desconectado”, “aguarda confirmação de conexão”, “aguarda ACK de confirmação de conexão” ou “conectado”. A representação da MEF de conexão explícita 3-*way-handshake*, do Modelo de Referência, para o objeto conector remetente e destinatário *Web services*, está disposta na figura 3.15.

   Nessa figura, o conector remetente é inicializado no estado de “desconectado” ao receber uma solicitação de envio de mensagem (vinda da aplicação, através do objeto adaptador). Em havendo mensagem a enviar, este conector remetente deve, primeiramente, estabelecer conexão com o conector destinatário. Sendo assim, o conector remetente envia um pedido de conexão ao conector destinatário (evento 1). O conector remetente entra em estado de “aguarda confirmação de conexão”, que deve vir do conector destinatário. Ao chegar essa confirmação (evento 3), o conector remetente envia um reconhecimento (ACK) de confirmação de conexão e entra em estado de “conectado”. Além disso, ao entrar em estado de “aguada confirmação de conexão”, inicia-se um temporizador. Caso a respectiva confirmação de conexão não chegue a tempo, o conector remetente volta ao estado de “desconectado” (evento 5).

   Por outro lado, o conector destinatário muda do estado “desconectado” assim que recebe um pedido de conexão proveniente de um conector remetente (evento 2). Este evento dispara o envio de uma confirmação de conexão ao

conector remetente, sendo que o conector destinatário entra em estado “aguarda ACK de confirmação de conexão”.

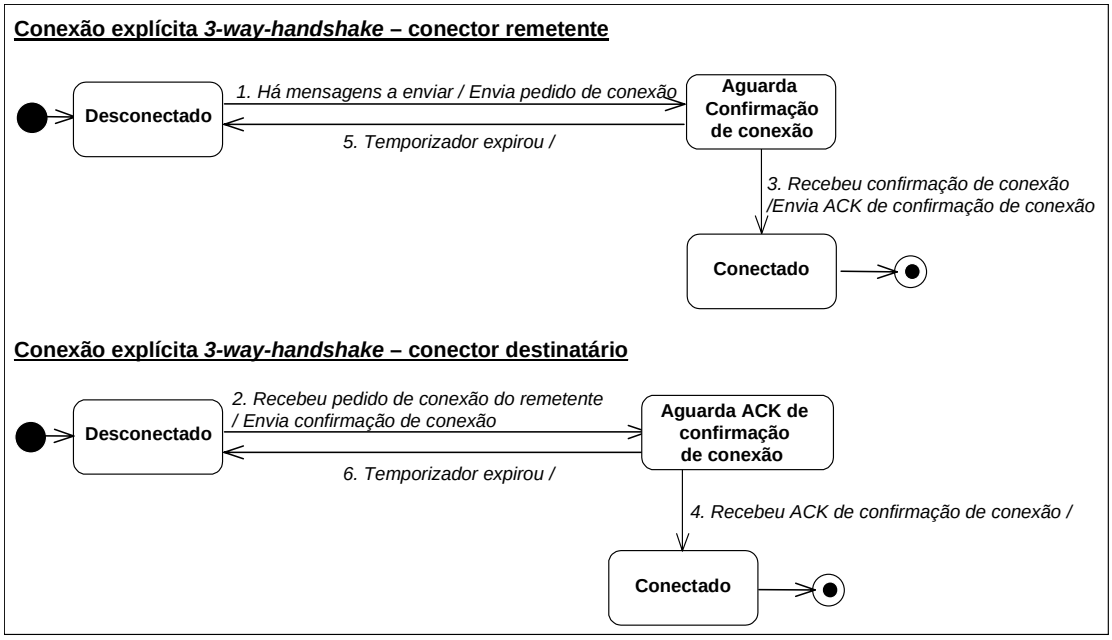


Figura 3.15. Diagrama UML de MEF para conexão explícita.

Ao entrar neste estado, o conector destinatário fica em espera até que, ou o ACK de confirmação de conexão, vinda do conector remetente chegue ao conector destinatário (evento 4), ou após intervalo de tempo o conector destinatário volte ao estado de “desconectado” (evento 6).

4) Desconexão implícita. Neste caso, estando conectado e não tendo mais mensagens a enviar, o conector remetente simplesmente muda para o estado de “desconectado”. Por outro lado, o conector destinatário estando conectado e não recebendo mais nenhuma mensagem dentro de um intervalo de tempo, controlado por um temporizador, este volta ao estado de “desconectado”.

5) Desconexão explícita *2-way-handshake*. Neste caso, ocorrem trocas de mensagens específicas para o encerramento de conexão entre os objetos conectores. A seqüência para esse processo é idêntica à ilustrada na figura

3.14.a. Seus objetos conectores suportam até três estados possíveis: “conectado”, “aguarda confirmação de desconexão” ou “desconectado”.

6) Desconexão explícita *4-way-handshake*. Neste caso, também ocorrem trocas de mensagens específicas para o encerramento de conexão entre os objetos conectores. A seqüência para esse processo é idêntica à ilustrada na figura 3.14.b. Seus objetos conectores suportam até quatro estados possíveis: “conectado”, “aguarda confirmação de desconexão”, “aguarda pedido de desconexão” ou “desconectado”.

3.3.4.3 Objetos protocolos

Os objetos de protocolo são responsáveis por prover comunicação confiável entre os objetos de conectores a quem prestam serviço. O contexto de protocolos está vinculado às interações. Sendo assim, estes objetos devem implementar um dos três modos de interação, definidos no ponto de vista da computação do Modelo de Referência: o modo básico, o modo *go-back-n* ou o modo de repetição seletiva.

Os objetos protocolos estabelecem seqüências distintas de mensagens e assumem estados distintos durante a interação. Embora o presente Modelo de Referência não intencione descrever a implementação desses objetos, adota-se descrever seus estados possíveis, bem como ilustrar, em alguns casos, os respectivos diagramas de MEFs (Máquinas de Estados Finitos).

1) Objeto protocolo do modo básico de interação. Neste modo de interação, o objeto protocolo realiza interações, tal como as ilustradas na figura 3.10. Para tal, ele pode assumir três estados distintos: “pronto para enviar”, “aguarda reconhecimento de mensagem” ou “pronto para receber”

   Como visto, o modo de interação básica é composto pelo envio de uma mensagem e recepção de seu respectivo reconhecimento. Para melhor ilustração, representam-se as MEFs do modo básico de interação, do Modelo de Referência, para os objetos protocolos remetente e destinatário, na figura 3.16.

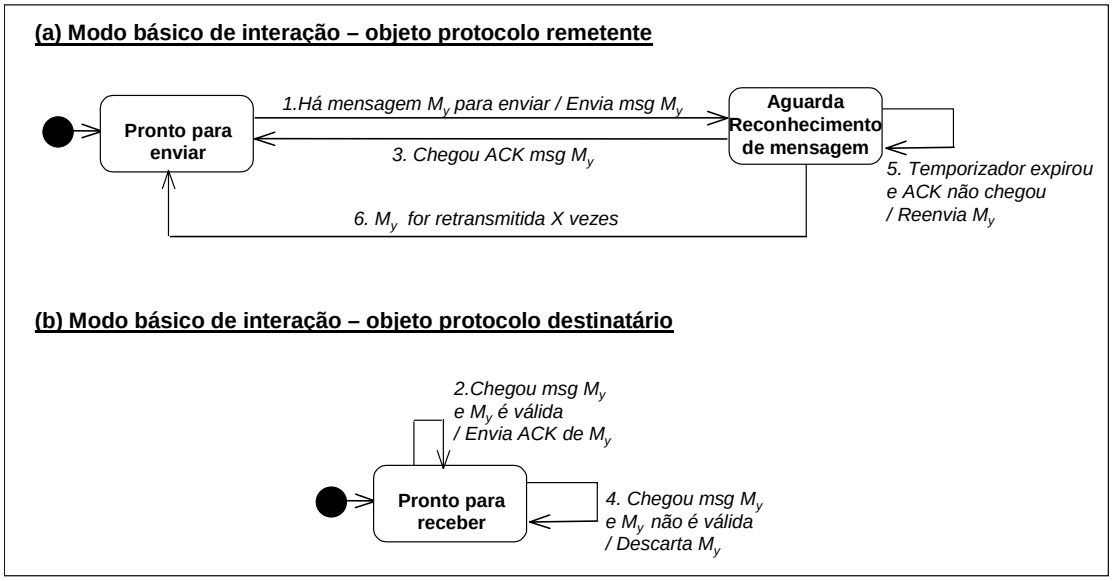


Figura 3.16. Diagrama UML de MEF para o modo básico de interação.

O objeto protocolo remetente é inicializado ao estado de “pronto para enviar”, após o estabelecimento de conexão realizado pelo seu objeto conector. Em havendo uma mensagem M_y a enviar, o objeto protocolo remetente a transmite (evento 1), passando ao estado de “aguarda reconhecimento de mensagem” para M_y. Se o reconhecimento da mensagem chegar (evento 3), o objeto protocolo remetente volta novamente ao estado “pronto para enviar”. Caso contrário, o objeto protocolo remetente fica em espera, reenviando a mensagem M_y X vezes, (evento 5) controlados por um temporizador. Após X retransmissões da mesma mensagem, caso não haja recebimento do reconhecimento desta, o objeto remetente aborta a comunicação - pois, provavelmente, o objeto destinatário está indisponível – e volta ao estado de “pronto para enviar”.

Na figura 3.16.b, que representa a MEF do objeto destinatário para o modo básico de interação, tem-se que esse objeto é inicializado ao estado de “pronto para receber”, após o estabelecimento de conexão realizado entre os objetos conectores. Uma vez recebida uma mensagem, verifica-se se esta é válida (ex: não expirada, em seqüência e íntegra). Caso positivo, envia-se um reconhecimento dessa mensagem, ao remetente (evento 2). Caso negativo, descarta-se a mensagem (evento 4).

2) Objeto protocolo do modo de interação *go-back-n*. Neste modo de interação, o objeto protocolo realiza as interações ilustradas na figura 3.11. Seus estados possíveis são idênticos aos dos objetos protocolos do modo básico de interação, apenas com a característica de que o objeto protocolo remetente é autorizado a transmitir múltiplas mensagens (se disponíveis) sem esperar por um reconhecimento, quando está no estado “pronto para enviar”. Um objeto protocolo destinatário, neste modo de interação, que recebe múltiplas mensagens, as reconhece, desde que estejam válidas e em ordem. Mensagens fora de ordem são descartadas.

3) Objeto protocolo do modo de interação repetição seletiva. Neste modo de interação, o objeto protocolo realiza as interações ilustradas na figura 3.12. Seus estados possíveis são idênticos aos dos objetos protocolos do modo básico de interação e *go-back-n*. Tal como no modo *go-back-n*, neste modo o objeto protocolo remetente é autorizado a transmitir múltiplas mensagens (se disponíveis) sem esperar por um reconhecimento, quando está no estado “pronto para enviar”. Um objeto protocolo destinatário, neste modo de interação, que recebe múltiplas mensagens as reconhece, desde que estejam válidas. Neste modo, mensagens recebidas fora de ordem não são descartadas.

Os três modos de interação realizados pelos objetos protocolos implementam controle de erro de transmissão que pressupõem retransmissão. Como visto, quando o objeto protocolo remetente envia uma mensagem a um destinatário, ele deve acionar um temporizador. Se o temporizador expirar antes de o remetente receber o reconhecimento respectivo à mensagem enviada, este retransmitirá a mensagem, pois é possível que uma das três condições tenha ocorrido: a mensagem enviada não chegou ao destinatário; a mensagem enviada foi descartada pelo destinatário; o reconhecimento, vindo do destinatário, foi perdido. Para tal, pressupõe-se que o objeto protocolo remetente adote um algoritmo de RTO (*Retransmition TimeOut*), com base em um padrão de medição de RTT (*Round Trip Time*) de mensagem a fim de estabelecer o intervalo de tempo a ser adotado pelo temporizador. Resalta-se que a WSA aponta a questão da necessidade de controle de *round trip*, embora não

estabeleça um padrão de algoritmo específico para usar em *Web services* (W3C, 2004c).

Os itens anteriores definiram todo o Modelo de Referência, elemento do Método de Verificação. A última etapa do presente capítulo apresenta o elemento Matriz de Mapeamento de Requisitos.

3.4 Matriz de Mapeamento de Requisitos

Neste item, apresenta-se a estrutura do elemento Matriz de Mapeamento de Requisitos que faz parte do Método de Verificação proposto.

A Matriz de Mapeamento de Requisitos possui uma estrutura bidimensional, em formato tabular, tal como representado na tabela 3.1, contendo as respectivas colunas:

- Requisito: corresponde ao requisito almejado em entrega confiável de mensagens. O conjunto de todos os requisitos é definido no elemento Modelo de Referência que faz parte do presente Método de Verificação;
- Questionamento: representa o questionamento elaborado a partir do respectivo requisito para entrega confiável de mensagens e visa a direcionar a inspeção sobre como e onde a EPMC indica atender a esse requisito;
- Requisitos encontrados[10]: representa o conjunto dos resultados da revisão sistemática da documentação da EPMC, esta viabilizada pelo Processo de Verificação. Nesta coluna registram-se a referência bibliográfica, a página e o resultado encontrado relativo ao respectivo requisito verificado na EPMC.

Tabela 3.1. Estrutura de Matriz de Mapeamento de Requisitos.

Requisito	Questionamento	Requisitos encontrados

[10] Requisitos encontrados são também chamados de *findings* em verificação de software.

Nas verificações de requisitos para os quais não são encontrados resultados, convenciona-se registrá-los como “não encontrado” na respectiva coluna de requisitos encontrados.

Para os requisitos verificados como não suportados pela EPMC, convenciona-se registrá-los como “não contemplado” na respectiva coluna de requisitos encontrados.

Informações adicionais e pertinentes aos resultados de verificação, que venham a complementar o entendimento aos requisitos encontrados, agregando características específicas do contexto da EPMC verificada, são inseridas à parte, como “observações complementares” a cada Matriz de Mapeamento de Requisitos.

Definidos o Método de Verificação e cada um de seus elementos, passa-se à aplicação deste sobre as EPMCs, no capítulo seguinte.

4 VERIFICAÇÃO E COMPARAÇÃO ENTRE AS EPMCs

Este capítulo apresenta os resultados de verificação, representados pelas Matrizes de Mapeamento de Requisitos para cada uma das duas EPMCs (*WS-Reliability* e *WS-ReliableMessaging*), bem como a comparação qualitativa entre elas.

4.1 Mapeamento de Requisitos da EPMC *WS-Reliability*

Neste item, apresentam-se as Matrizes de Mapeamento de Requisitos referente à verificação da EPMC *WS-Reliability*, segundo o Método de Verificação proposto. Para a verificação desTa EPMC, adota-se como referência técnica a sua documentação de especificação em (OASIS, 2004b).

A cada matriz, informações adicionais e pertinentes à EPMC, para cada requisito verificado, que venham a complementar o entendimento aos requisitos encontrados, são inseridas à parte, como observações complementares.

4.1.1 Matrizes para o ponto de vista da empresa

1) Requisitos de escopo, comunidade, federação, funções empresariais e procedimentos. A Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto a esTes requisitos do ponto de vista da empresa é apresentada na tabela 4.1.

Tabela 4.1. Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao escopo, comunidade, federação, funções empresariais e procedimentos.

Requisito	Questionamento	Requisitos encontrados
Escopo: prover interoperabilidade confiável.	Esta EPMC tem como propósito prover garantia de entrega de mensagem?	**(OASIS, 2004b, p. 1):** "O *WS-Relibaility* é um protocolo baseado em SOAP para a troca de mensagens SOAP com garantia de entrega, não duplicação e garantia de entrega ordenada de mensagens." **(OASIS, 2004b, p. 6):** "O *WS-Relibaility* é uma especificação baseada no SOAP, que atende aos requerimentos de entrega confiável de mensagens, críticos a algumas aplicações de *Web services.*"

(Continua)

(Continuação)

Requisito	Questionamento	Requisitos encontrados
Comunidade e federação: componentes de interoperabilidade.	Segundo esta EPMC, a interação entre agentes é implementada por componentes de interoperabilidade que trocam mensagens entre si?	**(OASIS, 2004b, p.7):** Apresenta-se um diagrama com modelo de troca de mensagens (vide Anexo A do presente trabalho), onde representa-se uma interação ente dois agentes *Web services* (nesta EPMC chamados de *Sending Party* e *Receiving Party*). Esses agentes são compostos por elementos, dentre eles, os componentes de aplicação (*Producer e Consumer*) que, por sua vez, interagem entre si por meio de componentes de interoperabilidade, nessa EPMC chamados de *Sending RMP* e *Receiving RMP*. **(OASIS, 2004b, p. 9):** "O RMP (*Reliable Messaging Processor*) é um processador de SOAP e infra-estrutura capaz de realizar troca confiável de mensagens tal como descrito nesta especificação. Com relação à transmissão de mensagem confiável, de um RMP para outro, o primeiro é chamado de remetente e o segundo de destinatário." **(OASIS, 2004b, p. 15):** "Os RMPs são as únicas partes envolvidas na implementação do protocolo de mensagem confiável."
Componentes de interoperabilidade assumem funções empresariais distintas e realizam respectivos procedimentos.	Esta EPMC delimita as funções empresariais e respectivos procedimentos do componente de interoperabilidade a: 1) Componente remetente: aquele que envia a mensagem de forma confiável; e, 2) Componente destinatário: aquele que recebe a mensagem de forma confiável?	**(OASIS, 2004b, p. 6):** "O RM é um protocolo que inclui cabeçalhos e interações específicas entre uma parte remetente e uma parte destinatária." **(OASIS, 2004b, p. 9):** "O RMP (*Reliable Messaging Processor*) é um processador de SOAP e infra-estrutura capaz de realizar troca confiável de mensagens tal como descrito nesta especificação. Com relação à transmissão de mensagem confiável, de um RMP para outro, o primeiro é chamado de remetente e o segundo de destinatário." **(OASIS, 2004b, p. 11):** "Não há outros papéis para um RMP diferente de RMP remetente ou RMP destinatário."

Observações complementares:

Quanto à comunidade e federação, a EPMC *WS-Reliability*, utiliza a notação "RMP" (*Reliable Messaging Processor*), referindo-se especificamente ao componente de interoperabilidade, que é parte de um agente *Web services*. O RMP é o componente

que implementa a camada de envio e recebimento de mensagens. Ele é parte de um agente *Web services* e presta serviço à outras entidades desse, que são as aplicações *Web services*.

2) Requisitos de contrato. A Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao contrato é apresentada na tabela 4.2.

Tabela 4.2. Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao contrato.

Requisito	Questionamento	Requisitos encontrados
Entrega garantida de mensagem	Esta EPMC provê entrega de mensagem, garantido que uma mensagem enviada seja entregue ao seu destino?	**(OASIS, 2004b, p. 1):** "O *WS-Reliability* é um protocolo baseado em SOAP para a troca de mensagens SOAP com garantia de entrega, não duplicação e garantia de entrega ordenada de mensagens." **(OASIS, 2004b, p. 7):** "A especificação corrente define as seguintes características de confiabilidade: entrega garantida de mensagens, [...]." **(OASIS, 2004b, p. 9):** "Mensagem confiável (RM): é ato de processamento de um conjunto de características SOAP, definidas pelo *WS-Reliability*, que resulta em um protocolo que suporta características de qualidade de serviço tais como entrega garantida, [...]."
Entrega única de mensagem	Esta EPMC provê entrega de mensagem, garantindo a não duplicidade na entrega de uma mensagem ao seu destinatário?	**(OASIS, 2004b, p. 1):** "O *WS-Reliability* é um protocolo baseado em SOAP para a troca de mensagens SOAP com garantia de entrega, não duplicação e garantia de entrega ordenada de mensagens." **(OASIS, 2004b, p. 7):** "A especificação corrente define as seguintes características de confiabilidade: entrega garantida de mensagens, garantia de eliminação de duplicação de mensagens [...]." **(OASIS, 2004b, p. 9):** "Mensagem confiável (RM): é ato de processamento de um conjunto de características SOAP, definidas pelo *WS-Reliability*, que resulta em um protocolo que suporta características de qualidade de serviço tais como entrega garantida, eliminação de duplicação de mensagens [...]."

(Continua)

(Continuação)

Requisito	Questionamento	Requisitos encontrados
Entrega ordenada de mensagens.	Esta EPMC provê entrega de mensagem, garantido que uma seqüência de mensagens seja entregue na ordem em que foi enviada?	**(OASIS, 2004b, p. 1):** "O *WS-Reliability* é um protocolo baseado em SOAP para a troca de mensagens SOAP com garantia de entrega, não duplicação e garantia de entrega ordenada de mensagens." **(OASIS, 2004b, p. 7):** "A especificação corrente define as seguintes características de confiabilidade: entrega garantida de mensagens, garantia de eliminação de duplicação de mensagens e ordenação garantida de mensagens." **(OASIS, 2004b, p. 9):** "Mensagem confiável (RM): é ato de processamento de um conjunto de características SOAP, definidas pelo *WS-Reliability*, que resulta em um protocolo que suporta características de qualidade de serviço tais como entrega garantida, eliminação de duplicação de mensagens e ordenação de mensagens."
Entrega íntegra de mensagem.	Esta EPMC provê entrega de mensagem, garantindo que uma mensagem que chegue a um destinatário seja a mesma que foi enviada pelo remetente?	Não encontrado.

Considerações complementares:

Na EPMC *WS-Reliability*, as propriedades de entrega garantida, entrega única e entrega ordenada são opcionais, ou seja, podem estar habilitadas ou não. Quanto ao requisito de integridade, esta EPMC recomenda fortemente que seja garantida a integridade fim-a-fim, do cabeçalho de mensagem, pelo uso de opções de segurança adequadas que são descritas em outra proposta de padrão, o *WS-Security* (OASIS, 2006d).

Portanto, a EPMC *WS-Reliability* define confiabilidade independentemente de segurança. Este aspecto foi delegado, a outro grupo de estudos de novos padrões em *Web services*.

3) Políticas. A Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto às políticas está representada na tabela 4.3.

Tabela 4.3. Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto às políticas.

Requisito	Questionamento	Requisitos encontrados
Independência de protocolo de transporte.	Esta EPMC visa a implementar entrega confiável de mensagens sem dependência quanto ao protocolo de transporte subjacente utilizado?	**(OASIS, 2004b, p. 1):** "O *WS-Reliability* é definido como extensões de cabeçalho SOAP e é independente de protocolos subjacentes." **(OASIS, 2004b, p. 6):** "O RM é a execução de um protocolo baseado em SOAP e agnóstico quanto ao transporte."
Suporte a agrupamento de mensagens.	Esta EPMC permite que mensagens possam ser associadas a um grupo específico de mensagens?	**(OASIS, 2004b, p. 7):** "A especificação corrente define a seguinte característica de confiabilidade: [...] garantia de ordenação de mensagens para entrega dentro de um grupo de mensagens." **(OASIS, 2004b, p. 15):** "Uma mensagem confiável contém um identificador que é globalmente único e baseado no conceito de um grupo. Uma mensagem sempre pertence a um grupo."
A interação é unidirecional entre componentes.	Esta EPMC tem um modelo de interoperabilidade cujo sentido de envio de mensagens com conteúdo específico de aplicação ocorre do componente remetente para componente destinatário?	**(OASIS, 2004b, p. 9):** "Com relação à transmissão confiável de uma mensagem de um RMP a outro, o primeiro é sempre chamado de remetente e o segundo é sempre chamado de destinatário." **(OASIS, 2004b, p. 11):** Esta EPMC apresenta um diagrama com seu modelo de troca de mensagens (vide Anexos A e B do presente trabalho). Nesse modelo, agentes *Web services*, possuem componentes de interoperabilidade referenciados pelas notações "*Sending RMP*" (parte que envia mensagem) e "*Receiving RMP*" (parte que recebe mensagens). No caso, o componente *Sending RMP* envia mensagem ao componente *Receiving RMP* , que a confirma, retornando um reconhecimento (*RM-Reply*).
Deve-se atender aos requisitos de contrato.	Esta EPMC objetiva atender aos requisitos de contrato (entrega garantida, única, ordenada e íntegra)?	Vide Matriz de Mapeamento dos requisitos quanto ao contrato, apresentada na tabela 4.2

Observações complementares:

Quanto à independência de protoloco de transporte, apesar da EPMC *WS-Reliability* ser independente deste, sua especificação apresenta, adicionalmente e a titulo de suporte a desenvolvedores, exemplos de utilização do método POST[11] para uso do HTTP no transporte de mensagens SOAP estendidas. Trata-se de uma iniciativa pró-ativa do grupo que elaborou a especificação, pelo fato da grande difusão atual do HTTP na Internet.

A EPMC *WS-Reliability* utiliza a notação "RM" (*Reliable Messaging*), referindo-se especificamente ao protocolo implementado por ela. Quanto à interação ser unidirecional, o *RM-Reply* corresponde a um reconhecimento. É a confirmação, do destinatário para o remetente, de que uma mensagem com conteúdo de aplicação fora recebida.

4.1.2 Matrizes para o ponto de vista da informação

4.1.2.1 Quanto ao esquema invariante

Representando condições sempre verdadeiras das mensagens, a Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao esquema invariante é apresentada na tabela 4.4.

Tabela 4.4. Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao esquema invariante.

Requisito	Questionamento	Requisitos encontrados
A interação é baseada em mensagem.	Nesta EPMC, o objeto informação corresponde a uma mensagem, com estrutura baseada na estrutura do padrão SOAP?	**(OASIS, 2004b, p. 1):** "O *WS-Reliability* é um protocolo baseado em SOAP para intercâmbio de mensagens SOAP, com garantias de entrega, não duplicação e garantia de ordenação de mensagem. Ele é definido como extensões de cabeçalho SOAP." **(OASIS, 2004b, p. 6):** "O RM, é um protocolo que inclui cabeçalhos de mensagem específicos e interações específicas de mensagem entre uma entidade remetente e uma entidade destinatária."

(Continua)

[11] POST é um método de envio de informação no HTTP.

(Continuação)

Requisito	Questionamento	Requisitos encontrados
Mensagens possuem integridade.	Nesta EPMC, as mensagens são enviadas contendo parâmetro de controle de integridade no cabeçalho?	Não contemplado.
Mensagens possuem numeração de seqüência.	Nesta EPMC, as mensagens são enviadas com um parâmetro de número de seqüência em seu cabeçalho?	**(OASIS, 2004b, p. 15):** "Um número de seqüência, quando presente, é um inteiro que é único dentro de um grupo." **(OASIS, 2004b, p. 24):** "O RMP remetente deve incluir o elemento <*SequenceNum*> em todas as mensagens confiáveis de um grupo que tenha mais de uma mensagem. O elemento <*SequenceNum*> contém o número de seqüência." **(OASIS, 2004b, p. 26):** "Este atributo <*SequenceNum*> contém o número de seqüência que identifica a mensagem dentro de seu grupo e é utilizado para suportar a ordenação de mensagens [...]. O RMP remetente deve inicializar esse com 0 (zero) para a primeira mensagem de um grupo. O RMP remetente deve, a partir dai, incrementar esse valor de 1 (um) para cada mensagem submetida nesse grupo."
Mensagens possuem identificação única.	Nesta EPMC, cada mensagem possui, em seu cabeçalho, um parâmetro de identificação única de mensagem?	**(OASIS, 2004b, p. 15):** "Uma mensagem confiável contém um identificador que é globalmente único e baseado no conceito de um grupo. [...] O identificador de mensagem confiável é uma combinação de um ID de grupo e um número de seqüência opcional. [...] Quando há apenas uma mensagem no grupo: o ID de grupo, que é um identificador de grupo único global, pode ser utilizado sozinho como identificador de mensagem. [...] Quando a mensagem pertence a um grupo de várias mensagens: a mensagem é identificada pelo ID de grupo e um número de seqüência único."
Mensagens possuem identificação de grupo de mensagens.	Nesta EPMC, cada mensagem possui, em seu cabeçalho, um parâmetro de identificação referente ao grupo ao qual pertence?	**(OASIS, 2004b, p. 15):** "Uma mensagem confiável sempre pertence a um grupo." **(OASIS, 2004b, p. 15):** "O ID de grupo é um identificador de grupo único global." **(OASIS, 2004b, p. 24):** "Este atributo <*GroupId*> identifica um grupo de mensagem. O RMP remetente deve usar um valor único globalmente distinto para cada grupo distinto de mensagens. Dentro de um grupo, todas as mensagens terão o mesmo valor de <*GroupId*>. Essa identificação (o valor) é do tipo URI, tal como definida na RFC 2396."

(Continua)

(Continuação)

Requisito	Questionamento	Requisitos encontrados
Mensagens possuem marcas de tempo.	Esta EPMC insere parâmetros de marcas de tempo no cabeçalho das mensagens enviadas?	**(OASIS, 2004b, p. 16):** Existem três elementos de marca de tempo utilizados por esta EPMC em cada mensagem: O elemento <*ExpiryTime*> é do tipo *xs:DateTime* e definido como "data e hora após as quais uma mensagem não poderá mais ser entregue", ou seja, expirou; O elemento <*GroupExpiryTime*> que também é do tipo *xs:DateTime* e definido como "data e hora após as quais o grupo pode ser terminado", ou seja o grupo, expirou; O elemento <*GroupMaxIdleDuration*> é do tipo *xs:Duration* e definido como "tempo limite decorrido desde a última mensagem enviada ou recebida em um grupo, após o qual o grupo pode ser terminado." Esses três elementos são opcionais. Para os dois primeiros acima, seus conteúdos são expressos de acordo com o UTC.

Observações complementares:

Quanto ao requisito de parâmetro de controle de integridade, a presente EPMC cita ser extremamente recomendado que a integridade de mensagem seja garantida fim-a-fim, pelo uso de opções adequadas de segurança tais como as descritas em outra proposta de padrão, para segurança em *Web services*, a *WS-Security*.

Quanto às mensagens possuírem marcas de tempo, *xs:DateTime* é um tipo de dados primitivo de XML que representa um instante específico de tempo. O espaço de valores desse tipo de dados é o espaço de combinações de datas e horas definidos na norma ISO 8601, de 1988. O *xs:Duration* é um tipo de dados primitivo de XML que representa uma duração de tempo, com seis dimensões (ano, mês, dia, hora, minutos e segundos), de acordo com a norma ISO 8601, de 1988. O UTC é o acrônimo de *Universal Time Coordinated*, também conhecido como tempo civil. Trata-se do fuso horário de referência a partir do qual se calculam todas as outras zonas horárias do mundo.

4.1.2.2 Quanto ao esquema estático

Representando estados possíveis de uma mensagem, a Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao esquema estático é apresentada na tabela 4.5.

Tabela 4.5. Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao esquema estático.

Requisito	Questionamento	Requisitos encontrados
Estados possíveis de um objeto informação: "mensagem de interação" e "mensagem de reconhecimento".	Nesta EPMC, uma mensagem pode ser de interação (que corresponde ao envio de conteúdo específico de aplicação) ou de reconhecimento (que corresponde à resposta do destinatário ao remetente, reconhecendo uma mensagem recebida)?	**(OASIS, 2004b, p. 11):** O modelo de troca de mensagens desta EPMC é ilustrado por meio de um diagrama (vide Anexo B do presente trabalho) onde tem-se que o objeto informação trocado entre os dois componentes de interoperabilidade (*Sending RMP* e *Receiving RMP*) pode ser: *Reliable Message* ou *RM-Reply*. O primeiro objeto informação são mensagens de conteúdo de aplicação definidas como "uma mensagem SOAP contendo uma solicitação" ou "uma mensagem de resposta com conteúdo (*payload*) referente a uma mensagem de solicitação previamente recebida." O outro, *RM-Reply*, é uma mensagem de reconhecimento.

4.1.2.3 Quanto ao esquema dinâmico

A Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao esquema dinâmico, referente às ações sobre as mensagens, que implicam em mudanças de estado destas, é apresentada na tabela 4.6.

Tabela 4.6. Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao esquema dinâmico.

Requisito	Questionamento	Requisitos encontrados
Atividades do remetente para enviar mensagem: preparação e envio.	Nesta EPMC, quando um componente remetente é solicitado a enviar mensagens (pelas camadas de aplicação em seu agente), ele, antes de enviar ao destinatário, prepara a mensagem (inclui cabeçalhos com informações adicionais para controle de entrega confiável)?	**(OASIS, 2004b, p. 21):** A especificação define a estrutura dos blocos de cabeçalho para um envelope de mensagem confiável SOAP. Todo o contexto inserido no cabeçalho de uma mensagem SOAP está identificado pelo prefixo *wsrm* (*Web Services Reliable Messaging).* Nele, inserem-se os elementos: <*GroupID*> que identifica o grupo e o <*SequenceNum*> que contém o número de seqüência da mensagem nesse grupo. Ou seja, a mensagem, antes de ser enviada, recebe um bloco *MessageID*, no cabeçalho, contendo os identificadores de grupo e de número de seqüência de mensagem. A identificação única de mensagem é obtida pela concatenação desses dois elementos (<*GroupID*> e <*SequenceNum*>). Além disso, são inseridos elementos de tempo de expiração de mensagem <*ExpiryTime*> e de expiração de grupo <*GroupExpiryTime*> e o <*GroupMaxIdleDuration*>. Ou seja, todas as mensagens de conteúdo específico de aplicação possuem marcas de tempo. Elementos de controle de integridade não são inseridos, pois, como visto, esses aspectos não são contemplados nesta EPMC.
Atividades do destinatário ao receber mensagem: checagem.	Nesta EPMC, ao receber uma mensagem, o componente destinatário realiza atividades de checagem da mensagem, antes de entregá-la à camada superior (aplicação) de seu agente *Web services*?	**(OASIS, 2004b, p. 12):** "Os seguintes requisitos estão associados com o uso de operações RMP: Para cada mensagem válida e não expirada recebida, o RMP destinatário deve invocar a operação de entrega assim que os requisitos de confiabilidade associados (ordenação, ausência de duplicação) tenham sido satisfeitos." **(OASIS, 2004b, p. 26):** "O elemento <*ExpiryTime*> representa um acordo de tempo de expiração [...]. Ele indica a data e hora limites depois dos quais o RMP destinatário não deverá entregar a mensagem recebida à aplicação."
Reconhecimento de mensagem.	Nesta EPMC, as mensagens corretamente recebidas e aceitas pelo destinatário, determinam o envio de reconhecimento do destinatário ao remetente?	**(OASIS, 2004b, p. 10):** "Reconhecimento é uma indicação que se refere a uma mensagem previamente recebida pelo destinatário, indicando que a mensagem foi corretamente recebida (ou seja, que a mensagem atendeu aos requisitos de entrega confiável)." **(OASIS, 2004b, p. 13):** A especificação ilustra os modelos de resposta de mensagem (*Message Reply Patterns*), inserindo reconhecimento (chamados de *RM-Reply* nessa EPMC), respectivos a cada tipo de MEP (*Message Exchange Pattern*) do WSA (*Web Services Architecture*). Nesse caso, o destinatário sempre envia um reconhecimento (*RM-Reply*) ao remetente, assim que aceita uma mensagem. Vide Anexo B do presente trabalho.

4.1.3 Matrizes para o ponto de vista da computação

Este ponto de vista foca na captura dos componentes de interoperabilidade *Web services* e suas ligações, em termos de interações, interfaces e conexões.

4.1.3.1 Quanto ao tipo de interface

A Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao tipo de interface utilizado nessa EPMC está representada na tabela 4.7.

Tabela 4.7. Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao tipo de interface.

Requisito	Questionamento	Requisitos encontrados
A interface do componente de interoperabilidade é do tipo "operação".	Nesta EPMC, a interoperabilidade ocorre por interações que designam uma relação entre cliente/servidor, caracterizando uma interface do tipo operação?	**(OASIS, 2004b, p. 7):** As mensagens são envidas do produtor para o consumidor via seus componentes de interoperabilidade, chamados de RMP (*Reliable Messaging Processors*). Vide Anexo A do presente trabalho. **(OASIS, 2004b, p. 9):** "O produtor é um componente abstrato que produz o conteúdo (*payload*) de uma mensagem a ser enviada." "O consumidor é um componente abstrato que consome o conteúdo (*payload*) de uma mensagem recebida." **(OASIS, 2004b, p. 12):** "Esta especificação suporta capacidades de mensagem confiável para tipo de operação *request/response*" em *Web services*. **(OASIS, 2004b, p. 11):** Segundo o item "*Messaging Context*", representa-se um diagrama de seqüência (vide Anexo B do presente trabalho) onde o objeto informação trocado entre os dois componentes de interoperabilidade (*Sending RMP* e *Receiving RMP*) pode ser *Reliable Message* ou *RM-Reply*, que correspondem, respectivamente a uma mensagem SOAP contendo uma solicitação e uma mensagem de reconhecimento.

4.1.3.2 Quanto ao modo de interação

A Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao modo de interação utilizado nesta EPMC está representada na tabela 4.8.

Tabela 4.8. Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto ao modo de interação.

Requisito	Questionamento	Requisitos encontrados
Utiliza-se reconhecimento.	Esta EPMC utiliza esquema de reconhecimento onde o componente remetente recebe reconhecimentos positivos e, opcionalmente negativos, advindos do componente destinatário, referentes a mensagens recebidas e aceitas por esse?	**(OASIS, 2004b, p. 6):** "Entrega garantida de mensagem é definida por um protocolo de mensagem envolvendo indicações de reconhecimento e cabeçalhos de mensagem específicos." **(OASIS, 2004b, p. 10):** "A *RM-Reply* é uma indicação – indicação de reconhecimento – referindo-se a uma mensagem confiável previamente enviada."
A interação ocorre segundo um modo específico.	Nesta EPMC, o envio de mensagens entre as duas partes, após estabelecimento de conexão, pode ocorrer segundo um dos modos de interação (modo básico, modo *go-back-n* ou modo de repetição seletiva)?	**(OASIS, 2004b, p. 12):** "Os seguintes requisitos estão associados com o uso de operações RMP: Para cada mensagem válida e não expirada recebida, o RMP destinatário deve invocar a operação de entrega assim que os requisitos de confiabilidade associados (ordenação, ausência de duplicação) tenham sido satisfeitos." **(OASIS, 2004b, p.13):** São descritos os dois MEPs suportados. No caso do *one-way*, o RMP remetente envia uma mensagem com conteúdo ao RMP destinatário. Embora não seja esperada uma resposta com conteúdo, uma mensagem de reconhecimento pode ser retornada do RMP destinatário. No caso do MEP tipo *request-response*, a mensagem de retorno corresponderá a um reconhecimento (*RM-Reply*). Vide Anexo B do presente trabalho.

Observações complementares:

Na EPMC *WS-Reliability*, o conceito de operação de entrega refere-se à passagem da mensagem, recebida pelo componente de interoperabilidade destinatário, à respectiva aplicação de seu agente *Web services*, com subseqüente envio de reconhecimento positivo ao remetente.

Segundo verificado nesta EPMC, o modo de interação utilizado é o modo básico. Ou seja, para cada mensagem enviada do remetente ao destinatário, espera-se um reconhecimento antes que outra mensagem seja enviada.

4.1.3.3 Quanto aos modos de conexão e desconexão

A Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto aos modos de conexão e desconexão utilizados está representada na tabela 4.9.

Tabela 4.9. Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto aos modos de conexão e desconexão.

Requisito	Questionamento	Requisitos encontrados
Há estabelecimento de conexão.	Nesta EPMC, o envio de mensagens é precedido por uma fase preliminar para o estabelecimento de conexão, sendo que essa pode ser implícita ou explícita?	**(OASIS, 2004b, p. 13):** "O remetente está apto a iniciar o envio de mensagem, sem necessidade de uma ação prévia de protocolo." No item "*Message Reply Pattern*" são representados os modelos de resposta de mensagem com inserção de reconhecimentos (chamados de *RM-Reply* nessa EPMC), respectivos a cada tipo de MEP (*Message Exchange Pattern*) do WSA (*Web Services Architecture*). Nos modelos apresentados pela especificação, o remetente inicia envio de mensagem sem necessidade de uma interação preliminar. Tal abordagem corresponde a um modo de conexão implícita. Vide Anexo A do presente trabalho. **(OASIS, 2004b, p. 18):** "Um dos princípios dessa especificação é não ser necessário enviar um estabelecimento de acordo prévio entre as partes antes de iniciar transações de negócios." "Nenhuma comunicação prévia para estabelecimento de acordo com a parte destinatária (*Receiving RMP*) é necessária."
Há encerramento de conexão (desconexão).	Nesta EPMC, o envio de mensagens é seguido por uma fase de término (desconexão implícita ou explícita)?	**(OASIS, 2004b, p. 43):** Nesta EPMC, o encerramento de uma conexão, ou seja, quando os recursos persistentes referentes ao envio de uma seqüência de mensagens são liberados, é chamado de "término de grupo." Um término de grupo pode ocorrer por *time-out*, ou seja, quando o tempo especificado no elemento <*GroupMaxIdleDuration*> for ultrapassado, o remetente considerará o fim da persistência referente a essa seqüência de mensagens (esse grupo). Além disso, um término de grupo pode ocorrer quando ultrapassado o tempo previsto a esse, especificado no elemento <*GroupExpiryTime*>.

Observações complementares:

Quanto aos modos de conexão e desconexão, pelo verificado, essa EPMC utiliza conexão e desconexão implícitas. O elemento <*GroupMaxIdleDuration*> faz parte do cabeçalho das mensagens nesta EPMC que corresponde ao máximo intervalo de tempo entre uma mensagem e outra, em uma seqüência. Caso esse tempo seja ultrapassado, considera-se como desconexão, ou seja, término do grupo.

4.1.4 Matriz para o ponto de vista da engenharia

O ponto de vista da engenharia segundo o Modelo de Referência proposto, foca no modelo de canal, composto por objetos adaptador, conector e protocolo. Na tabela 4.10 apresenta-se a Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability*, referentes este ponto de vista.

Tabela 4.10. Matriz de Mapeamento de Requisitos da EPMC *WS-Reliability* quanto aos objetos do canal.

Requisito	Questionamento	Requisitos encontrados
Objetos adaptadores.	Nesta EPMC, há especificação de objetos adaptadores que são responsáveis pela adaptação do cabeçalho das mensagens para comportar os parâmetros necessários ao controle de erros: número de seqüência, identificação de mensagem, identificação de grupo de mensagem e parâmetro de detecção de erro de integridade?	Não encontrado.
Objetos conectores.	Nesta EPMC, há especificação de objetos conectores, responsáveis por implementar estabelecimento de conexão (implícita ou explícita)?	Não encontrado.

(Continua)

(Continuação)

Requisito	Questionamento	Requisitos encontrados
Objetos protocolos.	Nesta EPMC, há especificação de objeto protocolo, responsável por implementar modos de interação segundo um dos modos: modo básico, modo *go-back-n* ou modo de repetição seletiva?	Não encontrado.
Temporização de retransmissão	Nesta EPMC, há especificação de algoritmo de para controle de tempo de retransmissão de mensagens pelo objeto protocolo, segundo algum padrão?	Não encontrado.

Observações complementares:

Sobre os objetos adaptadores, apesar da EPMC *WS-Reliabiltiy* não apresentar especificação de objeto responsável por implementar adaptação de mensagem, suas atividades são realizadas como verificado em sua Matriz de Mapeamento de Requisitos quanto ao esquema dinâmico (tabela 4.6).

Para os objetos conectores, apesar desta EPMC não apresentar especificação de objeto responsável por implementar estabelecimento e encerramento de conexão, suas atividades são realizadas como verificado nos modos de interação, na Matriz de Mapeamento de Requisitos quanto aos modos de conexão (tabela 4.9).

Quanto aos objetos protocolos, apesar desta EPMC não apresentar especificação de objeto responsável por implementar interações, suas atividades são realizadas, como verificado nos modos de interação, na Matriz de Mapeamento de Requisitos do ponto de vista quanto ao modo de interação (tabela 4.8).

A presente EPMC, não cita qual algoritmo é utilizado para controle de intervalo de tempo de retransmissão.

Com esta última matriz apresentada, conclui-se a verificação dos trinta requisitos definidos no Modelo de Referência sobre a especificação da EPMC *WS-Reliability*. Passa-se então à verificação da EPMC *WS-ReliableMessaging*.

4.2 Mapeamento de Requisitos da *EPMC WS-ReliableMessaging*

Neste item, apresentam-se as Matrizes de Mapeamento de Requisitos referente à verificação da *WS-ReliableMessaging*, segundo o Método de Verificação proposto. Para a verificação desta EPMC, adota-se como referência técnica a sua documentação de especificação em (OASIS, 2006c).

A cada matriz, informações adicionais e pertinentes à EPMC, para cada requisito verificado, que venham a complementar o entendimento aos requisitos encontrados, são inseridas à parte, como observações complementares.

4.2.1 Matrizes para o ponto de vista da empresa

1) Requisitos de escopo, comunidade, federação, funções empresariais e procedimentos. A Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto a esses requisitos do ponto de vista da empresa é apresentada na tabela 4.11.

Tabela 4.11. Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao escopo, comunidade, federação, funções empresariais e procedimentos.

Requisito	Questionamento	Requisitos encontrados
Escopo: prover interoperabilidade confiável.	Esta EPMC tem como propósito prover garantia de entrega de mensagem?	**(OASIS, 2006c, p. 1):** "Esta especificação (*WS-ReliableMessaging*) descreve um protocolo que permite mensagens serem transferidas de forma confiável entre nós que implementem este protocolo, na presença de falhas de componentes de software, sistema ou rede de comunicação." **(OASIS, 2006c, p. 4):** "O objetivo primário desta especificação é criar um mecanismo modular para transferência confiável de mensagens." **(OASIS, 2006c, p. 6):** "A especificação *WS-ReliableMessaging* define um protocolo interoperável que habilita um remetente de

(Continua)

(Continuação)

Requisito	Questionamento	Requisitos encontrados
		mensagem confiável a determinar precisamente a entrega de cada mensagem que ele transmite ao destinatário, permitindo resolver quaisquer dúvida de status relativo ao recebimento de uma mensagem transmitida. O protocolo também habilita um destinatário a determinar eficientemente quais das mensagens que ele recebe já tenham sido previamente recebidas, permitindo a ele descartar mensagens duplicadas, causadas por retransmissão pelo remetente. Além disso, ele também habilita ao componente destinatário a entrega de mensagens à sua respectiva aplicação destinatária, na ordem em que essas foram transmitidas pelo remetente, no caso de eventual recebimento fora de ordem."
Comunidade e federação: componentes de interoperabilidade.	Segundo esta EPMC, a interação entre agentes é implementada por componentes de interoperabilidade que trocam mensagens entre si?	**(OASIS, 2006c, p. 6):** "A especificação *WS-ReliableMessaging* define um protocolo interoperável que habilita um remetente de mensagem confiável a determinar precisamente a entrega de cada mensagem que ele transmite ao destinatário[...]." Nessa mesma página, apresenta-se um diagrama com modelo de troca de mensagens (vide Anexo C do presente trabalho), onde representa-se uma interação ente dois agentes *Web services* (nesta EPMC chamados de *Initial Sender* e *Ultimate Receiver*). Esses agentes são compostos por elementos, dentre eles, os componentes de aplicação (*Application Source* e *Application Destination*) que, por sua vez, interagem entre si por meio de componentes de interoperabilidade, nesta EPMC chamados de *RM Source* e *RM Destination*. **(OASIS, 2006c, p. 7):** "*RM Source*: é o *endpoint* que transmite mensagens de forma confiável a um *RM Destination*." "*RM Destination*: é o *endpoint* que recebe mensagens transmitidas de forma confiável por um *RM Source*."
Componentes de interoperabilidade assumem funções empresariais distintas e realizam respectivos procedimentos.	Esta EPMC delimita as funções empresariais e respectivos procedimentos do componente de interoperabilidade a: 1) Componente remetente: aquele que envia a mensagem de forma confiável; e, 2) Componente destinatário: aquele que recebe a mensagem de forma confiável?	**(OASIS, 2006c, p. 6):** Apresenta-se um modelo de troca de mensagens (vide Anexo C do presente trabalho), onde representam-se os dois papéis possíveis entre componentes de interoperabilidade *Web services*, referenciados pelas notações "*RM Source*" e "*RM Destination*". **(OASIS, 2006c, p. 7):** "*RM Source*: é o *endpoint* que transmite mensagens de forma confiável a um *RM Destination*.." "*RM Destination*: é o *endpoint* que recebe mensagens transmitidas de forma confiável por um *RM Source*."

Observações complementares:

Embora a documentação cite que o objetivo primário da EPMC *WS-ReliableMessaging* é criar um mecanismo modular para transferência confiável de mensagens, aspectos de implementação são considerados fora de escopo nessa especificação (OASIS, 2006c).

Esta EPMC, utiliza a notação "*RM Source*" (*Reliable Messaging Source*), e "*RM Destination*" (*Reliable Messaging Destination*), que correspondem aos componentes de interoperabilidade de agentes *Web services*. A presente EPMC foca-se somente no escopo de comunicação entre estes, não se estendendo aos níveis de aplicação (chamados de *Application Source* e *Application Destination*).

2) Requisitos de contrato. A Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao contrato é apresentada na tabela 4.12.

Tabela 4.12. Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao contrato.

Requisito	Questionamento	Requisitos encontrados
Entrega garantida de mensagem	Esta EPMC provê entrega de mensagem, garantido que uma mensagem enviada seja entregue ao seu destino?	**(OASIS, 2006c, p. 6):** "A especificação *WS-ReliableMessaging* define um protocolo interoperável que habilita um remetente de mensagem confiável a determinar precisamente a entrega de cada mensagem que ele transmite ao destinatário[...]." "Este protocolo habilita a implementação de uma grande faixa de características de confiabilidade que incluem entrega em ordem, eliminação de duplicação e recebimento garantido."
Entrega única de mensagem	Esta EPMC provê entrega de mensagem, garantindo a não duplicidade na entrega de uma mensagem ao seu destinatário?	**(OASIS, 2006c, p. 6):** "O protocolo também habilita um destinatário a determinar eficientemente quais das mensagens que ele recebe já tenham sido previamente recebidas, permitindo a ele descartar mensagens duplicadas, causadas por retransmissão pelo remetente." "Este protocolo habilita a implementação de uma grande faixa de características de confiabilidade que incluem entrega em ordem, eliminação de duplicação e recebimento garantido."

(Continua)

(Continuação)

Requisito	Questionamento	Requisitos encontrados
Entrega ordenada de mensagens.	Esta EPMC provê entrega de mensagem, garantido que uma seqüência de mensagens seja entregue na ordem em que foi enviada?	**(OASIS, 2006c, p. 6):** "A especificação *WS-ReliableMessaging* define um protocolo interoperável que habilita um remetente de mensagem confiável a determinar precisamente a entrega de cada mensagem que ele transmite ao destinatário, permitindo resolver quaisquer dúvidas de status relativo ao recebimento de uma mensagem transmitida [...]. Além disso, ele também habilita ao componente destinatário a entrega de mensagens à sua respectiva aplicação destinatária na ordem em que essas foram transmitidas pelo remetente, no caso de eventual recebimento fora de ordem." "Este protocolo habilita a implementação de uma grande faixa de características de confiabilidade que incluem entrega em ordem, eliminação de duplicação e recebimento garantido."
Entrega íntegra de mensagem.	Esta EPMC provê entrega de mensagem, garantindo que uma mensagem que chegue a um destinatário seja a mesma que foi enviada pelo remetente?	Não contemplado.

Observações complementares:

No requisito integridade, a EPMC *WS-ReliableMessaging* não faz assunções sobre requisitos de segurança das aplicações que utilizam entrega confiável. Segundo essa EPMC, se for necessária uma troca segura de mensagens, o remetente e o destinatário deverão ter um contexto de segurança a ser provido por outro padrão, no caso, o *WS-Security*.

3) Políticas. A Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto às políticas está representada na tabela 4.13.

Tabela 4.13. Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto às políticas.

Requisito	Questionamento	Requisitos encontrados
Independência de protocolo de transporte.	Esta EPMC visa a implementar entrega confiável de mensagens sem dependência quanto ao protocolo de transporte subjacente utilizado?	**(OASIS, 2006c, p. 1):** "O protocolo é descrito nesta especificação em um modo independente de transporte, permitindo que seja implementado utilizando-se diferentes tecnologias de rede."
Suporte a agrupamento de mensagens.	Esta EPMC permite que mensagens possam ser associadas a um grupo específico de mensagens?	**(OASIS, 2006c, p. 7):** "Durante o tempo de vida de uma seqüência, duas invariantes são requeridas para a correta operação: O *RM Source* deve atribuir a cada mensagem dentro de uma seqüência, um número de mensagem [...]." **(OASIS, 2006c, p. 8):** Apresenta-se um diagrama de seqüência, ilustrando exemplo de troca de mensagens (vide Anexo D do presente trabalho). Esta EPMC conceitua um grupo de mensagem como uma seqüência de mensagens. No referido diagrama, todas as mensagens enviadas de um componente a outro, pertencentes ao mesmo grupo (mesma seqüência), possuem o mesmo identificador de seqüência. **(OASIS, 2006c, p. 18):** "O protocolo RM usa um bloco de cabeçalho chamado *Sequence* [...]. O *RM Source* deve incluir um bloco de cabeçalho <*Sequence*> em todas as mensagens para as quais entrega garantida é requerida."
A interação é unidirecional entre componentes.	Esta EPMC tem um modelo de interoperabilidade cujo sentido de envio de mensagens com conteúdo específico de aplicação ocorre do componente remetente para componente destinatário?	**(OASIS, 2006c, p. 7):** "*RM Source*: é o *endpoint* que transmite mensagens de forma confiável a um *RM Destination*." "*RM Destination*: é o *endpoint* que recebe mensagens transmitidas de forma confiável por um *RM Source*." **(OASIS, 2006c, p. 8):** Apresenta-se um diagrama de seqüência que ilustra um exemplo de troca de mensagens segundo esta EPMC (vide Anexo D do presente trabalho). No caso, as seqüências de mensagens, com conteúdo de aplicação, são transmitidas sempre de um *endpoint* (A) para o *endpoint* (B). As mensagens em sentido contrário são reconhecimentos.
Deve-se atender aos requisitos de contrato.	Esta EPMC objetiva atender aos requisitos de contrato (entrega garantida, única, ordenada e íntegra)?	Vide Matriz de Mapeamento dos requisitos quanto ao contrato, apresentada na tabela 4.12.

4.2.2 Matrizes para o ponto de vista da informação

4.2.2.1 Quanto ao esquema invariante

Representando condições sempre verdadeiras de uma mensagem, a Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao esquema invariante é apresentada na tabela 4.14.

Tabela 4.14. Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao esquema invariante.

Requisito	Questionamento	Requisitos encontrados
A interação é baseada em mensagem.	Nesta EPMC, o objeto informação corresponde a uma mensagem, com estrutura baseada na estrutura do padrão SOAP?	**(OASIS, 2006c, p. 1):** "Esta especificação (*WS-ReliableMessaging*) descreve um protocolo que permite mensagens serem transferidas de forma confiável entre nós que implementem este protocolo, na presença de falhas de componentes de software, sistema ou rede de comunicação." **(OASIS, 2006c, p. 4):** Essa EPMC "define um protocolo de mensagens para a transferência confiável de mensagens entre um remetente e um destino."
Mensagens possuem integridade.	Nesta EPMC, as mensagens são enviadas contendo parâmetro de controle de integridade no cabeçalho?	Não contemplado.
Mensagens possuem numeração de seqüência.	Nesta EPMC, as mensagens são enviadas com um parâmetro de número de seqüência em seu cabeçalho?	**(OASIS, 2006c, p. 7):** "Durante o tempo de vida de uma seqüência, duas invariantes são requeridas para correta operação: O *RM Source* deve atribuir a cada mensagem dentro de uma seqüência, um número de mensagem [...]." **(OASIS, 2006c, p. 18):** "*RM Source* deve atribuir a cada mensagem dentro de uma seqüência um elemento <*MessageNumber*> que é incrementado de 1, a partir de um valor inicial igual a 1. Esses valores estão contidos em um bloco de cabeçalho chamado *Sequence*."
Mensagens possuem marcas de tempo.	Esta EPMC insere parâmetros de marcas de tempo nas mensagens enviadas?	**(OASIS, 2006c, p. 11):** Na inicialização de um grupo (chamado de criação de seqüência), "o *RM Source* deve solicitar a criação de uma seqüência, enviando um elemento <*CreateSequence*>."

(Continua)

(Continuação)

Requisito	Questionamento	Requisitos encontrados
		Um dos elementos que compõem a sintaxe do *CreateSequence* é o <*wsrm:expires*>. “Esse elemento, do tipo *xs:Duration* especifica a duração requerida pelo *RM Source* para uma seqüência”.
Mensagens possuem identificação de grupo de mensagens.	Nesta EPMC, cada mensagem possui, em seu cabeçalho, um parâmetro de identificação referente ao grupo ao qual pertence?	**(OASIS, 2006c, p. 18):** “O protocolo RM usa um bloco de cabeçalho de seqüência [...]. O *RM Source* deve incluir um bloco de cabeçalho de seqüência para cada mensagem que seja requerida como entrega confiável. O *RM Source* deve identificar seqüências com elementos identificadores únicos.” Na sintaxe do bloco *Sequence*, apresentada no documento de especificação dessa EPMC, esse bloco contem o elemento <*Identifier*>. Quanto ao elemento <*Identifier*>, o documento menciona que “Um *RM Source* que inclui um bloco de cabeçalho *Sequence* em um envelope SOAP deve incluir este elemento no bloco. O *RM Source* deve ajustar o valor desse elemento para conter um URI (em conformidade com a RFC 2396) absoluto, que identifique unicamente a seqüência.”
Mensagens possuem identificação única.	Nesta EPMC, cada mensagem possui, em seu cabeçalho, um parâmetro de identificação única de mensagem?	**(OASIS, 2006c, p. 18):** “O *RM Source* deve identificar seqüências com elementos identificadores únicos e o *RM Source* deve atribuir a cada mensagem de uma seqüência, um elemento de número de mensagem que incrementa de 1, a partir de um valor inicial igual a 1. Esses valores estão contidos dentro de um bloco de cabeçalho chamado *Sequence* que acompanha cada mensagem sendo transferida no contexto de uma seqüência.” Nesta EPMC, cada mensagem é única dentro de um grupo (chamado de seqüência), em função de seu elemento <*MessageNumber*> e cada seqüência é única, em função de seu identificador que é baseado em URI. A partir da combinação, concatenando-se esses dois elementos, tem-se uma identificação única de mensagem.

Observações complementares:

Na identificação de grupo, nesta EPMC, um grupo denomina-se “seqüência” (*sequence*) e é identificado pelo elemento <*Identifier*>, no cabeçalho da mensagem.

Referente à presença de parâmetro de controle de integridade, a presente EPMC delega ao escopo de outra proposta de padrão, a de segurança em *Web services* (a *WS-Security*), a incumbência de tratar a questão de integridade de mensagem.

No caso de presença de marcas de tempo, *Duration* é um tipo de dados primitivo de XML que representa uma duração de tempo, com seis dimensões (ano, mês, dia, hora, minutos e segundos), de acordo com a norma ISO 8601, de 1988 (W3C, 2001).

4.2.2.2 Quanto ao esquema estático

Representando estados possíveis de uma mensagem, a Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao esquema estático é apresentada na tabela 4.15.

Tabela 4.15. Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao esquema estático.

Requisito	Questionamento	Requisitos encontrados
Estados possíveis de um objeto informação: "mensagem de interação" e "mensagem de reconhecimento".	Nesta EPMC, uma mensagem pode ser de interação (que corresponde ao envio de conteúdo específico de aplicação) ou de reconhecimento (que corresponde à resposta do destinatário ao remetente, reconhecendo uma mensagem recebida)?	**(OASIS, 2006c, p. 6):** Apresenta-se um modelo de troca de mensagens, que representa os objetos de informação (mensagens) que são trocadas entre dois agentes (vide Anexo C do presente trabalho). Junto a este, a EPMC define: "O *RM Source* transmite uma ou mais vezes. Depois de aceitar a mensagem, o *RM Destination* a reconhece." **(OASIS, 2006c, p. 8):** Apresenta-se o protocolo, por meio de diagrama de seqüências, representando os objetos de informação (mensagens) que são trocadas entre dois agentes (chamados de *endpoints* A e B). Segundo esse diagrama, o *endpoint* A está instanciado como remetente e envia mensagens (*sequences*) 1, 2 e 3 que devem conter conteúdo específico de aplicação. O retorno de B para A são sempre reconhecimentos (*SequenceAcknowledgments*). Vide Anexo D do presente trabalho.

4.2.2.3 Quanto ao esquema dinâmico

A Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao esquema dinâmico, referente às ações sobre as mensagens, que implicam em mudanças de estado dessas, é apresentada na tabela 4.16.

Tabela 4.16. Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao esquema dinâmico.

Requisito	Questionamento	Requisitos encontrados
Atividades do remetente para enviar mensagem: preparação e envio.	Nesta EPMC, quando um componente remetente é solicitado a enviar mensagens (pelas camadas de aplicação em seu agente), ele, antes de enviar ao destinatário, prepara a mensagem (inclui cabeçalhos com informações adicionais para controle de entrega confiável)?	**(OASIS, 2006c, p. 6):** Descrevendo seu modelo de troca de mensagem, (vide Anexo C do presente trabalho), "Primeiro, o *Application Source* envia uma mensagem para transferência confiável. O *RM Source* aceita a mensagem e a transmite uma ou mais vezes. Depois de aceitar a mensagem, o *RM Destination* a reconhece. Finalmente o *RM Destination* entrega a mensagem ao *Application Destination*." **(OASIS, 2006c, p. 11):** O protocolo, definido por esta EPMC utiliza um bloco de cabeçalho chamado <*Sequence*> a fim de controlar a transferência confiável de mensagens. Um componente remetente deve incluir um bloco de cabeçalho <S*equence*> em todas as mensagens SOAP para as quais transferência confiável seja requerida. **(OASIS, 2006c, p. 18):** "O *RM Source* deve incluir um bloco de cabeçalho de seqüência <*Sequence*> para cada mensagem que seja requerida como entrega confiável. O *RM Source* deve identificar seqüências com elementos identificadores únicos." Na sintaxe do bloco <*Sequence*>, que é inserido no cabeçalho da mensagem SOAP para controle de transferência confiável de mensagens, esse bloco contém os elementos <*Identifier*> e <*MessageNumber*>. O elemento <*Identifier*> representa a identificação do grupo de mensagem e quanto a isso, esta EPMC convenciona que: "O *RM Source* deve ajustar o valor desse elemento para conter um URI (em conformidade com a RFC 2396) absoluto, que identifique unicamente a seqüência." O elemento <*MessageNumber*>, representa o número da mensagem dentro da seqüência (grupo de mensagens). Quanto a este, o "RM Source deve atribuir a cada mensagem, dentro de uma seqüência um número de mensagem, iniciando com 1 e sendo acrescido de exatamente 1 para cada mensagem subseqüente."

(Continua)

(Continuação)

Requisito	Questionamento	Requisitos encontrados
Atividades do destinatário ao receber mensagem: checagem.	Nesta EPMC, ao receber uma mensagem, o componente destinatário realiza atividades de checagem da mensagem, antes de entregá-la à camada superior (aplicação) de seu agente *Web services*?	**(OASIS, 2006c, p. 6):** Descrevendo seu modelo de troca de mensagem (vide Anexo C do presente trabalho), "Primeiro, o *Application Source* envia uma mensagem para transferência confiável. O *RM Source* aceita a mensagem e a transmite uma ou mais vezes. Depois de aceitar a mensagem, o *RM Destination* a reconhece. Finalmente o *RM Destination* entrega a mensagem ao *Application Destination*." "O protocolo também habilita um destinatário a determinar eficientemente quais das mensagens que ele recebe já tenham sido previamente recebidas, permitindo a ele descartar mensagens duplicadas, causadas por retransmissão pelo remetente. Além disso, ele também habilita ao componente destinatário a entrega de mensagens à sua respectiva aplicação destinatária, na ordem em que essas foram transmitidas pelo remetente, no caso de eventual recebimento fora de ordem." "Aceite: é o ato de qualificação de mensagem pelo *RM Destination* tal que ela se torna elegível para entrega e reconhecimento."
Reconhecimento de mensagem	Nesta EPMC, as mensagens corretamente recebidas e aceitas pelo destinatário, determinam o envio de reconhecimento do destinatário ao remetente?	**(OASIS, 2006c, p. 6):** Descrevendo seu modelo de troca de mensagem, (vide Anexo C do presente trabalho), "Primeiro, o *Application Source* envia uma mensagem para transferência confiável. O *RM Source* aceita a mensagem e a transmite uma ou mais vezes. Depois de aceitar a mensagem, o *RM Destination* a reconhece. Finalmente o *RM Destination* entrega a mensagem ao *Application Destination*." "Depois de aceitar a mensagem, o *RM Destination* envia um reconhecimento".

Observações complementares:

Quanto às atividades do destinatário ao receber mensagem, o conceito de "entrega", na EPMC *WS-ReliableMessaging* refere-se à passagem da mensagem, recebida pelo componente de interoperabilidade destinatário (*RM Destination*), à respectiva aplicação, chamada por esta EPMC de *Application Destination.*

4.2.3 Matrizes para o ponto de vista da computação

Este ponto de vista foca na captura dos componentes de interoperabilidade *Web services* e suas ligações, em termos de interações, interfaces e conexões.

4.2.3.1 Quanto ao tipo de interface

A Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao tipo de interface utilizado nesta EPMC está representada na tabela 4.17.

Tabela 4.17. Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao tipo de interface.

Requisito	Questionamento	Requisitos encontrados
A interface do componente de interoperabilidade é do tipo "operação".	Nesta EPMC, a interoperabilidade ocorre por interações que designam uma relação entre cliente/servidor, caracterizando uma interface do tipo operação?	**(OASIS, 2006c, p. 6):** "O *WS-ReliableMessaging* define um protocolo interoperável que habilita remetente de mensagem confiável (*RM Source*) determinar precisamente a entrega de uma mensagem que ele transmite a um destinatário (*RM Destination*)." **(OASIS, 2006c, p. 7):** "*RM Source*: é o *endpoint* que transmite mensagens de forma confiável a um *RM Destination*." "*RM Destination*: é o *endpoint* que recebe mensagens transmitidas de forma confiável por um *RM Source*." **(OASIS, 2006c, p. 8):** Representa-se o protocolo, por meio de um diagrama de seqüências (vide Anexo D do presente trabalho), mostrando que *endpoints* A e B interagem, trocando mensagens entre si. As duas primeiras mensagens, representam uma interação de interrogação, onde o *endpoint* A solicita algo (um pedido de inicio de seqüência) e o *endpoint* B responde (confirmando e provendo um identificador para essa seqüência). Após várias interações, de envio de mensagens com conteúdo específico de aplicação, ocorrem duas últimas interações onde o *endpoint* A solicita finalização da seqüência ao *endpoint* B e esse último confirma.

4.2.3.2 Quanto ao modo de interação

A Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao modo de interação utilizado nesta EPMC está representada na tabela 4.18.

Tabela 4.18. Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto ao modo de interação.

Requisito	Questionamento	Requisitos encontrados
Utiliza-se reconhecimento.	Esta EPMC utiliza esquema de reconhecimento onde o componente remetente recebe reconhecimentos positivos e, opcionalmente negativos, advindos do componente destinatário, referentes a mensagens recebidas e aceitas por esse?	**(OASIS, 2006c, p. 6):** Descrevendo seu modelo de troca de mensagem, (vide Anexo C do presente trabalho), "Primeiro, o *Application Source* envia uma mensagem para transferência confiável. O *RM Source* aceita a mensagem e a transmite uma ou mais vezes. Depois de aceitar a mensagem, o *RM Destination* a reconhece. Finalmente o *RM Destination* entrega a mensagem ao *Application Destination*." "Depois de aceitar a mensagem, o *RM Destination* a reconhece." **(OASIS, 2006c, p. 7):** "Reconhecimento: a comunicação do *RM Destination* para o *RM Source* indicando correto recebimento de uma mensagem." **(OASIS, 2006c, p. 8):** "Dentro de toda mensagem de reconhecimento enviada, o *RM Destination* deve incluir [...] o número de mensagem de cada mensagem aceita pelo *RM Destination*." Apresenta-se um diagrama de seqüências que ilustra, em sua 6ª interação (linha pontilhada do destinatário ao remetente), reconhecimento acumulativo das mensagens 1 a 3, que são recebidas pelo *endpoint* B (no caso, o *RM Destination*). Vide Anexo D do presente trabalho. **(OASIS, 2006c, p. 9):** "O *RM Source* espera receber reconhecimentos do *RM Destination* durante o curso de uma troca de mensagem." **(OASIS, 2006c, p. 21):** "O *RM Destination* pode incluir o elemento NACK dentro de um bloco de cabeçalho de reconhecimento. Se utilizado, o *RM Destination* deve ajustar o valor desse elemento para o número da mensagem correspondente a uma mensagem não recebida em uma seqüência."
A interação ocorre segundo um modo específico.	Nesta EPMC, o envio de mensagens entre as duas partes, após estabelecimento de conexão, pode ocorrer segundo um dos modos de interação (modo básico, modo *go-back-n* ou modo de repetição seletiva)?	**(OASIS, 2006c, p. 8):** Representa-se o protocolo, por meio de um diagrama de seqüências (vide Anexo D do presente trabalho), mostrando que *endpoints* A e B interagem, trocando mensagens entre si. Após as duas primeiras mensagens, que representam o estabelecimento de conexão, dá-se inicio ao envio de uma seqüência de mensagens. A seqüência é representada por três mensagens enviadas do *endpoint* A para o *endpoint* B, com números de mensagem <*MessageNumber*> iguais a 1, 2 e 3. A mensagem de retorno chamada de *SequenceAccknowledgement* faz um reconhecimento acumulativo de uma faixa de mensagens, deixando de reconhecer a mensagem número 2 (que foi perdida). Em seguida, o *endpoint* A, reenvia somente a mensagem com <*MessageNumber*> igual a 2.

Observações complementares:

O modelo de interoperabilidade verificado na EPMC *WS-ReliableMessaging*, apresenta-se, conforme Modelo de Referência, como sendo do modo repetição seletiva.

Quanto à utilização de reconhecimento positivo, esse aspecto é mandatório na presente EPMC. Opcionalmente é aceito o uso de reconhecimento negativo.

4.2.3.3 Quanto aos modos de conexão e desconexão

A Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto aos modos de conexão utilizados nesta EPMC está representada na tabela 4.19.

Tabela 4.19. Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto aos modos de conexão e desconexão.

Requisito	Questionamento	Requisitos encontrados
Há estabelecimento de conexão.	Nesta EPMC, o envio de mensagens é precedido por uma fase preliminar para o estabelecimento de conexão, sendo que essa pode ser implícita ou explícita?	**(OASIS, 2006c, p. 8):** Representa-se o protocolo, por meio de um diagrama de seqüências (vide Anexo D do presente trabalho), mostrando que *endpoints* A e B interagem, trocando mensagens entre si. Antes de iniciar o envio de uma seqüência de três mensagens, há uma fase preliminar onde o *endpoint A* solicita a criação de uma seqüência ao *endpoint B* e esse último responde confirmando a criação de seqüência. Trata-se de um modo de estabelecimento de conexão do tipo *2-way-handshake*. **(OASIS, 2006c, p. 10):** "Criação de seqüência: O *RM Source* deve solicitar a criação de uma seqüência de envio, enviando um elemento <*CreateSequence*> ao *RM Destination* que responde de volta com uma confirmação <*CreateSequenceResponse*> ou recusa <*CreateSequenceRefused*>."

(Continua)

(Continuação)

Requisito	Questionamento	Requisitos encontrados
Há encerramento de conexão (desconexão).	Nesta EPMC, o envio de mensagens é seguido por uma fase de término (desconexão implícita ou explícita)?	**(OASIS, 2006c, p. 8):** Representa-se o protocolo, por meio de um diagrama de seqüências (vide Anexo D do presente trabalho), mostrando que *endpoints* A e B interagem, trocando mensagens entre si. Após o término do envio de uma seqüência de mensagens do *endpoint A* para o *endpoint B*, o primeiro envia um pedido de desconexão <*TerminateSequence*>. Nesse caso, o *endpoint B* confirma esse término com um <*TerminateSequenceResponse*>. **(OASIS, 2006c, p. 16):** "Quando o *RM Source* conclui o envio de uma seqüência, ele envia um elemento <*TerminateSequence*> no corpo de uma mensagem para indicar que a seqüência está concluída e que ele não irá enviar quaisquer mensagens adicionais relativas a essa seqüência. O *RM Destination* pode liberar quaisquer recursos associados com essa seqüência."

Observações complementares:

O modelo de interoperabilidade verificado na EPMC *WS-ReliableMessaging*, apresenta uma interação preliminar ao envio de seqüência de mensagens, que caracteriza o estabelecimento de uma conexão (no caso, o estabelecimento e criação de uma seqüência de envio). Essa conexão é explícita e composta por uma solicitação de criação, seguida de uma confirmação/recusa. Trata-se de uma conexão explícita com procedimento *2-way-handshake*.

O modelo de interoperabilidade verificado nesta EPMC, apresenta uma interação posterior ao envio de seqüência de mensagens, que caracteriza o encerramento de uma conexão (no caso, o término de uma seqüência de envio). A desconexão é explícita e composta por uma solicitação de término de seqüência enviada do remetente (*RM Source*) ao destinatário (*RM_Destination*), sendo que esse último confirma o término de seqüência. Trata-se de uma desconexão com procedimento *2-way-handshake*, conforme modos de desconexão definidos no ponto de vista da computação do Modelo de Referência proposto no presente trabalho. O tipo de desconexão implícita também é possível, nesta EPMC, em função do elemento <*expires*> que determina o tempo máximo de duração de uma seqüência.

4.2.4 Matriz para o ponto de vista da engenharia

O ponto de vista da engenharia segundo o Modelo de Referência proposto, foca no modelo de canal, composto por objetos adaptador, conector e protocolo. Na tabela 4.20 apresenta-se a Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging*, referentes a este ponto de vista.

Tabela 4.20. Matriz de Mapeamento de Requisitos da EPMC *WS-ReliableMessaging* quanto aos objetos do canal.

Requisito	Questionamento	Requisitos encontrados
Objetos adaptadores.	Nesta EPMC, há especificação de objetos adaptadores que são responsáveis pela adaptação do cabeçalho das mensagens para comportar os parâmetros necessários ao controle de erros: número de seqüência, identificação de mensagem, identificação de grupo de mensagem e parâmetro de detecção de erro de integridade?	Não encontrado.
Objetos conectores.	Nesta EPMC, há especificação de objetos conectores, responsáveis por implementar estabelecimento de conexão (implícita ou explícita)?	Não encontrado.

(Continua)

(Continuação)

Requisito	Questionamento	Requisitos encontrados
Objetos protocolos.	Nesta EPMC, há especificação de objeto protocolo, responsável por implementar modos de interação segundo um dos modos: modo básico, modo *go-back-n* ou modo de repetição seletiva?	Não encontrado.
Temporização de retransmissão	Nesta EPMC, há especificação de algoritmo de para controle de tempo de retransmissão de mensagens pelo objeto protocolo, segundo algum padrão?	**(OASIS, 2006c, p. 9):** "Os implementadores são encorajados a utilizar mecanismos adaptativos, que ajustam retransmissão dinamicamente, que sejam apropriados à natureza de transporte. Um mecanismo similar ao descrito como RTTM na RFC 1323 deverá ser considerado."

Observações complementares:

Sobre objetos adaptadores, apesar da EPMC *WS-ReliableMessaging* não apresentar especificação de objeto responsável por implementar adaptação de mensagem, suas atividades são realizadas como verificado na Matriz de Mapeamento de Requisitos quanto ao esquema dinâmico (tabela 4.16).

Para objetos conectores, apesar desta EPMC não apresentar especificação de objeto responsável por implementar estabelecimento e encerramento de conexão, suas atividades são realizadas como verificado nos modos de interação, na Matriz de Mapeamento de Requisitos quanto aos modos de conexão (tabela 4.19).

Quanto aos objetos protocolos, esta EPMC não apresenta especificação de objeto responsável por implementar interações. Entretanto, suas atividades são realizadas, como verificado nos modos de interação, na Matriz de Mapeamento de Requisitos do ponto de vista quanto ao modo de interação (tabela 4.18).

Com esta última matriz apresentada, conclui-se a verificação dos trinta requisitos definidos no Modelo de Referência sobre a especificação da EPMC *WS-ReliableMessaging*. Para melhores conclusões qualitativas sobre os resultados, de verificação, passa-se a uma comparação entre as respectivas Matrizes de Mapeamento de Requisitos das duas EPMCs, a seguir.

4.3 Comparação entre as EPMCs *WS-Reliability* e *WS-ReliableMessaging*

A partir de resultados individuais de verificação é possível estabelecer comparação qualitativa entre as EPMCs (DOBRICA; NIEMELA, 2002).

Ao comaparar as EPMCs, tomando-se seus resultados de verificação, tem-se melhor suporte a conclusões sobre qual das duas EPMCs tem melhor conformidade às necessidades de mensagem confiável e, portanto, é mais indicada a ser adotada – segundo o presente método – ao padrão de mensagens confiáveis em *Web services*.

Para estabelecer a comparação entre as EPMCs, tomam-se cada um dos trinta requisitos, referentes aos quatro pontos de vista estabelecidos no Modelo de Referência, como parâmetros de comparação.

Visando a melhor representação de comparação, adotam-se formato e estrutura similares às utilizadas na apresentação das Matrizes de Verificação de Requisitos. Portanto, a comparação é dividida em quatro grandes contextos, referentes aos quatro pontos de vista, sendo que cada ponto de vista, com subdivisões, agrupam respectivos requisitos. Cada grupo de requisitos é apresentado em uma tabela de comparação.

Cada tabela de comparação possui três colunas, sendo uma para requisitos – que são parâmetro de comparação - e uma para cada uma das duas EPMCs comparadas (*WS-Reliability* e *WS-ReliableMessaging*). Cada requisito estabelece uma linha de comparação na tabela.

A cada intersecção de linha-coluna da tabela, descreve-se como a EPMC atende ao respectivo requisito. O contexto dessa descrição tem origem nos resultados de

verificação da EPMC, para o respectivo requisito, mas adotando-se uma linguagem natural, representando o entendimento aos requisitos encontrados na verificação.

Além disso, para cada tabela apresentada, procede-se um descritivo sucinto sobre o comparativo a cada requisito, incluindo-se informações adicionais e pertinentes que venham a complementar o entendimento da comparação. Para alguns requisitos onde a comparação entre as duas EPMCs apresenta diferenciação, complementa-se com aspectos sobre como outros sistemas, no caso, protocolos ditos de entrega confiável e/ou sistemas *middlewares* proprietários orientados a mensagem atendem aos respectivos requisitos. Como exemplo de protocolos, citam-se o T/TCP, NETBLT, VMTP e TCP, padrões IETF. Como exemplo de *middleware* orientado à mensagem cita-se o *MQ Series*. O *WebSphere*[12] *MQ Series* é um software aplicativo proprietário de mercado que implementa uma plataforma MOM (*Message Oriented Middleware*), ou seja, *middleware* orientado a mensagens que provê entrega confiável de mensagens (IBM, 2005b).

Convenciona-se não considerar para comparação os requisitos cujo resultado de verificação em ambas as EPMCs, foi considerado “não encontrado”.

4.3.1 Comparação quanto ao ponto de vista da empresa

1) Comparação quanto ao escopo, comunidade, federação, funções empresariais e procedimentos. A partir das Matrizes de Mapeamento de Requisitos referentes à verificação de cada EPMC, obtém-se a tabela 4.21 que apresenta a comparação qualitativa entre ambas quanto a estes requisitos do ponto de vista da empresa.

[12] Sua escolha no presente trabalho deu-se por ter sido esse o MOM oficialmente homologado pelo Banco Central do Brasil, para compor a infra-estrutura de mensagem confiável da arquitetura do SPB (Sistema de Pagamentos Brasileiro).

Tabela 4.21. Comparação entre as EPMCs quanto ao escopo, comunidade, federação, funções empresariais e procedimentos.

Requisito	***WS-Reliability***	***WS-ReliableMessaging***
O escopo consiste em prover interoperabilidade confiável.	O objetivo desta EPMC é prover um protocolo para a troca de mensagens com garantia de entrega, não duplicação e garantia de entrega ordenada de mensagens. Trata-se de uma especificação baseada em SOAP que atende aos requerimentos de entrega confiável de mensagens, críticos a algumas aplicações de *Web services.*	Esta EPMC tem por objetivo descrever um protocolo de interoperabilidade em *Web services*, que permite mensagens serem transferidas de forma confiável. Ela habilita um remetente de mensagem confiável a determinar precisamente a entrega de mensagem ao destinatário, permitindo resolver quaisquer dúvida de status relativo ao recebimento de uma mensagem transmitida.
Comunidade e federação são compostas por componentes de interoperabilidade.	Nesta EPMC, cada agente *Web services* produz e consome mensagens. Agentes se utilizam dos serviços de componentes específicos de interoperabilidades, chamados RMP (*Reliable Messaging Processor*) que têm a função de troca de mensagens entre si (vide Anexo A do presente trabalho). O RMP (*Reliable Messaging Processor*), componente de interoperabilidade, é a única parte envolvida na implementação do protocolo de mensagem confiável *WS-Reliability*.	Nesta EPMC cada agente *Web services* interage por meio de componentes de interoperabilidade, chamados de *RM Source* e *RM Destination* (vide Anexo C do presente trabalho). O *RM Source* é o componente que transmite mensagens de forma confiável a um *RM Destination*. O *RM Destination* é o componente que recebe mensagens transmitidas de forma confiável por um *RM Source*.
Componentes de interoperabilidade assumem funções empresariais distintas: remetente ou destinatário. Essas funções realizam procedimentos de envio e recebimento de mensagem, respectivamente.	Esta EPMC implementa um protocolo chamado RM (*Reliable Messaging*), referindo-se especificamente ao protocolo implementado por ela para entrega confiável de mensagens. Ele inclui cabeçalhos e interações específicas entre uma parte remetente e uma parte destinatária. O protocolo é implementado pelo RMP (*Reliable Messaging Processor*), que é o componente de interoperabilidade para troca confiável de mensagens. Com relação à transmissão de mensagem confiável, de um RMP para outro (vide Anexo A do presente trabalho), o primeiro é chamado de remetente de mensagens (*Sending RMP*) e o segundo é o destinatário de mensagens (*Receiving RMP*). Segundo esta EPMC, não há outros papéis para um RMP diferentes de RMP remetente ou RMP destinatário.	Esta EPMC ilustra o modelo de mensagem confiável (vide Anexo C do presente trabalho), onde representam-se duas funções possíveis entre componentes de interoperabilidade *Web services*, referenciados pelas notações: *RM Source* e *RM Destination*. Segundo essa EPMC, o *RM Source* é o componente que transmite mensagens de forma confiável a um *RM Destination*. O *RM Destination* é o componente que recebe mensagens transmitidas de forma confiável por um *RM Source*.

Como resultado comparativo, as duas EPMCs (*WS-Reliability* e *WS-ReliableMessaging*) indicam eqüidade com relação aos requisitos de escopo, comunidade, federação, funções empresariais e procedimentos.

2) Comparação quanto ao contrato. A partir das Matrizes de Mapeamento de Requisitos referentes à verificação de cada EPMC, obtém-se a tabela 4.22 que representa a comparação qualitativa entre ambas frente aos requisitos de contrato, para entrega confiável de mensagens.

Tabela 4.22. Comparação entre as EPMCs quanto ao contrato.

Requisito	***WS-Reliability***	***Ws-ReliableMessaging***
Entrega garantida de mensagem.	A EPMC *WS-Relibability* define um protocolo baseado em SOAP para a troca de mensagens com garantia de entrega. O atendimento a esse requisito é opcional.	A EPMC *WS-ReliableMessaging* define um protocolo interoperável que habilita a implementação de uma grande faixa de características de confiabilidade que inclui o recebimento garantido.
Entrega única de mensagem.	A EPMC *WS-Relibability* define um protocolo baseado em SOAP para a troca de mensagens com garantia de não duplicação de mensagens. O atendimento a esse requisito é opcional.	A EPMC *WS-ReliableMessaging* define um protocolo interoperável que habilita a implementação de uma grande faixa de características de confiabilidade que inclui eliminação de duplicação.
Entrega ordenada de mensagens.	A EPMC *WS-Relibability* define um protocolo baseado em SOAP para a troca de mensagens com garantia de entrega ordenada de mensagens. O atendimento a esse requisito é opcional.	A EPMC *WS-ReliableMessaging* define um protocolo interoperável que habilita a entrega de mensagens, por um componente destinatário, à sua respectiva aplicação destinatária, na ordem em que essas foram transmitidas pelo componente remetente.
Entrega íntegra de mensagem.	Não contemplado.	Não contemplado.

O comparativo entre as duas EPMCs (*WS-Reliability* e *WS-ReliableMessaging*) para requisitos de contratos, indica considerável eqüidade entre elas. Com exceção de um dos requisitos (integridade), as EPMCs indicam atender aos demais respectivos requisitos de contrato. Maiores considerações e eventuais aspectos divergentes identificados na presente comparação, são abordados a seguir.

- Flexibilidade no atendimento aos requisitos. Tanto a EPMC *WS-Reliability* quanto a EPMC *WS-ReliableMessaging*, indica atender aos requisitos de implementar propriedades de entrega garantida, entrega única e entrega ordenada. Entretanto, excepcionalmente na *WS-Reliability*, essas propriedades são opcionais, ou seja, passíveis de estarem individualmente habilitadas ou desabilitadas.

  Numa primeira visão sobre essa divergência, é natural questionar essa flexibilidade na EPMC *WS-Reliability*. Afinal, se as EPMCs devem atender aos requisitos de entrega garantida, entrega única e entrega ordenada, a flexibilidade de atender a esses requisitos de forma opcional pode expor a credibilidade desta EPMC em relação à outra EPMC, a *WS-ReliableMessaging* (que atende a todos esses requisitos, de forma invariante). Para abordagem a essa diferença, vale ressaltar que *Web services* são sistemas distribuídos que visam prover um modo padrão de interoperação entre diferentes aplicações de software que são executadas em uma variedade de estruturas computacionais. Os *Web services* suportam interoperabilidade máquina-a-máquina e são baseados em padrões abertos tais como SOAP, HTTP, XML e WSDL. Além disso, um objetivo de um sistema distribuído aberto é ser flexível, o que implica em prover fácil configuração de acordo com diferentes necessidades (TANENBAUM; STEEN, 2002; W3C, 2004a).

  Pelo exposto, embora ambas as EPMCs indiquem atender aos requisitos de entrega garantida, entrega única e entrega ordenada, a EPMC *WS-Reliability* apresenta-se mais aderente ao conceito de sistema distribuído aberto, por permitir flexibilidade no uso opcional dessas propriedades, de acordo com a necessidade de aplicação.

- Integridade de mensagens. Nenhuma das duas EPMCs trata integridade de mensagens, e delegam essa propriedade ao âmbito de segurança, a ser tratado por outro padrão em *Web services*.

A EPMC *WS-Reliability*, em sua especificação, recomenda fortemente que seja garantida a integridade fim-a-fim, do cabeçalho de mensagem, pelo uso de opções de segurança adequadas que são descritas em outra proposta de padrão, o *WS-Security* [13] . Ou seja, *WS-Reliabiltiy* define confiabilidade independentemente de segurança. Esse aspecto foi delegado, nesta EPMC, a outro grupo de estudos de novos padrões em *Web services*.

A EPMC *WS-ReliableMessaging*, em sua especificação, segue a mesma abordagem, ou seja, não faz assunções sobre requisitos de segurança das aplicações que utilizam entrega confiável. Segundo esta EPMC, se for necessária uma troca segura e íntegra de mensagens, o remetente e o destinatário deverão tratar aspectos de segurança.

De forma geral, integridade de dados é considerada uma disciplina de segurança da informação, sendo, portanto, razoável que estas EPMCs deleguem tal propriedade a outro padrão, específico de segurança. Além disso - como exemplo de outras implementações de sistemas orientados a mensagens - o *MQ Series* tem a mesma abordagem. Sendo um produto de software aplicativo proprietário com função de gerenciador de filas de mensagens que provê entrega confiável de mensagens, o *MQ Series* também delega aspectos de manutenção da integridade das mensagens trocadas entre suas partes, utilizando-se de uma infra-estrutura independente dele baseada em SSL - *Secure Sockets Layer* (IBM, 2005b).

Pelo exposto, é factível considerar que essas duas EPMCs considerem irrelevante tratar integridade no âmbito de entrega confiável de mensagens, delegando esse aspecto a outras frentes de padronização.

3) Comparação quanto às políticas. A partir das Matrizes de Mapeamento de Requisitos referentes à verificação de cada EPMC, obtém-se a tabela 4.23 que representa a comparação qualitativa entre ambas frente aos requisitos de políticas, para entrega confiável de mensagens.

[13] Mais informações sobre *WS-Security* em (OASIS, 2006d).

Tabela 4.23. Comparação entre as EPMCs quanto às políticas.

Requisito	*WS-Reliability*	*Ws-ReliableMessaging*
Independência de protocolo de transporte.	A EPMC *WS-Reliability* é definida como extensões de cabeçalho SOAP e é independente de protocolos subjacentes. Esta EPMC implementa um protocolo baseado em SOAP, agnóstico quanto ao transporte. Adicionalmente, há uma especificação de uso de método POST do HTTP para transporte de mensagens confiáveis.	A EPMC *WS-ReliableMessaging* define-se independente de transporte, permitindo que seja implementado utilizando-se diferentes tecnologias de transporte.
Suporte a agrupamento de mensagens.	Esta EPMC suporta agrupamento de mensagens. Segundo ela, uma mensagem confiável contém uma identificação de grupo que é globalmente único e uma mensagem sempre pertence a um grupo.	Esta EPMC suporta agrupamento de mensagens (nesta, chamado de seqüência). Durante o tempo de vida de uma seqüência, uma das invariantes requeridas é que o componente remetente deve atribuir a cada mensagem dentro de uma seqüência, um elemento de identificação de grupo.
A interação é unidirecional entre componentes.	Segundo esta EPMC, (vide Anexo B do presente trabalho), uma transmissão confiável de mensagem ocorre entre dois componentes, o primeiro é sempre chamado de remetente (*Sending RMP*) e o segundo é sempre chamado de destinatário (*Receiving RMP*). O remetente é a parte que envia mensagem e o destinatário é a parte que recebe mensagens. Cabe a esse último somente retornar um reconhecimento da mensagem recebida.	Nesta EPMC, o componente remetente (*RM Source*) é aquele que transmite mensagens de forma confiável a um componente destinatário (*RM Destination*). Segundo esta EPMC, as seqüências de mensagens, com conteúdo de aplicação, são transmitidas sempre de um remetente para um destinatário. As mensagens em sentido contrário são reconhecimentos (vide Anexo C do presente trabalho).
Deve-se atender aos requisitos de contrato.	Segundo verificado, essa EPMC indica atender a todos os requisitos de contrato, exceto a integridade de mensagem.	Segundo verificado, essa EPMC indica atender a todos os requisitos de contrato, exceto a integridade de mensagem.

O comparativo entre as duas EPMCs (*WS-Reliability* e *WS-ReliableMessaging*), para cada requisito de políticas, indica eqüidade entre elas. Considerações em maiores detalhes sobre os resultados comparativos são abordados abaixo:

- Atendimento aos requisitos de contrato. De forma geral, e como visto anteriormente, as duas EPMCs indicam atender a três dos quatro requisitos

de contratos. O único requisito não atendido por ambas é o de integridade. Entretanto, pelo exposto anteriormente, é razoável que essas EPMCs deleguem essa propriedade a outro padrão *Web services*, específico de segurança, uma vez que integridade é considerada uma das disciplinas de segurança da informação (BASS; CLEMENS; KAZMAN, 2003).

- Independência de protocolo de transporte. No requisito de independência de protocolo de transporte, ambas as EPMCs são especificadas como extensões de SOAP, em sintaxes usando estrutura XML, independentes (sem vínculo técnico), podendo-se utilizar quaisquer protocolos para seu transporte (ex: HTTP, SMTP e FTP).

  Particularmente, a EPMC *WS-Reliability* tem um diferencial. Apesar de essa ser independente de protocolo, ela apresenta adicionalmente a especificação de uso dos métodos do protocolo HTTP para transporte de mensagens SOAP (No caso, mensagens SOAP estendidas conforme a EPMC *WS-Reliabiltiy* para implementar entrega confiável, sendo transportadas sobre HTTP). Trata-se de uma iniciativa pró-ativa e sem compromisso com o uso exclusivo desse protocolo, visando à suportar os desenvolvedores. Esta iniciativa baseia-se no fato de que o HTTP é o mais provável de ser usado em implementações de *Web services*, no transporte de mensagens, uma vez que esse é o protocolo mais difundido na Internet (KUROSE; ROSS, 2001).

- Sentido de envio de mensagens entre componentes. Segundo a WSA, uma interação de troca de mensagens é ponto-a-ponto e unidirecional, envolvendo duas entidades: o remetente, que envia a mensagem; e, o destinatário, que recebe a mensagem (W3C, 2004c, 2004e).

  Na comparação das EPMCs *WS-Reliabiltiy* e *WS-ReliableMessaging* para o requisito de sentido de interação, conforme tabela 4.23, segundo o qual uma mensagem é enviada do componente remetente para o componente destinatário, tem-se que ambas EPMCs utilizam este conceito. Na EPMC *WS-Reliability*, com relação à transmissão confiável de uma mensagem de um componente (nesta chamado de RMP – *Reliable Messaging Processor*) a

outro, o primeiro é sempre chamado de remetente (*Sending RMP*) e o segundo é sempre chamado de destinatário (*Receiving RMP*). Nesta EPMC, a mensagem é transmitida do componente *Sending RMP* para o componente *Receiving RMP*, sendo que esse último retorna um reconhecimento. Na EPMC *WS-ReliableMessaging* tem-se a mesma abordagem, apenas com a diferença de que os nomes dos componentes remetente e destinatário, nesta EPMC, são *RM Source* e *RM Destination*, respectivamente.

Analogamente, alguns protocolos de entrega confiável de dados, como os protocolos VMTP e NETBLT, implementam comunicação unidirecional. Da mesma forma em sistemas aplicativos *middlewares* proprietários orientados a mensagem, como o *MQ Series*, também adota-se comunicação unidirecional entre componentes, no processo de envio de mensagem. No *MQ Series*, particularmente, o componente de software que manipula envio e recebimento de mensagens é chamado de MCA (*Message Channel Agent*). Neste caso, mensagens são transmitidas entre esses componentes, através de um canal que é uma ligação lógica de comunicação unidirecional entre dois MCAs (IBM, 2005a, 2005c).

- Suporte a grupos de mensagens. Quanto a este requisito, segundo a WSA, as mensagens devem poder ser agrupadas. O conceito de grupo de mensagens tem a função de uma mensagem poder ser associada a um âmbito de negócio. Para tal, seu cabeçalho deve conter um parâmetro que identifique o grupo ao qual a mensagem pertence.

  Na comparação das EPMCs acima, ambas suportam agrupamento de mensagens e para tal utilizam parâmetros de identificação de grupo no cabeçalho. Analogamente, protocolos ditos de entrega confiável como o T/TCP e *middlewares* proprietários como o *MQ Series* também aplicam esse conceito de suporte ao agrupamento de mensagens.

4.3.2 Comparação quanto ao ponto de vista da informação

4.3.2.1 Comparação quanto ao esquema invariante

A partir das Matrizes de Mapeamento de Requisitos referentes à verificação de cada EPMC, obtém-se a tabela 4.24 que representa a comparação qualitativa entre ambas frente aos requisitos de esquema invariante, para entrega confiável de mensagens.

Tabela 4.24. Comparação entre as EPMCs quanto ao esquema invariante.

Requisito	***WS-Reliability***	***WS-ReliableMessaging***
A interação é baseada em mensagem.	Esta EPMC define um protocolo para intercâmbio de mensagens SOAP. Esse protocolo inclui cabeçalhos de mensagem específicos para permitir interoperabilidade com mensagem confiável entre uma entidade remetente e uma entidade destinatária. O objeto informação é uma mensagem SOAP.	Esta EPMC corresponde a um protocolo que permite mensagens SOAP serem transferidas de forma confiável entre nós que implementem este protocolo. O objeto informação é uma mensagem SOAP.
Mensagens possuem controle de integridade no cabeçalho.	Não contemplado.	Não contemplado.
Mensagens possuem numeração de seqüência no cabeçalho.	Nesta EPMC, o número de seqüência, quando presente, é um inteiro que é único dentro de um grupo. Neste caso, o componente remetente inclui o elemento <*SequenceNum*> no cabeçalho de todas as mensagens confiáveis de um grupo que tenha mais de uma mensagem. Esse elemento <*SequenceNum*> contém o número de seqüência que identifica a ordem da mensagem dentro de seu grupo. Ele é utilizado para suportar a ordenação de mensagens. O RMP, componente de interoperabilidade remetente, inicializa esse parâmetro com 0 (zero) para a primeira mensagem. A partir dai, incrementa-se seu valor de 1 (um) para cada mensagem adicional do mesmo grupo, a ser enviada.	Nesta EPMC, durante o tempo de vida de uma seqüência, dentre as invariantes requeridas para correta operação do protocolo, o *RM Source* (componente de interoperabilidade remetente), deve atribuir a cada mensagem dentro de uma seqüência, um número de mensagem. No caso, o componente remetente, atribui a cada mensagem dentro de uma seqüência um elemento <*MessageNumber*> que é incrementado de 1, a partir de um valor inicial igual a 1.

(Continua)

(Continuação)

Requisito	*WS-Reliability*	*WS-ReliableMessaging*
Mensagens possuem identificação única.	Nesta EPMC, uma mensagem contém um parâmetro identificador que é globalmente único. Esse identificador de mensagem confiável é uma combinação do identificador de grupo e do número de seqüência da mensagem dentro desse grupo.	Nesta EPMC, uma mensagem contém um identificador que é globalmente único. Esse identificador de mensagem confiável é uma combinação do identificador de grupo e do número de seqüência da mensagem dentro desse grupo.
Mensagens possuem parâmetro de identificação de grupo de mensagens no cabeçalho.	Nesta EPMC, uma mensagem confiável sempre pertence a um grupo. O ID de grupo é um identificador de grupo único global, representado pelo elemento <*GroupId*>, no cabeçalho. Esse elemento contém uma URI (*Uniform Resource Identifier*) que identifica unicamente um grupo de mensagem.	Esta EPMC denomina um grupo como "seqüência". No cabeçalho de cada mensagem, introduz-se o bloco *Sequence*, que possui o elemento <*Identifier*>. Esse elemento contém uma URI (*Uniform Resource Identifier*) que identifica unicamente um grupo de mensagem.
Mensagens possuem parâmetro de marcas de tempo.	Nesta EPMC, cada mensagem pode conter um ou mais elementos de expiração. Esses elementos são o <*ExpiryTime*>, o <*GroupExpiryTime*> e o <*GroupMaxIdleDuration*>. Esses elementos são opcionais. Os dois primeiros são do tipo *xs:DateTime* [14] com conteúdo de acordo com o UTC (*Universal Time Coordinated*). O último deles é do tipo *xs:Duration*[15].	Nesta EPMC, o conceito de expiração está somente relacionado com um grupo (uma seqüência). No processo de iniciação de uma seqüência, um dos elementos que compõem a sintaxe do *CreateSequence* é o <*wsrm:expires*>. Esse elemento é do tipo *xs:Duration* e especifica a duração requerida pelo *RM Source* (componente de interoperabilidade remetente), para uma seqüência. As mensagens trocadas não possuem marcas de tempo.

No comparativo entre as EPMC´s (*WS-Reliability* e *WS-ReliableMessaging*), para cada requisito de esquema invariante, identificam-se algumas particulariedades quanto ao modo como cada uma visa a atender ao respectivo requisito. Essas são descritas a seguir:

- Interação baseada em mensagem. Uma mensagem de *Web Services*, também chamada de mensagem SOAP, é a unidade básica de comunicação em *Web services* (W3C, 2004a, 2004f).

[14] *DateTime* é um tipo de dados primitivo de XML que representa um instante específico de tempo. O espaço de valores desse tipo de dados é o espaço de combinações de datas e horas. definidos pela norma ISO 8601, de 1988 (W3C, 2001).
[15] *Duration* é um tipo de dados primitivo de XML que representa uma duração de tempo, com seis dimensões (ano, mês, dia, hora, minutos e segundos), de acordo com a norma ISO 8601, de 1988 (W3C, 2001).

Na comparação das EPMCs *WS-Reliability* e *WS-ReliableMessaging*, para o requisito de que a interação deve ser baseada em mensagens, tem-se que ambas as EPMCs verificadas se baseiam na extensibilidade de mensagem SOAP. Para tal, ambas propõem inclusão de parâmetros no cabeçalho, bem como modelos de interação orientados à troca de mensagens entre *Web services*. Em suma, em ambas EPMCs, o objeto de informação enviado de uma entidade à outra, de forma confiável, é uma mensagem SOAP.

- Mensagens possuem integridade. Na comparação das EPMCs *WS-Reliability* e *WS-ReliableMessaging*, para o requisito de que mensagens possuem controle de integridade, tem-se que ambas as EPMCs verificadas, não indicam implementação dessa propriedade. Como abordado anteriormente, ambas delegam essa propriedade ao âmbito de segurança, a ser tratado por outro padrão em *Web services*.

- Mensagens possuem numeração de seqüência. Na comparação das EPMCs *WS-Reliability* e *WS-ReliableMessaging*, para o requisito de que mensagens possuem numeração de seqüência tem-se que, em ambas as EPMCs verificadas, cada mensagem possui um número de seqüência dentro do grupo ao qual ela pertence. Na *WS-Reliability*, esse elemento é o <*SequenceNum*> e na *WS-ReliableMessaging*, esse elemento é o <*MessageNumber*>.

  Entretanto, algumas diferenças entre essas EPMCs aparecem quanto à iniciação de contagem, à capacidade de mensagens por grupo e ao modo como tratam quando atingem o limite de mensagens por grupo.

  Na EPMC *WS-Reliability*, o elemento <*SequenceNum*> é iniciado com valor 0 (zero), para a primeira mensagem de um grupo, sendo incrementado de exatamente 1 (um), a cada mensagem. Seu limite é de 2^64 mensagens por grupo. Quando esse limite é atingido (esse evento chama-se *rollover*), o remetente inicia um novo grupo, automaticamente. Além disso, nesta EPMC, o elemento <*SequenceNum*> é opcional.

Na EPMC *WS-ReliableMessaging*, o elemento <*MessageNumber*> é iniciado com valor 1 (um), para a primeira mensagem de um grupo, sendo incrementado de exatamente 1 (um), a cada mensagem. Seu limite é de 2^63 mensagens por grupo. Quando esse limite é atingido, o grupo é considerado finalizado, sendo necessário o remetente solicitar a criação de novo grupo, enviando um <*CreateSequence*> ao destinatário.

Numa primeira visão sobre essas diferenças, a EPMC *WS-ReliableMessaging* apresenta vantagem em relação à outra EPMC. Em muitas linguagens de programação o valor padrão associado a uma variável não explicitamente inicializada é zero (ex: variável *long integer* em Java). Na implementação da EPMC *WS-ReliableMessaging*, esse valor necessita ser explicitamente atualizado para 1 (um), no elemento <*MessageNumber*>, uma vez que zero é um valor inválido para esse elemento. Isso elimina ambigüidades quanto ao fato de se a variável de contagem de mensagem foi inicializada ou não.

Um outro diferencial da EPMC *WS-ReliableMessaging* é o fato de que, ao contrário da EPMC *WS-Reliability*, seu elemento de numeração de seqüência <*MessageNumber*> é mandatório.

Como exemplo de como um produto de mercado, *middleware* proprietário orientado a mensagem, implementa numeração de seqüência de mensagens, o *MQ Series* tem uma abordagem similar à EPMC *WS-ReliableMessaging*. Segundo esse, todas as mensagens dentro de um grupo são numeradas, utilizando-se o elemento <*MsgSeqNumber*> em seu cabeçalho. Este elemento <*MsgSeqNumber*> representa o número de seqüência de uma mensagem dentro de um grupo. Tal como na EPMC *WS-ReliableMessaging*, ele é mandatório, é sempre inicializado com valor 1 (um) e incrementado de exatamente 1 (um) a cada mensagem subseqüente. O limite de mensagens suportado por grupo no *MQ Series* é de 1x10^9 mensagens (IBM, 2005a).

- Mensagens possuem identificação de grupo de mensagens. Na comparação das EPMCs *WS-Reliability* e *WS-ReliableMessaging*, para o requisito de que mensagens possuem parâmetro de identificação de grupo de mensagens tem-se

que ambas as EPMCs verificadas utilizam elementos de parametrização no cabeçalho para a identificação do grupo ao qual a mensagem pertence. No caso da EPMC *WS-Reliability*, esse elemento chama-se <*GroupID*>. No caso da EPMC *WS-ReliableMessaging*, esse elemento chama-se <*Identifier*>.

Em ambos os casos comparados, o parâmetro de identificação de grupo é do tipo URI (*Uniform Resource Identifiers*) em conformidade com a RFC 2396. Esta abordagem das duas EPMCs está alinhada com o W3C. Segundo o W3C, deve-se utilizar URIs para identificar recursos *Web services*.

- Mensagens possuem identificação única. Na comparação entre as EPMCs *WS-Reliabiltiy* e *WS-ReliableMessaging*, segundo o requisito de presença de parâmetro identificação único de mensagem, tem-se que ambas suportam esse requisito, embora não tenham um elemento exclusivo para esse parâmetro em sua estrutura de cabeçalho. Para ambas as EPMCs, a identificação única de mensagem é obtida a partir da concatenação do elemento que possui a identificação do grupo ao qual ela pertence (esse é baseado em URI), com o número de seqüência dessa mensagem, dentro do respectivo grupo.

  Apenas ressalta-se que a EPMC *WS-Reliability*, em particular, define seu elemento de número de seqüência <*SequenceNum*> como opcional no cabeçalho de uma mensagem. Esta flexibilidade específica da EPMC *WS-Reliability*, de que é opcional o uso de número de seqüência, pode, neste caso, inviabilizar a identificação única de mensagem.

  Analogamente a essas EPMCs, em protocolos ditos de entrega confiável, uma forma mais consistente de detecção de duplicidade de um pacote de dados, é obtida pela inclusão de um parâmetro de identificação única para o pacote (IREN; AMER; CONRAD, 1999).

  No protocolo T/TCP (RFCs 1379 e 1644), um pacote tem sua identificação única pela combinação de dois elementos: o contador de conexão (uma espécie de identificação de grupo, que mantém uma persistência entre as partes, referente a uma sessão de transmissão); e o numerador de seqüência, que provê a

contagem do pacote, dentro da respectiva conexão. Também, em sistema proprietário de *middleware* orientado a mensagens, *MQ Series*, cada mensagem possui um parâmetro de identificação exclusivo, que é o <*MsgId*>.

- Mensagens possuem marcas de tempo. A importância da associação de marcas de tempo às mensagens, está em permitir não só o estabelecimento da ordem cronológica dessas dentro de um grupo de mensagens, mas também em identificar mensagens que estejam expiradas.

  Na comparação das EPMCs *WS-Reliability* e *WS-ReliableMessaging*, para o requisito de que mensagens possuem marcas de tempo tem-se que ambas as EPMCs verificadas contemplam o conceito de marcas de tempo. Entretanto, com algumas diferenças a seguir abordadas.

  Na EPMC *WS-Reliability*, uma mensagem pode possuir até três elementos distintos de marcas de tempo, no cabeçalho. O primeiro deles <*ExpiryTime*>, estabelece a data e hora após a qual essa mensagem não poderá mais ser entregue, pois expirou; o segundo <*GroupExpiryTime*>, estabelece a data e hora após a qual o grupo de mensagens, ao qual essa mensagem pertence, deixará de ser válido; por último, <*GroupMaxIdleDuration*> estabelece o tempo limite, decorrido desde a última mensagem enviada ou recebida em um grupo, após o qual esse grupo pode ser terminado. Os dois primeiros elementos são do tipo *xs:DateTime* e expressam tempo absoluto, baseado no UTC (*Universal Time Coordinated*). Portanto é necessário que ambos os componentes de interoperabilidade estejam em sincronismo de relógio por meio de um serviço externo. O último, do tipo *xs:Duration*, é expresso como tempo relativo em segundos.

  Na EPMC *WS-ReliableMessaging*, as mensagens trocadas não possuem marcas de tempo. Esse aspecto é contemplado somente no processo de iniciação do grupo (nessa EPMC chamado de criação de seqüência). Durante o processo de iniciação de um grupo, o componente remetente envia uma solicitação <*CreateSequence*> ao destinatário, informando qual será o tempo de duração dessa seqüência. Esse tempo de duração, do tipo *XS: Duration*, expresso como

tempo relativo, especifica o intervalo de tempo após o qual esse grupo será considerado inválido e quaisquer mensagens, pertencentes a esse, que cheguem ao destinatário serão desconsideradas.

Nota-se portanto que, na EPMC, *WS-Reliability*, o destinatário pode determinar se a mensagem expirou, apenas comparando-se a data e horário correntes, por meio do UTC, com o expresso no cabeçalho da mensagem. Por outro lado, na EPMC *WS-ReliableMessaging*, as mensagens não possuem marcas de tempo e é necessário manter uma persistência de informação referente ao tempo validade do grupo da mensagem, durante toda a duração desse grupo.

Analogamente, e como exemplo de produto proprietário que implementa *middleware* orientado a mensagem, o *MQ Series*, observa-se uma abordagem similar à EPMC *WS-Reliability*. Nesse sistema proprietário, todas as mensagens possuem marcas de tempo. Cada mensagem entregue a esse, para ser enviada de um remetente a um destinatário, recebe três parâmetros de expiração: <*PutDate*> e <*PutTime*> que representam, respectivamente, a data e a hora de entrada no sistema de mensagem; bem como, <*Expiry*> quantidade de tempo de sua validade (a partir dessa data e hora em que a mensagem foi inserida na fila). Neste caso, remetente e destinatário devem estar sincronizados a um serviço de tempo baseado no UTC (*Universal Time Coordinated*) e não necessitam guardar informações de persistência para determinação quanto a uma mensagem estar expirada ou não (IBM, 2005a, 2005b).

4.3.2.2 Comparação quanto ao esquema estático

A partir das Matrizes de Mapeamento de Requisitos referentes à verificação de cada EPMC, obtém-se a tabela 4.25 que representa a comparação qualitativa entre ambas frente aos requisitos de esquema estático, para entrega confiável de mensagens.

Na comparação entre as EPMCs *WS-Reliabiltiy* e *WS-ReliableMessaging*, segundo os estados possíveis de um objeto informação (mensagem de interação ou mensagem de reconhecimento), tem-se que ambas possuem essa propriedade. Na

EPMC *WS-Reliability*, o objeto mensagem pode ser de conteúdo específico de aplicação (chamado *ReliableMessage*) ou de reconhecimento, (chamado *RM-Reply*). Na EPMC *WS-ReliableMessaging*, o objeto mensagem pode ser de conteúdo específico de aplicação (chamado *Message*) ou de reconhecimento, (chamado *SequenceAcknowledgement*).

Tabela 4.25. Comparação entre as EPMCs quanto ao esquema estático.

Requisito	***WS-Reliability***	***WS-ReliableMessaging***
Estados possíveis da mensagem: "mensagem de interação" ou "mensagem de reconhecimento".	Nessa EPMC, o objeto informação trocado entre dois componentes de interoperabilidade (*Sending RMP* e *Receiving RMP*) pode ser: mensagem de aplicação (*ReliableMessage*) ou mensagem de reconhecimento (*RM-Reply*). Vide Anexo B do presente trabalho.	Nessa EPMC, o objeto informação trocado entre dois componentes de interoperabilidade (*RM Surce* e *RM Destination*) pode ser: mensagem de aplicação (*Message*) ou mensagem de reconhecimento (*SequenceAcknowledgement*). Vide Anexo D do presente trabalho.

Analogamente, em protocolos de transporte ditos de entrega confiável, o objeto informação também tem comportamento similar. Nesse caso, a comunicação entre pares de entidades é feita por meio de TPDUs (*Transport Protocol Data Units*), que assumem dois tipos: informação e controle. As TPDUs de informação transportam dados e as TPDUs de controle são utilizadas para reconhecimento (positivo e/ou negativo, mensagens de erro, etc.). Também na área de *middlewares* proprietários orientados a mensagem, no *MQ Series*, têm-se as mensagens de conteúdo específico de aplicação e as de reconhecimento (essas chamadas de *report*). O pedaço de informação que é trocado entre entidades remetente e destinatária é considerado uma mensagem. Embora a infra-estrutura de mensagem não precise ter entendimento sobre a informação que está sendo transmitida, seu único compromisso é assegurar que a mensagem será corretamente entregue no destino. Em resposta a uma mensagem com conteúdo específico de aplicação, o componente destinatário (*Message Channel Agent*), do *MQ Series*, pode retornar mensagens de reconhecimento. Uma mensagem de reconhecimento é chamada de confirmação de chegada (COA – *Confirmation of Arrival*) e indica a um remetente que uma mensagem chegou e foi aceita pelo destinatário (IBM, 2005b, 2005c).

4.3.2.3 Comparação quanto ao esquema dinâmico

O esquema dinâmico, inerente ao ponto de vista da informação, representa ações, sobre as mensagens. A partir das Matrizes de Mapeamento de Requisitos referentes à verificação de cada EPMC, obtém-se a tabela 4.26 que representa a comparação qualitativa entre ambas, frente aos requisitos de esquema dinâmico, para entrega confiável de mensagens.

Tabela 4.26. Comparação entre as EPMCs quanto ao esquema dinâmico.

Requisito	***WS-Reliability***	***WS-ReliableMessaging***
Atividades do remetente para enviar mensagem: preparação e envio.	Para se enviar uma mensagem, segundo esta EPMC, o componente de interoperabilidade remetente estende o cabeçalho SOAP, inserindo um bloco com prefixo *wsrm*. Nessa estrutura estendida, são inseridos os elementos <*GroupID*> e <*SequenceNum*> para a identificação do grupo e do número de seqüência de mensagem, respectivamente. Além disso, são inseridos elementos de tempo de expiração de mensagem <*ExpiryTime*>, de expiração de grupo <*GroupExpiryTime*> e o <*GroupMaxIdleDuration*>. Todas as mensagens de conteúdo específico de aplicação possuem marcas de tempo. Não há elementos de controle de integridade.	Para se enviar uma mensagem, segundo esta EPMC, o componente de interoperabilidade remetente estende o cabeçalho SOAP, inserindo um bloco com prefixo *wsrm*. Nessa estrutura estendida, são inseridos os elementos <*Identifier*> e <*MessageNum*> para a identificação do grupo e do número de seqüência de mensagem, respectivamente. Nessa EPMC, não se inserem marcas de tempo em cabeçalhos de mensagens. Não há elementos de controle de integridade.
Atividades do destinatário ao receber mensagem: checagem.	Ao receber uma mensagem, segundo esta EPMC, o componente de interoperabilidade destinatário entrega a mensagem à respectiva aplicação assim que checar se a mensagem está apta a ser entregue à aplicação. Mensagens fora de ordem ou expiradas são rejeitadas.	Ao receber uma mensagem, segundo esta EPMC, o componente de interoperabilidade destinatário entrega a mensagem à respectiva aplicação assim que checar se a mensagem está apta a ser entregue à aplicação. Mensagens fora de ordem podem ser aceitas.
Reconhecimento de mensagem.	Nesta EPMC, o destinatário envia reconhecimento para o remetente, indicando que uma mensagem foi aceita (reconhecimento positivo).	Nesta EPMC, o destinatário envia reconhecimento para o remetente, indicando que uma mensagem foi aceita (reconhecimento positivo) ou não (reconhecimento negativo).

No comparativo entre as EPMC´s (*WS-Reliability* e *WS-ReliableMessaging*), para cada requisito do esquema dinâmico, nota-se relativa equivalência entre elas, entretanto com algumas características que diferenciam o modo como cada uma visa a atender aos respectivos requisitos. Considerações em maiores detalhes sobre os resultados comparativos seguem abaixo:

- Preparação e envio de mensagem. Como proposto no Modelo de Referência, para o esquema dinâmico, quando um componente de interoperabilidade remetente é solicitado a enviar mensagens (pelas camadas de aplicação em seu agente), ele, antes de enviar ao destinatário, prepara a mensagem adicionando alguns elementos de parametrização no cabeçalho, tais como número de seqüência de mensagem, parâmetro de detecção de erros de intergridade, identificação única de mensagem, identificação de grupo de mensagem e marca de tempo.

  Na comparação entre as EPMCs *WS-Reliabiltiy* e *WS-ReliableMessaging*, segundo as atividades realizadas por um componente de interoperabilidade remetente para envio de uma mensagem, tem-se que ambas têm características semelhantes, apresentando poucas diferenciações, essas descritas a seguir.

  Na EPMC *WS-Reliability*, ao obter uma mensagem a ser enviada, proveniente da aplicação no seu respectivo agente remetente, o componente de interoperabilidade do remetente (nesta EPMC, chamado de *Sending RMP*) estende a estrutura do cabeçalho SOAP, incluindo um *namespace*[16] com prefixo *wsrm*[17]. Nesta estrutura insere-se o bloco <*MessageID*>, contendo os parâmetros de identificação de grupo e de número de seqüência de mensagem. São eles: o elemento <*GroupID*> que identifica o grupo com uma URI (em conformidade com a RFC 2396) e o elemento <*SequenceNum*> que contém o número de seqüência da mensagem nesse grupo. Além disso, são inseridos elementos de marca de tempo, referentes à expiração de mensagem <*ExpiryTime*>, bem como à

[16] Um *namespace* XML é uma coleção de nomes, propostos por órgãos de padronização que são utilizados em documentos XML. Prefixos XML de *namespaces* são utilizados para definir, dentro de uma estrutura XML, um escopo específico de tipos de elementos.
[17] O *namespace* reservado para extensibilidade do SOAP, visando a suporte de mensagem confiável, foi proposto pela OASIS e seu prefixo é *wsrm* (que é um acrônimo para *Web Services Reliable Messaging*).

expiração de grupo, com os elementos <*GroupExpiryTime*> e <*GroupMaxIdleDuration*>.

Na EPMC *WS-ReliableMessaging*, ao obter uma mensagem a ser enviada, proveniente da aplicação no seu respectivo agente remetente, o componente de interoperabilidade do remetente (nesta EPMC, chamado de *RM Source*) estende a estrutura do cabeçalho SOAP, incluindo um *namespace* com prefixo *wsrm*. Nesta estrutura insere-se o bloco <*Sequence*>, contendo os identificadores de grupo e de número de seqüência de mensagem. São eles, respectivamente: o elemento <*Identifier*> que identifica o grupo com uma URI (em conformidade com a RFC 2396) e o elemento <*MessageNumber*> que contém o número de seqüência da mensagem nesse grupo. Nesta EPMC, não são inseridas marcas de tempo em cabeçalhos de mensagens.

Em ambas as EPMCs, parâmetros de detecção de erros de integridade não são inseridos. Além disso, somente na EPMC *WS-Reliability*, ao contrário da EPMC *WS-ReliableMessaging*, todas as mensagens de conteúdo específico de aplicação possuem marcas de tempo.

Analogamente, em protocolos ditos de entrega confiável, ao receber uma APDU (*Application Protocol Data Unit*), da camada de aplicação, o componente de transporte insere um cabeçalho de controle de transporte, formando a TPDU (*Transport Protocol Data Unit*), para comportar elementos de controle (seqüência, marcas de tempo, etc.). Da mesma forma, em *middlewares* proprietários orientados a mensagem, como o *MQ Series*, para se transportar uma mensagem ao seu destino, informações extras devem ser adicionadas enquanto ela passa pela infra-estrutura de mensagens. No *MQ Series*, o cabeçalho da mensagem recebe elementos tais como o número de seqüência de mensagem <*MsgSeqNumber*>, o identificador único de mensagem <*MsgId*>, o identificador de grupo de mensagem <*GroupId*> e marcas de tempo. Para essas, o *MQ Series* utiliza três elementos: <*PutDate*> e <*PutTime*> que representam, respectivamente, a data e a hora de entrada no sistema de mensagem; bem como <*Expiry*> que indica a quantidade de tempo de sua validade.

- Checagem de mensagem. Como definido no Modelo de Referência, quando o componente destinatário recebe uma mensagem, ele deve realizar atividades de checagem da mensagem, antes de entregá-la às camadas superiores de aplicação em seu agente *Web services*. Nessa checagem, verifica-se quanto à mensagem recebida estar íntegra, estar em seqüência, ser válida, não ser repetida; e, pertencer a um grupo válido de mensagens (no caso de não ser a primeira mensagem de um grupo).

  Na comparação entre as EPMCs *WS-Reliabiltiy* e *WS-ReliableMessaging*, segundo as atividades realizadas por um componente de interoperabilidade destinatário ao receber uma mensagem, tem-se que ambas são semelhantes na semântica de verificação e tratamento das mensagens recebidas, com algumas poucas diferenciações, descritas a seguir.

  Tanto na EPMC *WS-Reliability*, quanto na EPMC *WS-ReliableMessaging*, para cada mensagem recebida, o componente de interoperabilidade destinatário entrega a mensagem à respectiva aplicação, assim que realizar checagem sobre ela (ex: mensagem pertence a um grupo válido e não está duplicada). Em ambas as EPMCs a checagem de interidade não é realizada.

  Como diferenças encontradas, entre as EPMCs, na EPMC *WS-Reliability* não são aceitas as mensagens recebidas pelo destinatário, que estejam fora de ordem. Entretanto, na EPMC *WS-ReliableMessaging*, mensagens que chegam fora de ordem ao destinatário, pertencentes a um grupo válido podem ser aceitas. Isso se deve ao fato de a EPMC *WS-ReliableMessaging* utilizar modo de interação baseado em repetição seletiva.

  Além disso, na EPMC *WS-Reliability*, o componente destinatário checa se a mensagem não está expirada, caso contrário ela é descartada. Na EPMC *WS-ReliableMessaging*, esse tipo de controle não existe para cada mensagem recebida, uma vez que essa EPMC não utiliza marcas de tempo em cada mensagem.

Analogamente, em protocolos ditos de entrega confiável, ao receber uma TPDU (*Transport Protocol Data Unit*), vinda do remetente, o componente destinatário checa o cabeçalho quanto aos elementos de controle (seqüência, marcas de tempo, etc.), antes de entregá-la à camada de aplicação. Da mesma forma, em *middleware* orientado à mensagem, no *MQ Series,* ao receber uma mensagem, seu MCA (*Message Channel Agent*) destinatário checa as informações extras que acompanham a mensagem. No cabeçalho de uma mensagem transmitida pelo *MQ Series* existem elementos tais como o número de seqüência de mensagem <*MsgSeqNumber*>, identificador único de mensagem <*MsgId*>, identificador de grupo de mensagem <*GroupId*> e marcas de tempo que precisam ser checadas pelo destinatário, antes de serem disponibilizadas para a aplicação.

- Reconhecimento de mensagem. Ambas as EPMCs verificadas, se baseiam em esquemas de reconhecimento. Nelas, ao receber e aceitar uma mensagem, o componente destinatário envia uma resposta ao componente remetente, correspondendo a um aceite dessa mensagem. Na EPMC *WS-ReliableMessaging*, particularmente, também é possível indicar a rejeição de uma mensagem, enviando um reconhecimento negativo (NACK) ao remetente.

  Analogamente, tanto os protocolos considerados como “de entrega confiável”, quanto *middlewares* proprietários orientados a mensagem, como o *MQ Series*, realizam essa atividade de reconhecimento após o aceite de uma mensagem.

4.3.3 Comparação quanto ao ponto de vista da computação

4.3.3.1 Comparação quanto ao tipo de interface

A partir das Matrizes de Mapeamento de Requisitos referentes à verificação de cada EPMC, obtém-se a tabela 4.27 que representa a comparação qualitativa entre ambas frente ao requisito de tipo de interface, para entrega confiável de mensagens.

Tabela 4.27. Comparação entre as EPMCs quanto ao tipo de interface.

Requisito	***WS-Reliability***	***WS-ReliableMessaging***
A interface do componente de interoperabilidade é do tipo "operação", indicando uma relação cliente/servidor.	Nesta EPMC, entrega confiável ocorre entre duas entidades que são, respectivamente, um produtor de mensagens (chamado de *Sending RMP*) e um consumidor de mensagens (chamado de *Receiving RMP*). No seu modelo de interação, as mensagens são enviadas do remetente ao destinatário, sendo que o destinatário deve responder com reconhecimento (vide Anexos A e B do presente trabalho). O objeto informação trocado entre as partes são mensagens SOAP.	Nesta EPMC, entrega confiável ocorre entre duas entidades que são, respectivamente, um produtor de mensagens (chamado de *RM Source*) e um consumidor de mensagens (chamado de *RM Destination*). No seu modelo de interação, as mensagens são enviadas do remetente ao destinatário, sendo que o destinatário deve responder com reconhecimento (vide Anexos C e D do presente trabalho). O objeto informação trocado entre as partes são mensagens SOAP.

Na EPMC *WS-Reliability*, a troca confiável de mensagens ocorre entre duas entidades. Essas entidades são, respectivamente, um produtor de mensagens e um consumidor de mensagens. As mensagens são envidas do produtor para o consumidor via seus componentes de interoperabilidade, chamados de RMP (*Reliable Messaging Processors*).

A EPMC *WS-ReliableMessaging* define um protocolo interoperável que habilita ao componente remetente de mensagem confiável (*RM Source*) determinar precisamente a entrega de uma mensagem que ele transmite a um componente destinatário (*RM Destination*). No caso, o componente remetente faz solicitações ao destinatário (por exemplo: solicitação de criação de seqüência e solicitação de término de seqüência).

O objeto informação trocado entre as duas partes (remetente e destinatário), nessas duas EPMCs é baseado em mensagens SOAP 1.1.

Pelo exposto e verificado, ambas as EPMCs realizam interações que caracterizam uma interface de operação.

4.3.3.2 Comparação quanto ao modo de interação

A partir das Matrizes de Mapeamento de Requisitos referentes à verificação de cada EPMC, obtém-se a tabela 4.28 que representa a comparação qualitativa entre ambas frente aos requisitos quanto ao modo de interação, para entrega confiável de mensagens.

Tabela 4.28. Comparação entre as EPMCs quanto ao modo de interação.

Requisito	***WS-Reliability***	***WS-ReliableMessaging***
Utiliza-se reconhecimento.	Esta EPMC utiliza reconhecimento positivo (ACK). Segundo ela, implementa-se entrega garantida de mensagem, por meio de seu protocolo que utiliza indicações de reconhecimento. Nesta EPMC, uma mensagem chamada *RM-Reply* é uma resposta do componente destinatário, reconhecendo que uma mensagem foi corretamente recebida (vide Anexo B do presente trabalho).	Esta EPMC utiliza reconhecimento positivo (ACK). Segundo esta EPMC, reconhecimento é a comunicação do componente destinatário (chamado de *RM Destination*) para o componente remetente (*RM Source*), indicando correto recebimento de uma mensagem. Nesta EPMC, pode-se utilizar, opcionalmente, reconhecimento negativo (NACK), para indicar uma mensagem não aceita pelo componente destinatário (vide Anexos C e D do presente trabalho).
A interação ocorre segundo um modo específico: modo básico, modo *go-back-n* ou modo de repetição seletiva.	Nesta EPMC, o envio de mensagens entre as duas partes, após estabelecimento de conexão, ocorre segundo o modo básico de interação (vide Anexo B do presente trabalho).	Nesta EPMC, o envio de mensagens entre as duas partes, após estabelecimento de conexão, ocorre segundo o modo repetição seletiva de interação (vide Anexo D do presente trabalho).

No comparativo entre as EPMCs para cada requisito, notam-se divergências no modo como cada uma atende a esses. Maiores detalhes são descritas em seguida:

- Utiliza-se reconhecimento. Na comparação das EPMCs *WS-Reliability* e *WS-ReliableMessaging*, tem-se que ambas as EPMCs verificadas, baseiam-se em esquemas de reconhecimento positivo. Ressalta-se que o próprio W3C aponta essas duas EPMCs como “exemplos de especificações para uma infra-estrutura de reconhecimentos que tira vantagem do modelo de extensibilidade do SOAP” (W3C, 2004a).

Excepcionalmente, a EPMC *WS-ReliableMessaging*, adicionalmente ao reconhecimento positivo (ACK), também suporta a utilização de reconhecimento negativo (NACK). A adoção de reconhecimento negativo nessa EPMC, apesar de aumentar a complexidade de implementação do componente destinatário, provê retransmissões pró-ativas pelo componente remetente, tornando o processo de retransmissão mais eficiente.

Analogamente, os protocolos considerados como “de entrega confiável” adotam esquema de reconhecimento. O VMTP, o T/TCP, o NETBLT e o próprio TCP utilizam esquemas de controle de erros de transmissão baseados em reconhecimento, chamados de ARQ - *Automatic Response Request*. Todos esses protocolos utilizam reconhecimento positivo. Excepcionalmente, o TCP, utiliza também reconhecimento negativo (IREN; AMER; CONRAD, 1999; WONG; HILTUNEN; SCHLICHTING, 2001).

O conceito de reconhecimento também está presente em sistemas proprietários orientados a mensagem. No *MQ Series*, a mensagem chamada COA (*Confirmation Of Arrival*), é uma resposta indicando que uma dada mensagem foi entregue com sucesso em seu destino. Neste caso, trata-se de reconhecimento positivo (IBM, 2005a).

- Modos de interação. Na comparação das EPMCs *WS-Reliabiltiy* e *WS-ReliableMessaging*, segundo os modos de interação aplicados por essas EPMCs no envio de mensagens entre componentes de interoperabilidade, tem-se que ambas adotam modos distintos de interação.

  A EPMC *WS-Reliability* adota, em seu modelo de interoperabilidade, o modo básico de interação. Ou seja, para cada mensagem enviada do remetente ao destinatário, espera-se um reconhecimento imediato. Somente depois disso, o remetente envia a próxima mensagem. A EPMC *WS-ReliableMessaging* adota, em seu modelo de interoperabilidade o modo repetição seletiva de interação.

  Vale ressaltar que repetição seletiva é um modo de interação melhor que o modo básico e é comum em protocolos ditos de entrega confiável, bem como em

middlewares orientados a mensagem, tais como *MQ Series*. A repetição seletiva permite ao remetente utilizar a infra-estrutura de comunicação de forma mais eficiente, evitando assim, os problemas de baixa eficiência do modo básico.

4.3.3.4 Comparação quanto aos modos de conexão e desconexão

A partir das Matrizes de Mapeamento de Requisitos referentes à verificação de cada EPMC, obtém-se a tabela 4.29 que representa a comparação qualitativa entre ambas frente aos requisitos quanto aos modos de conexão e desconexão, para entrega confiável de mensagens.

Tabela 4.29. Comparação entre as EPMCs quanto aos modos de conexão e desconexão.

Requisito	***WS-Reliability***	***WS-ReliableMessaging***
Há estabelecimento de conexão.	Esta EPMC utiliza estabelecimento de conexão implícita para envio de mensagens.	Esta EPMC utiliza estabelecimento de conexão explícita com procedimento *2-way-handshake*.
Há enceramento de conexão (desconexão).	Esta EPMC utiliza encerramento de conexão implícita.	Esta EPMC utiliza encerramento de conexão explícita com procedimento *2-way-handshake*.

No comparativo entre as duas EPMCs notam-se divergências no modo como cada uma atende a esses requisitos. Maiores detalhes são abordados abaixo:

- Estabelecimento de conexão. Na comparação das EPMCs *WS-Reliabiltiy* e *WS-ReliableMessaging*, segundo os modos de conexão usados pelos respectivos componentes de interoperabilidade, tem-se que ambas são orientadas a conexão. Entretanto as duas EPMCs apresentam diferenças quanto ao modo como estabelecem a conexão.

  A EPMC *WS-Reliability* utiliza um modo de estabelecimento de conexão implícita. Neste caso, a conexão é estabelecida assim que a primeira mensagem de uma seqüência é enviada do remetente ao destinatário. Ou seja, quando o remetente precisa iniciar o envio de mensagens, ele simplesmente inicia sua transmissão. Portanto, não há comunicação preliminar especifica para o estabelecimento de

conexão, bem como qualquer verificação prévia de conexão, ou de disponibilidade, do destinatário. Nota-se que conexão implícita é mais indicada para conexões de pouca duração, usadas por aplicações com características tais como modelo assimétrico cliente-servidor e pouca troca de dados (IREN; AMER; CONRAD, 1999).

A EPMC *WS-ReliableMessaging* adota um modelo de interoperabilidade que contempla uma interação preliminar ao envio de seqüência de mensagens, caracterizando o estabelecimento de uma conexão. Essa interação preliminar é explícita e composta por uma solicitação de conexão, seguida de uma confirmação/recusa de conexão pelo destinatário. Trata-se de uma conexão explícita com procedimento *2-way-handshake*.

Analogamente, protocolos ditos de entrega confiável como o TCP, T/TCP e NETBLT, são orientados à conexão e utilizam conexão explícita. Nestes, a conexão é um pré-requisito para uma interação confiável cujo propósito é estabelecer ligações lógicas entre as partes de forma a que mantenham informações de estados, para persistência de informação, durante uma interação (IREN; AMER; CONRAD, 1999; KUROSE; ROSS, 2001).

Da mesma forma, em sistema proprietário orientado à mensagem, no *MQ Series* o conceito de conexão é chamado de sessão. Quando aplicações se comunicam utilizando *MQ Series*, é necessário estabelecer sessões. Uma sessão é uma conexão lógica ou virtual entre dois componentes que se comunicam e trocam dados. Quando dois gerenciadores de fila estabelecem uma sessão, cria-se um canal de mensagem por onde se transportam mensagens de um MCA (*Message Channel Agent*) ao outro (IBM, 2005b).

O estabelecimento de sessão (*session initiation*), no *MQ Series*, é explícito com procedimento *2-way-handshake*, similar ao verificado na EPMC *WS-ReliableMessaging*.

- Encerramento de conexão. Uma vez estabelecida a conexão e realizada transferência de dados entre dois *Web services*, o último estágio é o

encerramento da conexão. Na comparação das EPMCs *WS-Reliabiltiy* e *WS-ReliableMessaging*, segundo os modos de desconexão usados pelos respectivos componentes de interoperabilidade, nota-se que ambas realizam desconexão de formas diferentes.

A EPMC *WS-Reliability* utiliza um modo de encerramento de conexão implícita. O modo de desconexão implícita é usualmente utilizado com o modo de conexão implícita (que é o caso desta EPMC). Neste caso, quando um dos componentes de interoperabilidade não recebe informações de seu par por um certo período de tempo, ele simplesmente considera a conexão encerrada.

Na EPMC *WS-Reliability*, os elementos de marca de tempo <*GroupExpiryTime*> e o <*GroupMaxIdleDuration*> determinam o fim de uma conexão explícita, ou seja o encerramento de conexão (também chamado de término de grupo, nesta EPMC). O primeiro desses elementos de marca de tempo determina até quando o grupo será válido e, portanto, quando uma conexão implícita poderá ser desfeita entre dois componentes. O segundo elemento de marca de tempo, esse contém um tempo de duração para o intervalo entre uma mensagem e outra. Caso esse tempo seja ultrapassado, considera-se a conexão encerrada.

Na EPMC *WS-ReliableMessaging*, que adota um modelo de interoperabilidade que contempla conexão explícita, realiza-se encerramento de conexão de forma explícita com procedimento *2-way-handshake*. Neste caso, para encerrar uma conexão, o remetente envia um pedido de desconexão ao destinatário e esse retorna com uma confirmação da desconexão.

Da mesma forma, protocolos ditos de entrega confiável como o TCP, T/TCP e NETBLT, que são orientados a conexão, realizam desconexão explícita. Em *middlewares* proprietários orientados à mensagem, como o *MQ Series*, esse processo tem uma abordagem diferente. O *MQ Series* considera que canais de mensagens devem ser conexões de longo tempo. Portanto, sua forma natural de encerramento ocorre a partir de um *timeout* estabelecido por um parâmetro de controle chamado <*DisconnectInterval*> que é estipulado no estabelecimento do canal. Esse elemento representa o máximo tempo em segundos (até 1x10^6

segundos) para o qual um canal inicializado pode esperar entre uma mensagem e outra antes de encerrar essa conexão. Caso seu valor seja 0 (zero), isso indica que a conexão é permanente.

4.3.4 Comparação quanto ao ponto de vista da engenharia

A partir das Matrizes de Mapeamento de Requisitos referentes à verificação de cada EPMC, obtém-se a tabela 4.30 que representa a comparação qualitativa entre ambas frente aos requisitos da estrutura do canal, para entrega confiável de mensagens.

Tabela 4.30. Comparação entre as EPMCs quanto à estrutura de canal.

Requisito	***WS-Reliability***	***WS-ReliableMessaging***
Algoritmo de temporização de retransmissão adotado pelo objeto protocolo.	Não encontrado.	Deve-se utilizar mecanismos que ajustam retransmissão dinamicamente, similar ao RTTM, RFC 1323.

No comparativo entre as EPMCs (*WS-Reliability* e *WS-ReliableMessaging*), somente um requisito foi encontrado em sua verificação.

Com relação ao controle de tempo de intervalo de retransmissão, a EPMC *WS-Reliability* não indica como fazê-lo. Entretanto, a EPMC *WS-ReliableMessaging* indica a utilização de RTTM (*Round Trip Time Measurement*), RFC 1323, para controle de tempo de intervalo de retransmissão. Protocolos ditos de entrega confiável, tais como o TCP e o T/TCP, utilizam algoritmo de controle de tempo de retransmissão baseado em RTTM.

Com esta última tabela, completam-se as comparações entre EPMCs *WS-Reliabiblity* e *WS-ReliableMessaging*. Para finalizar este capítulo, o item subseqüente aborda conclusões gerais sobre os resultados comparativos, esses que foram obtidos a partir das verificações individuais das EPMCs.

4.4 Conclusões gerais sobre os resultados

Os resultados de verificação e comparação podem contribuir para os âmbitos de fóruns de discussão pela escolha entre uma ou outra proposta de padrão, bem como prover dados de verificação independente das EPMCs aqui verificadas, à comunidade de profissionais envolvidos em iniciativas de avaliação e utilização de tecnologias baseadas em *Web services*.

À exceção do requisito de prover integridade de mensagens, conclui-se que, pelos requisitos encontrados, ambas as EPMCs *WS-Reliabiltiy* e *WS-ReliableMessaging* verificadas e comparadas indicam contemplar os requisitos necessários para entrega confiável de mensagens em *Web services*.

Para o único requisito não contemplado – integridade – ambas justificam-se no fato de que integridade é uma competência de segurança de informação. Portanto, delega-se esse aspecto a um padrão de segurança em *Web services* – no caso, o *WS-Security*.

Nos demais requisitos estabelecidos no Modelo de Referência e utilizados para verificação individual das EPMCs, os resultados da comparação qualitativa entre ambas mostraram considerável eqüidade entre elas. Os aspectos em que ambas apresentam uma abordagem diferencial são a seguir comentados.

No aspecto de flexibilidade quanto à configuração, tanto a EPMC *WS-Reliability* quanto a EPMC *WS-ReliableMessaging*, indica atender aos requisitos de implementar propriedades de entrega garantida, entrega única e entrega ordenada. Entretanto, excepcionalmente na *WS-Reliability* estas propriedades são opcionais, ou seja, passíveis de estarem individualmente habilitadas ou desabilitadas. Essa característica de flexibilidade é particularmente mais adequada ao conceito de sistema distribuído aberto, por permitir flexibilidade no uso dessas propriedades, de acordo com a necessidade de aplicação.

Interações confiáveis entre *Web services* necessitam de mecanismos de reconhecimento. Para esse aspecto, verificou-se que ambas as EPMCs *WS-*

Reliability e *WS-ReliableMessaging* utilizam esta abordagem, baseando-se em esquemas de reconhecimento positivo. Em particular, a EPMC *WS-ReliableMessaging*, opcional e adicionalmente ao reconhecimento positivo, também suporta a utilização de reconhecimento negativo. Como a adoção de reconhecimento negativo em controle de erros de transmissão de dados habilita retransmissões pró-ativas pelo componente remetente, tem-se, portanto, que o processo de retransmissão na EPMC *WS-ReliableMessaging* apresenta melhor eficiência de controle de erros de transmissão que na EPMC *WS-Reliability*.

Há necessidade de que mensagens possuam numeração de seqüência. Ao comparar as EPMCs *WS-Reliability* e *WS-ReliableMessaging*, tem-se que em ambas EPMCs cada mensagem possui um número de seqüência dentro do grupo ao qual ela pertence. Entretanto, algumas diferenças entre essas EPMCs aparecem quanto à iniciação de contagem, capacidade de mensagens por grupo e no modo como tratam quando atingem o limite de mensagens por grupo. Esses aspectos são descritos a seguir:

- Para a contagem de mensagens, na EPMC *WS-Reliability*, a primeira mensagem de um grupo inicia seu elemento de numeração de seqüência com valor 0 (zero). Na EPMC *WS-ReliableMessaging*, esse elemento é iniciado com valor 1 (um). Esta característica da EPMC *WS-ReliableMessaging* mostra relativa robustez sobre a outra EPMC. Como zero é um valor inválido nesta EPMC, garante-se não haver ambigüidades quanto ao fato de se a variável de contagem de mensagem foi inicializada ou não. Uma outra vantagem da EPMC *WS-ReliableMessaging* é que, ao contrário da outra EPMC seu elemento de numeração de seqüência é mandatório.

- Para capacidade de mensagens, a EPMC *WS-Reliability* suporta até 2^64 mensagens por grupo. Quando esse limite é atingido, o remetente inicia um novo grupo, automaticamente. A EPMC *WS-ReliableMessaging* suporta até 2^63 mensagens por grupo e quando esse limite é atingido, o grupo é considerado finalizado, sendo necessário o remetente solicitar a criação de novo grupo. Embora a EPMC *WS-ReliableMessaging* tenha uma capacidade de numeração de seqüência, por grupo, 50% inferior em relação à EPMC *WS-Reliabilty*, essa

deficiência pode ser irrelevante quando considerada a capacidade de mensagens suportada por grupo em um sistema proprietário orientado à mensagem, usual de mercado utilizado em instituições financeiras (como exemplo, o *MQseries* suporta 1x10^9 mensagens). Um outro aspecto favorável à EPMC *WS-ReliableMessaging* é o tratamento para o evento de limite máximo de mensagens atingido em um grupo. Nesta EPMC, a decisão por iniciar novo grupo cabe à aplicação e não deliberadamente ao componente de interoperabilidade.

Outro requisito em que ambas as EPMCs abordam, mas com pequena variação entre elas, é o de que mensagens possuem identificação única. Na comparação entre as EPMCs *WS-Reliabiltiy* e *WS-ReliableMessaging*, segundo este requisito, tem-se que estas suportam a identificação única de mensagem, embora não tenham um elemento exclusivo para esse parâmetro em sua estrutura de cabeçalho. Em ambas as EPMCs o identificador único de uma mensagem é obtido a partir da concatenação do elemento que possui o identificador único do grupo ao qual ela pertence (esse baseado em URI), com o número de seqüência dessa mensagem, dentro do respectivo grupo. Apesar desta equivalência, ressalta-se que a EPMC *WS-Reliability*, em particular, define seu elemento de número de seqüência <*SequenceNum*> como opcional no cabeçalho de uma mensagem e essa característica pode inviabilizar a identificação única de mensagem. Portanto, para este aspecto, a EPMC *WS-ReliableMessaging* indica melhor robustez, uma vez que os elementos que compõem o identificador único de uma mensagem são sempre mandatórios no cabeçalho.

Quanto ao requisito de utilização de marcas de tempo em mensagens, a comparação entre as EPMCs *WS-Reliability* e *WS-ReliableMessaging* também indica que ambas contemplam esse aspecto, porém de formas diferentes. Segundo a EPMC *WS-Reliability*, num cabeçalho de mensagem pode haver até três elementos distintos de marcas de tempo que controlam a expiração da mensagem, a expiração do grupo e o tempo máximo do intervalo entre uma mensagem e outra do grupo, respectivamente. Os dois primeiros elementos expressam tempo absoluto, baseado no padrão UTC (*Universal Time Coordinated*) e, portanto, é necessário que ambos os componentes de interoperabilidade estejam em sincronismo de relógio por meio de um serviço externo. De forma diferente, na EPMC *WS-ReliableMessaging*

as mensagens trocadas não possuem marcas de tempo. Esse aspecto é tratado somente no processo de iniciação do grupo (estabelecimento de conexão). Durante o processo de iniciação de um grupo o componente remetente envia uma solicitação ao destinatário, informando qual será o tempo de duração dessa seqüência. Este tempo de duração é expresso como tempo relativo e representa o intervalo de tempo após o qual esse grupo será considerado inválido. Quaisquer mensagens pertencentes a este grupo que cheguem ao destinatário após esse tempo serão desconsideradas. Nota-se portanto que, para este requisito, a EPMC, *WS-Reliability* permite ao destinatário determinar se a mensagem expirou apenas comparando-se a data e o horário correntes (por meio do UTC) com a data e o horário expressos no cabeçalho da mensagem. Pelo contrário, na EPMC *WS-ReliableMessaging*, as mensagens não possuem marcas de tempo e é necessário manter uma persistência de informação referente ao tempo de grupo da mensagem durante toda a duração deste grupo. Tem-se, portanto, uma melhor abordagem pela EPMC *WS-Reliabiltiy*, frente ao respectivo requisito de uso de marcas de tempo.

Na comparação entre as EPMCs *WS-Reliabiltiy* e *WS-ReliableMessaging*, segundo as atividades realizadas em checagem de mensagem por um componente de interoperabilidade destinatário ao receber uma mensagem, tem-se que ambas são semelhantes na semântica de verificação e tratamento das mensagens recebidas, mas com algumas peculiaridades. Na EPMC *WS-Reliability* as mensagens recebidas pelo destinatário, que estejam fora de ordem, são descartadas. Na EPMC *WS-ReliableMessaging*, mensagens que chegam fora de ordem ao destinatário, pertencentes a uma seqüência válida, não são descartadas. Elas são armazenadas para posterior entrega ordenada à respectiva aplicação. Isso se deve ao fato de que a EPMC *WS-ReliableMessaging* utiliza modo de interação baseado em repetição seletiva. Essa característica torna a EPMC *WS-ReliableMessaging* mais eficiente no processo de transmissão de mensagens.

Outro requisito, com diferentes abordagens entre as EPMCs, é o de modo de interação. A EPMC *WS-Reliability* adota, em seu modelo de interoperabilidade, o modo básico (“pare-e-espere”). Embora esse modo seja simples, ele é ineficiente para o caso de muitas mensagens a enviar. Além disso, essa ineficiência pode se tornar inaceitável em sistemas onde o atraso de propagação é alto. Ou seja, para

cada mensagem enviada do componente remetente ao componente destinatário, espera-se um reconhecimento imediato. Além disso, esta EPMC não faz alusão a qual algoritmo é utilizado para o controle de tempo de retransmissão. Por outro lado, a EPMC *WS-ReliableMessaging* adota, em seu modelo de interoperabilidade, o modo repetição seletiva de interação, utilizando RTTM (*Round Trip Time Measurement*) para controle de tempo de intervalo de retransmissão. Portanto, para o requisito de modo de interação adotado, a EPMC *WS-ReliableMessaging* apresenta uma abordagem mais eficiente no controle de erros de transmissão.

Por último, mas não menos importante, estão os modos de estabelecimento de conexão e de desconexão. A comparação entre as EPMCs *WS-Reliabiltiy* e *WS-ReliableMessaging*, segundo os modos de conexão usados pelos respectivos componentes de interoperabilidade para o estabelecimento e encerramento de conexão, indica que ambas as EPMCs são orientadas a conexão, mas com diferenças no modo adotado. A EPMC *WS-Reliability* utiliza um modo de estabelecimento e de encerramento de conexão implícita, ou seja, quando o remetente precisa iniciar o envio de mensagens, ele simplesmente inicia sua transmissão. Portanto, não há comunicação preliminar especifica para o estabelecimento de conexão, bem como qualquer verificação prévia de conexão (ou de disponibilidade) do destinatário. Para desconexão, quando um dos componentes de interoperabilidade não recebe informações de seu par por um certo período de tempo, ele simplesmente considera a conexão encerrada. Com outra abordagem, a EPMC *WS-ReliableMessaging* adota um modelo de interoperabilidade que contempla uma interação preliminar ao envio de seqüência de mensagens, caracterizando o estabelecimento de uma conexão. Essa interação é explícita, com procedimento *2-way-handshake*. Essa EPMC também realiza encerramento de conexão de forma explícita com procedimento *2-way-handshake*.

Conexões e desconexões implícitas são geralmente indicadas para interações entre aplicações que compartilham características como um modelo assimétrico (cliente-servidor), comunicação de curta duração e pouca troca de dados. Entretanto, vale observar que interações entre parceiros de negócios, via *Web services*, tendem a ir além de uma única interação. Ou seja, em *Web services*, é factível haver várias

interações por um período indeterminado (horas, dias, semanas, meses), referente a um mesmo contexto de negócios (W3C 2004c).

Com esta característica em *Web services*, apontada pelo W3C, mecanismos de conexão explícita (*2-way-handshake* e/ou *3-way-handshake*) são mais indicados para conexões de longa duração e para evitar falsos estabelecimentos e encerramentos de conexão. Mais uma vez, a EPMC *WS-ReliableMessaging* apresenta seu diferencial favorável à implementação de mensagem confiável em *Web services*.

Visando a uma síntese sobre resultados de verificação e comparação obtidos, a tabela 4.31 consolida de forma sucinta e objetiva os resultados referentes a cada uma das EPMCs, frente aos requisitos definidos no Modelo de Referência e utilizados como parâmetros de verificação.

Tabela 4.31. Síntese dos resultados de verificação e comparativo entre as EPMCs.

Requisito	***WS-Reliability***	***WS-ReliableMessaging***
Tem por objetivo prover interoperabilidade confiável entre Web services.	Sim.	Sim.
As interações são realizadas por componentes de interoperabilidade.	Sim.	Sim.
Os componentes de interoperabilidade assumem funções empresariais distintas (remetente ou destinatário) que realizam procedimentos de envio e recebimento de mensagem, respectivamente.	Sim.	Sim.
Provê entrega garantida de mensagem.	Sim. Obs: Esta propriedade pode ser desabilitada nesta EPMC.	Sim. Obs: Esta propriedade é invariante nesta EPMC.
Provê entrega não duplicada de mensagem.	Sim. Obs: Esta propriedade pode ser desabilitada nesta EPMC.	Sim. Obs: Esta propriedade é invariante nesta EPMC.
Provê entrega ordenada de mensagens.	Sim. Obs: Esta propriedade pode ser desabilitada nesta EPMC.	Sim. Obs: Esta propriedade é invariante nesta EPMC.

(Continua)

(Continuação)

Requisito	*WS-Reliability*	*WS-ReliableMessaging*
Provê entrega íntegra de mensagem.	Não contemplado.	Não contemplado.
Há independência de protocolo de transporte.	Sim. Obs: Adicionalmente, há uma especificação de uso de método POST do HTTP para transporte de mensagens confiáveis.	Sim.
Mensagens pertencem a um grupo.	Sim.	Sim.
A interação é unidirecional entre componentes de interoperabilidade.	Sim.	Sim.
A interação é baseada em mensagem.	Sim. O objeto informação é uma mensagem SOAP.	Sim. O objeto informação é uma mensagem SOAP.
As mensagens possuem numeração de seqüência no cabeçalho.	Sim. Obs: - O parâmetro de número de seqüência é opcional nesta EPMC. - Esse parâmetro é sempre inicializado com 0 (zero) para a primeira mensagem. A partir daí, incrementa-se seu valor de 1 (um) para cada mensagem adicional, do mesmo grupo, a ser enviada. - O limite de mensagens em um grupo é de 2^64 mensagens. - Atingido o limite de mensagens (evento *rollover*), um novo grupo é automaticamente criado pelo componente de interoperabilidade remetente.	Sim. Obs: - O parâmetro de número de seqüência é mandatório nesta EPMC. - Esse parâmetro é sempre inicializado com 1 (um) para a primeira mensagem. A partir daí, incrementa-se seu valor de 1 (um) para cada mensagem adicional, do mesmo grupo, a ser enviada. - O limite de mensagens em um grupo é de 2^63 mensagens. - Atingido o limite de mensagens (evento *rollover*), finalize-se o grupo e a aplicação remetente deve solicitar criação de novo grupo.
As mensagens possuem identificação única.	Sim. A identificação única é obtida a partir da concatenação dos parâmetros de identificação de grupo (baseada em URI), com o número de seqüência da mensagem. Obs: Uma vez que o parâmetro de número de seqüência é opcional nesta EPMC, este requisito pode não ser atendido.	Sim. A identificação única é obtida a partir da concatenação dos parâmetros de identificação de grupo (baseada em URI), com o número de seqüência da mensagem.

(Continua)

(Continuação)

Requisito	*WS-Reliability*	*WS-ReliableMessaging*
As mensagens possuem parâmetro de identificação de grupo de mensagens no cabeçalho.	Sim. A identificação de um grupo é baseada em formato padrão URI (*Uniform Resource Identifier*).	Sim. A identificação de um grupo é baseada em formato padrão URI (*Uniform Resource Identifier*).
As mensagens possuem parâmetro de marca de tempo.	Sim. Todas as mensagens possuem marca de tempo em seu cabeçalho, com base no padrão UTC.	Não contemplado. Nesta EPMC, o conceito de expiração está somente relacionado com um grupo (chamado de seqüência).
As mensagens podem assumir dois estados possíveis: “mensagem de interação” ou “mensagem de reconhecimento”.	Sim.	Sim.
Antes de enviar uma mensagem, o componente de interoperabilidade remetente estende o cabeçalho SOAP, inserindo um bloco com prefixo *wsrm*. Nessa estrutura estendida, são inseridos os elementos de parâmetros para a identificação do grupo e do número de seqüência de mensagem, respectivamente. Além disso, são inseridos parâmetros de marca de tempo e de controle de integridade.	Sim. Obs: Todas as mensagens de conteúdo específico de aplicação possuem marca de tempo. Não há elementos de controle de integridade.	Sim. Obs: Não se inserem marca de tempo em cabeçalhos de mensagens. Não há elementos de controle de integridade.
Ao receber uma mensagem, o componente destinatário procede uma checagem desta.	Sim. Obs: Mensagens fora de ordem ou expiradas são rejeitadas.	Sim. Obs: Mensagens fora de ordem podem ser aceitas.
A interface do componente de interoperabilidade é do tipo “operação”, indicando uma relação cliente/servidor.	Sim.	Sim.
Utiliza-se reconhecimento.	Sim. Reconhecimento positivo.	Sim. Reconhecimento positivo e, opcionalmente, recolhecimento negativo.
A interação ocorre segundo um modo específico.	Sim. Modo básico (*stop-and-wait*).	Sim. Modo de repetição seletiva.
Há estabelecimento de conexão.	Sim. Conexão implícita.	Sim. Conexão explícita com procedimento *2-way-handshake*.
Há enceramento de conexão (desconexão).	Sim. Desconexão implícita.	Sim. Desconexão explícita com procedimento *2-way-handshake*.

(Continua)

(Continuação)

Requisito	*WS-Reliability*	*WS-ReliableMessaging*
Adota-se algoritmo de temporização de retransmissão de mensagens.	Não especificado.	Sim. Recomenda-se utilizar mecanismos que ajustam retransmissão dinamicamente, similar ao RTTM (RFC 1323).

Pelo exposto, quanto aos requisitos contemplados e cuja comparação qualitativa indica diferentes abordagens, observa-se relativo diferencial em vantagem à EPMC *WS-ReliableMessaging*. Essa EPMC destacou-se por utilizar mecanismos mais eficientes em controle de erros de transmissão e utilização de canal de comunicação, pelo uso de modo de interação baseado em repetição seletiva. Além disso, essa EPMC utiliza procedimentos de conexão e desconexão explícitos, hoje mais adequados às necessidades de expansão da aplicabilidade de *Web services*. Essa EPMC também, mostrou maior robustez no controle de numeração de mensagens, bem como na identificação única de mensagens. Por último, há um compromisso desta em recomendar uso de padrão específico para temporização de retransmissão de mensagens.

Portanto, com base no presente Método de Verificação proposto e aplicado, conclui-se que a EPMC *WS-ReliableMessaging* tem melhor indicação à adoção, como padrão oficial de mensagem confiável em *Web services,* pela indústria de software.

Entretanto, ressalta-se que a adoção de uma das duas EPMCs aqui verificadas pelo mercado depende de sua incorporação na próxima versão do *basic profile*[18] pela entidade WS-I (*Web Services Interoperability Organization*). A WS-I objetiva promover a interoperabilidade entre as implementações *Web services* da indústria, pela publicação dos "*basic profiles*". Ela anunciou em 28/03/2007 a publicação da versão preliminar (chamada *Board Approval Draft*) do *basic profile v1.2*, o qual pretende incluir outros padrões além dos suportados no basic profile v1.1 vigente. Entretanto, essa versão ainda não considera a definição de um padrão referente a entrega confiável de mensagens em *Web services* a ser adotado pela indústria. Esse aspecto está em pauta somente para o *basic profile* v 2.0 que ainda não possui previsão de publicação.

[18] Os *basic profiles* são descrições de convenções e práticas para o uso de combinações específicas de padrões através das quais os sistemas *Web services* de diferentes fornecedores de software poderão se interagir.

O presente capítulo apresentou os resultados da verificação obtidos, a partir do Método de Verificação proposto. A partir desses resultados, na forma das respectivas Matrizes de Mapeamento de Requisitos, foi possível estabelecer comparativo qualitativo entre as EPMCs, permitindo obter-se conclusões gerais sobre suas aderências aos requisitos de entrega confiável de mensagens bem como os pontos de divergência entre elas. Passa-se então ao capítulo final que apresenta as conclusões gerais e sugestão de continuidade de pesquisa.

5 CONSIDERAÇÕES FINAIS

Neste último capítulo, apresentam-se as conclusões deste trabalho com relação ao método proposto, bem como oportunidades identificadas para a continuidade de pesquisa em trabalhos subseqüentes a este.

5.1 Conclusões sobre o Método de Verificação

Este trabalho propôs e apresentou a aplicabilidade de um Método de Verificação de especificações de padrões de mensagem confiável em *Web services*. Aspectos conclusivos pertinentes a viabilidade, imparcialidade e reuso do Método de Verificação bem como aplicabilidade do RM-ODP são explanados a seguir.

Viabilidadde do Método. O Método de Verificação proposto estabeleceu um Processo de Verificação constituído por avaliação de especificação de arquitetura de software, baseada em revisão ativa de projeto, utilizando-se técnicas de questionamento e inspeção. Ele mostrou sua viabilidade ao ser aplicado sobre as duas EPMCs - *WS-Reliabilty* e *WS-ReliableMessaging* - que concorrem atualmente para se tornar o padrão de entrega confiável de mensagens em *Web services*, a ser adotado pela indústria de software, obtendo-se assim suas Matrizes de Mapeamento de Requisitos. Uma vez obtidas as Matrizes de Mapeamento de Requisitos individuais de verificação respectivas a cada uma dessas EPMCs, estabeleceu-se uma comparação qualitativa entre elas. Posto isto, obteve-se uma visão sobre como cada uma delas atende aos requisitos de entrega confiável de mensagens em *Web services*, bem como foi possível destacar pontos divergentes entre elas.

Aplicabilidade do RM-ODP. Outra característica explorada na composição do Método de Verificação foi a aplicabilidade do RM-ODP na especificação de um protocolo de aplicação como arquitetura de referência, visando a modelar um contexto não funcional (a entrega confiável de mensagens) de uma arquitetura *Web services*. O RM-ODP teve sua importância em prover uma abordagem sistemática, consistente e estruturada em níveis de abstração, viabilizando-se a especificação do

Modelo de Referência para entrega confiável de mensagens, conforme descrito no capítulo 3.

Imparcialidade. Para a especificação do Modelo de Referência, estruturado segundo o RM-ODP, tomou-se por base os requisitos de entrega confiável de mensagens, estabelecidos pelo W3C. Sendo assim, obteve-se um Modelo de Referência imparcial com relação à quaisquer EPMCs que tenham sido propostas aos órgãos de padronização *Web services*.

Verificação independente. Esta característica é intrínseca ao fato de que o Método de Verificação foi proposto e aplicado, sem quaisquer vínculos ou participação das entidades que atuaram na elaboração e proposição das EPMCs verificadas.

Reuso. O W3C, por meio de sua especificação da WSA, define um modelo de interoperabilidade que visa à obtenção de um *middleware* aberto orientado à mensagens. Este modelo e seus requisitos foram base para a especificação do Modelo de Referência, elemento do Método de Verificação, que provê os requisitos a serem verificados nas EPMCs. Pelo exposto, o presente Método de Verificação é aplicável, como visto, às duas EPMCs aqui verificadas, mas também viabiliza a verificação e comparação de outras especificações de padrão de entrega confiável de mensagem que venham a surgir.

5.2 Continuidade da Pesquisa

Com base no presente trabalho foram identificadas possíveis vertentes de continuidade de pesquisa, a seguir descritas.

Especificação de nova EPMC - Com base nos requisitos encontrados nas duas EPMCs verificadas no presente trabalho, a proposição de um terceiro possível padrão de mensagem confiável poderia ser uma continuidade factível ao presente trabalho. O principio dessa proposição seria o de preservar os requisitos já atendidos por ambas as EPMCs verificadas, porém tomando-se vantagem dos aspectos onde cada uma destacou-se por melhor abordagem a cada requisito na comparação qualitativa. Posto isto, uma terceira EPMC agregaria as propriedades de flexibilidade,

suporte a marcas de tempo em mensagens e uso de referência de tempo com base no UTC, herdados da EPMC *WS-Reliability*. Além disso, essa nova EPMC utilizaria uma abordagem similar à EPMC *WS-ReliableMessaging* quanto à melhor eficiência no controle de erros de transmissão (uso de repetição seletiva e reconhecimento negativo), melhor robustez na contagem de numeração de seqüência de mensagens e no tratamento de limite máximo de mensagem em grupo, bem como uma interoperabilidade baseada em procedimentos de conexão e desconexão explícitos.

Proposição de verificação ponderada frente aos pontos de vista do RM-ODP – um dos elementos do Método de Verificação, o Modelo de Referência, foi composto por trinta requisitos distribuídos em diferentes níveis de abstração. A verificação e comparação dos requisitos no presente trabalho, para as respectivas EPMCs *WS-Reliability* e *WS-ReliableMessaging*, teve uma abordagem equânime com respeito à importância dos requisitos a serem verificados. Durante a elaboração do presente trabalho, não foram encontrados trabalhos correlatos para subsídios em estabelecer ponderação na verificação relativa aos diferentes níveis de abstração no RM-ODP. Portanto, um aspecto de ponderação de requisitos ainda não publicado e/ou explorado caracterizaria uma possível continuidade deste trabalho.

Extensão do método para verificação de outros padrões – a partir dos elementos do método aqui proposto e considerando-se desenvolver uma abordagem mais abstrata e aberta ao Modelo de Referência, uma possível frente de pesquisa (com inerente potencialização da aplicação do RM-ODP), seria estender o presente método para avaliação e/ou comparação de outros padrões *Web services.* O objetivo é ir além do contexto de entrega confiável de mensagens, abrangendo outros padrões em pauta pelas entidades de padronização e que aguardam sua adoção pela indústria (ex: gerenciamento e segurança).

Referências

ALBIN, S. **The art of software architecture: design methods and techniques**. Indianapolis: John Wiley & Sons, 2003.

ALMEIDA, J.; SINDEREN, M.; PIRES, L. **The role of the RM-ODP computational viewpoint concepts in the MDA approach.** Ensachede: Centre for Telematics and Information Technology, University of Twente, 2004. Disponível em: <http://www.lcc.uma.es/~av/wodpec2004/papers/2-almeida.pdf>. Acesso em: 02 set. 2005.

BABAR, M. et al. **A framework for classifying and comparing software architecture evaluation**. [S.I.]: IEEE Computer Society, 2004. Disponível em: < http://ieeexplore.ieee.org/iel5/9061/28748/01290484.pdf?arnumber=1290484>. Acesso em: 13 ago. 2006.

BALCI, O. **Verification, validation and accreditation**. Washington: ACM WSC, 1998. Disponível em: <http://portal.acm.org/citation.cfm?id=293183&coll=ACM&dl=ACM&CFID=60763455&CFTOKEN=62577478>. Acesso em: 23 nov. 2005.

BASS, L.; CLEMENS, P.; KAZMAN, R. **Software architecture in parctice**. Boston: Adison Wesley, 2003.

BARCELOS, R. **Uma abordagem para inspeção de documentos arquiteturais baseadas em checklist**. 2006. Dissertação (Mestrado) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

BECERRA, J. **Aplicabilidade do padrão de processamento distribuído e aberto nos projetos de sistemas abertos de automação**. 1998. Tese (Doutorado) – Escoloa Politécnica, Universidade de São Paulo, São Paulo, 1998.

BIRMAN, K.; RENESSE, R.; VOGELS, W. **Adding high availability and autonomic behavior to web services**. [S.I.]: IEEE Computer Society, 2004a. Disponível em: <http://portal.acm.org/ft_gateway.cfm?id=999394&type=pdf&coll=ACM&dl=ACM&CFID=38982205&CFTOKEN=26603750>. Acesso em: 17 fev. 2005.

BIRMAN, K. **Like it or not, web services are distributed objects.** New York: ACM Press , 2004b. Disponível em: **<**http://portal.acm.org/citation.cfm?id=1035157&coll=ACM&dl=ACM&CFID=54827295&CFTOKEN=93308840 >. Acesso em: 17 fev. 2005.

CHEW, J.; SULLIVAN, C. **Verification, validation, and accreditation in the life cycle of models and simulations**. San Diego: Society for Coumputer Simulation International, 2000. Disponível em: <http://portal.acm.org/citation.cfm?id=510495&coll=ACM&dl=ACM&CFID=60763455&CFTOKEN=62577478>. Acesso em: 10 nov. 2005.

CLEMENTS, P. **Evaluating software architectures: methods and case studies**. [S.I.]: Addison Wesley, 2002.

CONRADI R. et al. Object-oriented reading techniques for inspection of UML models - an industrial experiment. In: EUROPEAN CONFERENCE ON OBJECT-ORIENTED PROGRAMMING, 2003, Darmstadt. **Proceedings...** 2003. Disponível em: < **http://www.idi.ntnu.no/grupper/su/publ/pdf/ecoop03-oort-experiment-final.pdf**>. Acesso em: 02 dez. 2006.

DOBRICA, L.; NIEMELÄ, E. **A survey on software architecture analysis methods**. [S.I.]: IEEE Computer Society, 2002. Disponível em:< http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/trans/ts/&toc=comp/trans/ts/2002/07/e7toc.xml&DOI=10.1109/TSE.2002.1019479>. Acesso em: 22 jul. 2006.

UNITED STATES. Department of Defense (DoD). **V&V techniques: verification, validation & accreditation recommended practices guide.** US**,** 2001. disponível em: http://vva.dmso.mil/Ref_Docs/VVTechniques/VVtechniques-pr.PDF, acesso em: 20 nov. 2005).

EGYHAZY, C.; MUKHERJI, R. **Interoperability architecture using RM-ODP**. New York: ACM Press, 2004. Disponível em: < http://portal.acm.org/ft_gateway.cfm?id=966397&type=pdf&coll=ACM&dl=ACM&CFID=53329651&CFTOKEN=60235755> Acesso em: 22 ago. 2005.

ENDREI, M. et al. **Patterns: service-oriented architecture and web services**. [S.I.]: IBM Redbooks, 2004. Disponível em: < http://publib-b.boulder.ibm.com/abstracts/sg246303.html?Open>. Acesso em: 02 fev. 2005.

ERRADI, A.; MAHESHWARI, P. **wsBus: a framework for reliable web services interactions**. New York: ACM Press, 2005. Disponível em: <http://portal.acm.org/citation.cfm?doid=1067070>. Acesso em: 28 ago. 2005.

FERGUSON, D. et al. **Reliable message delivery in a Web services world: a proposed architecture and roadmap**. [S.I.]: IBM, 2003. Disponível em: <http://www-128.ibm.com/developerworks/library/ws-rmdev/>. Acesso em: 01 mar. 2005.

HOLLEY, K. et al. **Migrating to a service-oriented architecture, part-1**. [S.I.]: IBM, 2003. Disponível em: <http://www128.ibm.com/developerworks/webservices/library/ws-migratesoa/> . Acesso em: 05 fev. 2005.

INTERNATIONAL BUSINESS MACHINES (IBM). **WebSphere MQApplication programming reference**. [S.I.], 2005a. Disponível em: <http://www-306.ibm.com/software/integration/wmq/library/library6x.html>. Acesso em: 22 out. 2006.

_____. **WebSphere MQ V6 fundamentals**. [S.I.], 2005b. Disponível em: <http://www-306.ibm.com/software/integration/wmq/library/library6x.html>. Acesso em: 22 out. 2006.

_____. **WebSphere Intercommunication**. [S.I.], 2005c. Disponível em: <http://www-306.ibm.com/software/integration/wmq/library/library6x.html>. Acesso em: 22 out. 2006.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS (IEEE). **IEEE Std 610.12-1990:**. Glossary of software engineering terminology. [S.I.], 1990.

_____. **IEEE/EIA-12207.0-1996:** Industry Implementation of International Standard ISO/IEC 12207: 1995 (ISO/IEC 12207) Standard for Information Technology- Software life cycle processes. [S.I.], 1996.

_____. **IEEE/EIA Std 1012-2004**: Software verification and validation. [S.I], 2004.

IREN, S.; AMER, P.; CONRAD, P. **The transport layer: tutorial and survey**. [S.I.]: ACM Press, 1999. Disponível em: <http://portal.acm.org/ft_gateway.cfm?id=344609&type=pdf&coll=GUIDE&dl=GUIDE&CFID=51389760&CFTOKEN=19679607>. Acesso em: 20 Jul. 2005.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO). **ISO/IEC Std 10746-2: 1996:** Information technology - open distributed processing - reference model: foundations. [S.I.], 1996a.

_____. **ISO/IEC Std 10746-3:1996**. Information technology - open distributed processing - reference model: architecture. [S.I.], 1996b.

_____. **ISO/IEC Std 10746-1: 1998**. Information technology - open distributed processing - reference model: overview. [S.I.], 1998a.

_____. **ISO/IEC Std 10746-4: 1998**. Information technology - open distributed processing - reference model: architectural semantics. [S.I.], 1998b.

KAYE, D. **Loosely coupled: the missing pieces of web services**. [S.l.]: RDS Associates, 2004.

KUROSE, J.; ROSS, K. **Computing networking: a top-down approach featuring the Internet**. New York: Addison Wesley Longman, 2001.

LOWELL, D.; CHEN, P. **Persistent messages in local transactions**. [S.I.]: ACM Press, 1998. Disponível em: <http://portal.acm.org/citation.cfm?id=277737&coll=portal&dl=ACM>. Acesso em: 20 abr. 2006.

MUKHI, N. et al. **Supporting policy-driven behaviors in web services: experiences and issues**. New York: ACM Press, 2004a. Disponível em: <http://portal.acm.org/ft_gateway.cfm?id=1035214&type=pdf&coll=ACM&dl=ACM&CFID=38210308&CFTOKEN=82143073>. Acesso em: 16 fev. 2005.

MUKHI, N.; KONURU, R.; CORBERA, F. **Cooperative middleware specialization for service oriented architecture**. New York: ACM Press, 2004b. Disponível em: <http://portal.acm.org/citation.cfm?id=1013401&coll=ACM&dl=ACM&CFID=67621117&CFTOKEN=34692228>. Acesso em: 16 fev. 2005.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA). **NASA-GB-A302:** Software formal inspections guidebook. [S.I.], 1993. Disponível em: <http://satc.gsfc.nasa.gov/fi/gdb/fitext.txt>. Acesso em: 15 jul. 2006.

NAUMENKO, A.; WEGMANN, A. **MDA and RM-ODP: two approaches in modern ontological engineering**. Lausanne: Laboratory of Systemic Modeling, Swiss Federal Institute of Technology, 2003. Disponível em: <http://lcawww.epfl.ch/Publications/Naumenko/NaumenkoW03.pdf>. Acesso em: 02 set. 2005.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). **Reference information for the software verification and validation process.** [S.I.], 1996. Disponível em: <http://www.armysoftwaremetrics.org/documents/NIST/NIST_234.PDF#search=%22Reference%20Information%20for%20the%20Software%20verification%20and%20validation%20pdf%22>. Acesso em: 23 jul. 2006.

ORGANIZATION FOR THE ADVANCEMENT OF STRUCTURED INFORMATION STANDARDS (OASIS). **Web services reliable messaging TC**. [S.I.], 2004a. Disponível em: <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrm>. Acesso em: 17 mar. 2005.

_____. **Web services reliable messaging v1.1 (WS-Reliability)**. [S.I.], 2004b. Disponível em: < http://docs.oasis-open.org/wsrm/ws-reliability/v1.1/wsrm-ws_reliability-1.1-spec-os.pdf>. Acesso em: 17 mar. 2005.

_____. **Reliable messaging: recent versions of WS-Reliability and WS-Reliable Messaging**. [S.I.], 2004c. Disponível em: <http://xml.coverpages.org/ni2004-03-11-a.html>. Acesso em: 18 mar. 2005.

_____. **Web services distributed management (WSDM) Technical Commitee.** [S.I.], 2005a. Disponível em: <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm>. Acesso em: 17 jul. 2006.

_____. **Web services reliable exchange (WS-RX) Technical Commitee.** [S.I.], 2005b. Disponível em: <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-rx>. Acesso em: 17 mar. 2005.

_____. **Cover Pages Technology Reports: Reliable Messaging.** [S.I.], 2006a. Disponível em: <http://xml.coverpages.org/reliableMessaging.html>. Acesso em: 15 jul. 2006.

_____. **Web services notification (WSN) Technical Commitee.** [S.I.], 2006b. Disponível em: <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn>. Acesso em: 17 jul. 2006.

_____. **Web Services Reliable Messaging v1.1 (WS-Reliable Messaging)**. [S.I.], 2006c. Disponível em: <http://docs.oasis-open.org/ws-rx/wsrm/200608/wsrm-1.1-spec-cd-04.pdf>. Acesso em: 01 set. 2006.

_____. **Web services security (WSS) Technical Commitee**. [S.I.], 2006d. Disponível em: <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss>. Acesso em: 08 jul. 2006.

_____. **Web services transaction (WS-TX) Technical Commitee**. [S.I.]: OASIS, 2006e. Disponível em: <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-tx>. Acesso em: 08 jul. 2006.

OBJECT MANAGEMENT GROUP (OMG). **Unified modeling language**. [S.I.], 2005. Disponível em: <www.uml.org>. Acesso em: 12 jul. 2005.

PALLICKARA, S.; FOX, G. **An analysis of reliable delivery specifications for web services**. [S.I.]: IEEE Computer Society, 2005. Disponível em: < http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/proceedings/itcc/&toc=comp/proceedings/itcc/2005/2315/01/2315toc.xml&DOI=10.1109/ITCC.2005.68>. Acesso em: 11 jul 2005.

PARNAS, D.; WEISS, D. **Active design review**. London: ACM Press, 1985. Disponível em: <http://portal.acm.org/citation.cfm?id=319599&coll=ACM&dl=ACM&CFID=2367568&CFTOKEN=18829004>. Acesso em: 18 jul. 2005.

PRESSMAN, R. S. **Software engineering: a practitioner's approach**. 5. ed. New York: McGrawHill, 2001.

PUTMAN, J. **Architecting with RM-ODP**. New Jersey: Prentice-Hall, 2001.

RAKITIN, S. **Software verification and validation for practitioners and managers**. 2. ed. Boston: Artech House, 2001.

RAMA, M. **Requirements-driven software test: a process-oriented approach**. New York: ACM Press, 1996. Disponível em: <http://portal.acm.org/citation.cfm?id=232088&coll=ACM&dl=ACM&CFID=5944973&CFTOKEN=70182714>. Acesso em: 16 jul. 2006.

RUMBAUGH, J.; BOOCH, G.; JACKOBSON, I. **The unified modeling language user guide**. 9. ed. Indianapolis: Addison Wesley, 2001.

SHULL, F. et. al. **How perspective-based reading can improve requirements inspections**. Los Alamitos: ACM Press, 2000. Disponível em: <http://portal.acm.org/citation.cfm?id=621516&coll=Portal&dl=GUIDE&CFID=15691051&CFTOKEN=29392721>. Acesso em: 02 dez. 2006.

SINDEREN, M. **On the design of application protocols**. Tese (Doutorado) – University of Twente, 1995. Disponível em:<http://wwwhome.cs.utwente.nl/~sinderen/publications/thesis.html>. Acesso em: 19 set. 2006.

SLEEPER, B. **The five missing pieces of SOA**. [S.I.]: WS-I, 2004. Disponível em: <http://www.ws-i.org/press/inthenews.aspx>. Acesso em: 19 fev. 2005.

TAI, S.; MIKALSEN, T.; ROUVELLOU, I. **Using message-oriented middleware for reliable web services messaging**. New York: IBM T. J. Watson Research Center, 2003. Disponível em: <http://www.research.ibm.com/AEM/pubs/wes2003final.pdf>. Acesso em: 14 out. 2005.

TANENBAUM, A.S; STEEN, M. **Distributed sytems – principles and paradigms**. New Jersey: Prentice-Hall, 2002.

WORLD WIDE WEB CONSORTIUM (W3C). **Simplle object access protocol SOAP (1.1)**. [S.l.], 2000. Disponível em: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>. Acesso em: 05 fev. 2005.

_____. **XML schema part 2: datatypes**. [S.I.], 2004. Disponível em: < http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#ISO8601>. Acesso em: 01 jun. 2006.

_____. **Web services architectures**. [S.I.], 2004a. Disponível em: < http://www.w3c.org/TR/ws-arch/>. Acesso em: 02 fev. 2005.

_____. **Web services architecture requirements**. [S.I.], 2004b. Disponível em: <http://www.w3.org/TR/wsa-reqs/>. Acesso em: 03 fev. 2005.

_____. **Web services architecture usage scenarios**. [S.I.], 2004c. Disponível em: <http://www.w3.org/TR/ws-arch-scenarios/ >. Acesso em: 03 fev. 2005.

_____. **WS choreography model overview**. [S.I.], 2004d. Disponível em: <http://www.w3.org/TR/2004/WD-ws-chor-model-20040324/>. Acesso em: 03 fev. 2006.

_____. **Web services internationalization usage scenarios**. [S.I.], 2004e. Disponível em: <http://www.w3.org/TR/2004/NOTE-ws-i18n-scenarios-20040730/>. Acesso em: 03 fev. 2005.

_____. **Web services glossary**. [S.I.], 2004f. Disponível em: <http://www.w3c.org/TR/ws-gloss/>. Acesso em: 03 fev. 2005.

_____. **WORKSHOP ON WEB OF SERVICES FOR ENTERPRISE COMPUTING, 2007. Position paper…** [S.I.], 2007. Disponível em: <http://www.w3.org/2007/01/wos-papers/gall>. Acesso em: 03 fev. 2007.

WEGMANN, A.; NAUMENKO, A. **Conceptual modeling of complex systems using an RM-ODP based ontology**. Lausanne: Laboratory of Systemic Modeling, Swiss Federal Institute of Technology, 2001. Disponível em: < http://icawww.epfl.ch/Publications/Wegmann/WegmannN01.pdf>. Acesso em: 11 jan. 2006.

WEB SERVICES INTEROPERABILITY ORGANIZATION (WSI). **Basic profile – version 1.1**. [S.I.], 2004. Disponível em: < http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html>. Acesso em: 24 set. 2005.

WONG, G.; HILTUNEN, M.; SCHLICHTING, R. **A configurable and extensible transport protocol**. Boston: Computer Science Institute – Boston University, 2001. Disponível em: < cs-people.bu.edu/gtw/wong01configurable.pdf >. Acesso em: 09 set. 2005.

Glossário

Agente *Web services*	Um *Web service* é implementado por um agente que é a entidade concreta de hardware ou software que recebe e envia mensagens em *Web services*.
Arquitetura de referência	Uma arquitetura de referência pode ser considerada como uma especificação de sistema em alto nível que define sua estrutura geral (componentes e relacionamentos entre eles), de uma forma sistemática de consistente. Ela é independente de implementação.
Binding	Um *binding* é um contrato entre duas interfaces de objetos, resultante de comportamento mutuamente acordado. Ele habilita comunicação entre dois objetos.
Componente	Um componente é um objeto de software que objetiva a interagir com outros componentes, encapsulando certas funcionalidades. Ele tem uma interface claramente definida e em conformidade com os demais componentes dentro de uma arquitetura.
Coreografia	Composição de processos de negócio (através de *Web services*) onde não existe a figura de um processo mestre (vide orquestração) que controla e coordena os demais processos. Neste tipo de composição, cada processo envolvido tem o conhecimento de que faz parte de uma composição de processos e que precisa interagir com outros processos de maneira ordenada para que a composição resultante tenha sucesso.
Diagramas de caso de uso	Diagramas de caso de uso mostram a interação entre sistemas e entidades externas ao sistema. Essas entidades externas podem ser usuários humanos ou sistemas externos.

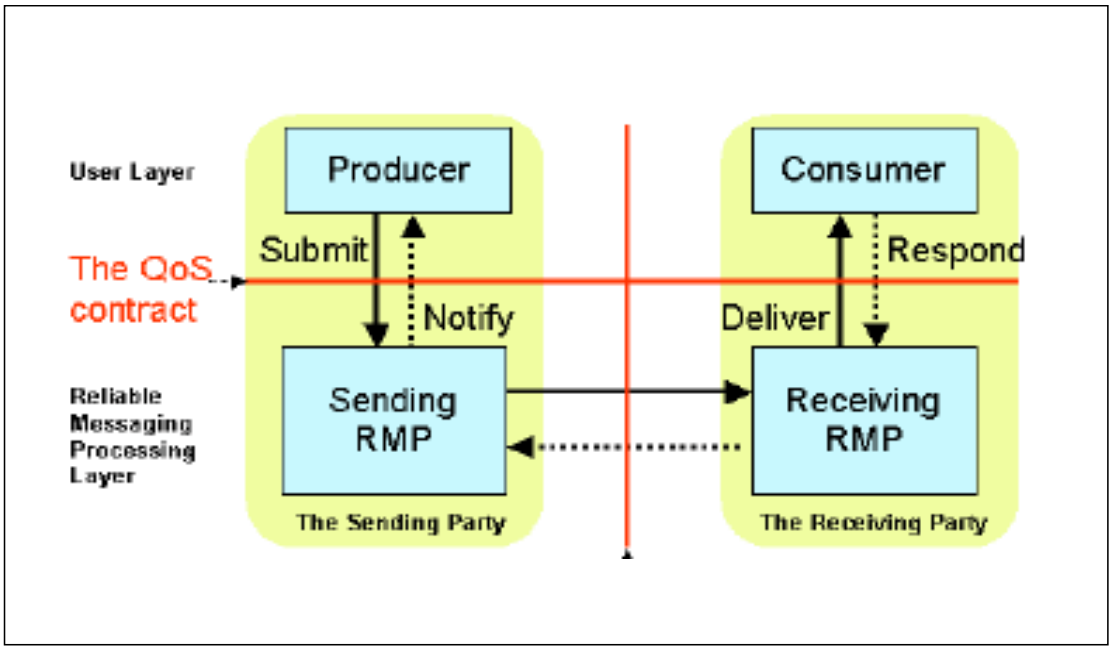
e-business	Negócios eletrônicos – qualquer atividade de negócios baseada na Internet que transformam relações internas e externas, para criar valor e explorar oportunidades de mercado direcionadas pela tendência de “economia conectada”, pela Internet.
e-commerce	Comércio eletrônico - Uso de tecnologias de informação e comunicação para transmissão de informações de negócios e transações de negócios. Esse termo é mais comumente associado com comércio sobre a Internet, mas esta é apenas um das muitas formas avançadas de *e-commerce* que contemplam o uso de tecnologia, aplicações integradas e processos de negócios para interligar empresas.
Endpoint	Uma entidade, recurso ou serviço onde mensagens dos *Web services* são originadas ou entregues.
Event Driven Architecture	Uma arquitetura orientada a eventos define um conceito de projeto e implementação de aplicações e sistemas nos quais eventos são transmitidos entre componentes e serviços de software com baixo acoplamento. Um sistema orientado a evento é tipicamente composto de consumidores e produtores de eventos.
Framework	Esse termo, relacionado à arquitetura de software, representa uma estrutura de suporte para criar uma especificação de arquitetura, geralmente endereçando aspectos correlatos tais como integração, portabilidade, interoperabilidade e distribuição.
Identificador	Identificador é um nome, endereço, etiqueta ou índice de distinção de um objeto em um programa de computador.
Inspeção	Inspeção é uma técnica de análise estática, baseada em um exame visual a produtos de desenvolvimento para detectar

	erros, violações de padrões de desenvolvimento e outros problemas. Inspeção é uma técnica que pode atingir significantes resultados em melhoria de qualidade de software.
Interface	Ponto ou meio de interação com um sistema, quer seja por um usuário ou outro sistema.
Interoperabilidade	A habilidade de um dispositivo ou sistema de trabalhar com outro. Interoperabilidade refere-se a troca de informação de uma forma mutuamente acordada entre as partes.
Metadado	São dados que descrevem dados. Informação sobre dados, incluindo atributos (ex: descrição, comprimento, localização).
Mensagem	No contexto *Web services*, mensagem é uma unidade de dados enviada de um agente para outro. Uma mensagem pode ser parte de uma seqüência de mensagens. Ela possui remetente, destinatário, e tem corpo e cabeçalho.
Mensagem confiável	O conceito de mensagem confiável, em *Web services*, refere-se à garantia de que uma mensagem será entregue e que tanto o remetente quanto o destinatário terão a mesma compreensão do status da entrega.
Middleware	Camada de *software* que “une” e permite que sistemas diferentes (que podem estar em computadores diferentes), executem uma tarefa juntos. Sua principal função é permitir comunicação entre componentes diferentes de *software*.
Namespace	Conjunto de nomes, propostos por órgãos de padronização, utilizados em documentos XML. Prefixos XML de *namespaces* são utilizados para definir, dentro de uma estrutura XML, um escopo específico de tipos de elementos. O *namespace* reservado para extensibilidade do SOAP visando a suporte de mensagem confiável foi proposto pela OASIS e seu prefixo é *wsrm* (um acrônimo de *Web Services Reliable Messaging*).

Reliable Messaging Processor	Corresponde a um componente, contendo objeto processador de mensagens SOAP com mecanismos capazes de realizar entrega confiável de mensagens.
Orquestração	Composição de processos de negócio (por meio de *Web services*) onde existe a figura de um processo central (um processo mestre) que controla e coordena os demais processos. Neste tipo de composição, cada processo participante não tem conhecimento de que faz parte de uma composição de processos, com exceção do processo mestre.
Parâmetro	Uso de constantes, variável ou expressão usada para passagem de valores entre módulos de software.
Sistema	Um sistema é uma entidade de processamento de informação. Uma coleção de componentes organizados para atingir função específica ou conjunto de funções.
SOAP binding	Conjunto formal de regras para transporte de uma mensagem SOAP por um protocolo (protocolo de camada inferior).
Web Services Distributed Management	É um comitê técnico da OASIS que está definindo padrões para gerenciamento de recursos distribuídos, usando *Web services*.
Web Services Notification	É um comitê técnico da OASIS, com propósito de definir um conjunto de especificações que padronizam o modo como *Web services* interagem entre si, usando notificações ou eventos.
Web Services	Sistema de *software* projetado para suportar interação máquina-a-máquina em uma rede que tem uma interface descrita em um formato processável por máquina.

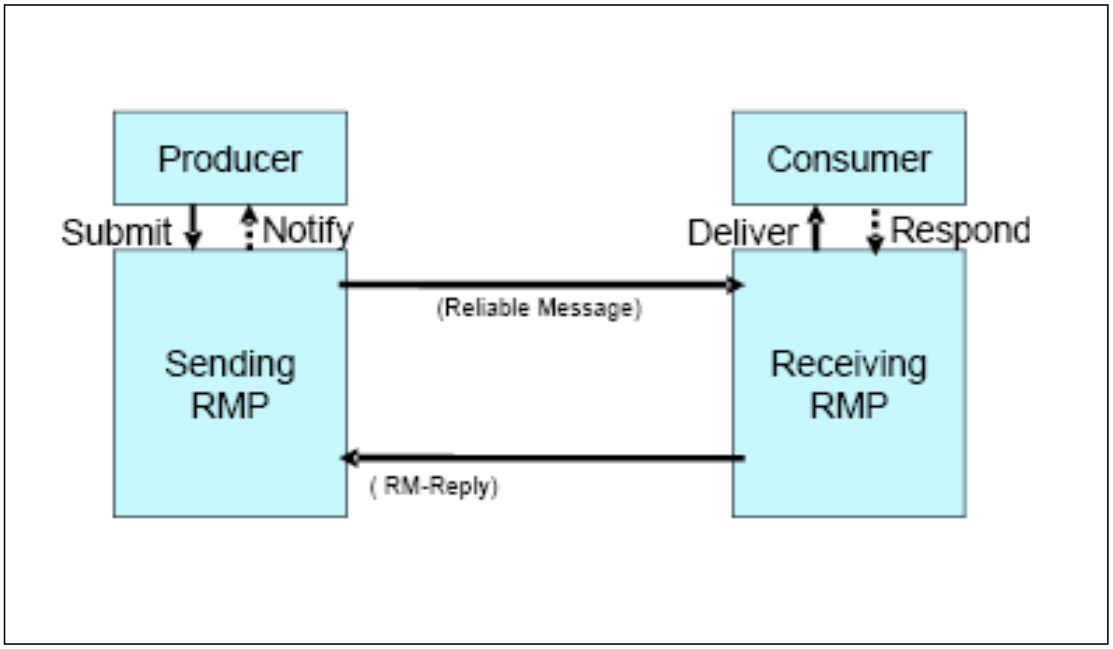
ANEXO A

Componentes produtor e consumidor de mensagens na EPMC *WS-Reliability* (OASIS, 2004b, p.7).



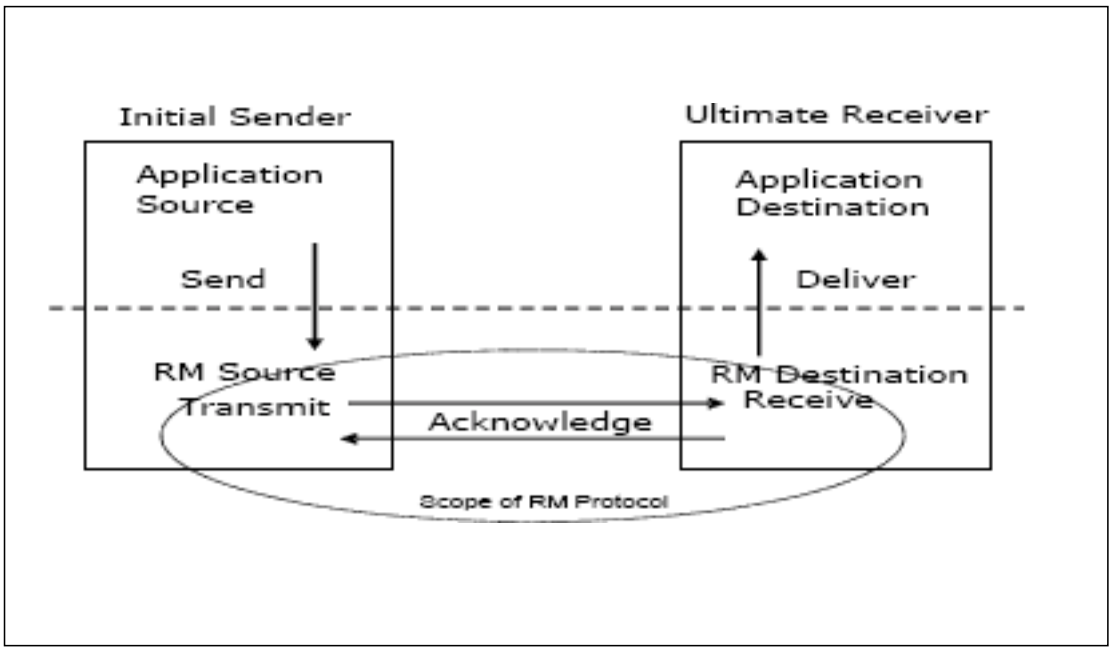
ANEXO B

Modelo de troca de mensagens na EPMC *WS-Reliability* (OASIS, 2004b, p.11).



ANEXO C

Modelo de troca de mensagens na EPMC *WS-ReliableMessaging* (OASIS, 2006c, p.6).



ANEXO D

Seqüência possível de mensagens entre dois *endpoints*, na EPMC *WS-ReliableMessaging* (OASIS, 2006c, p.8).