UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

ROGÉRIO GARCIA DUTRA

Seleção adaptativa e interativa de serviços móveis em ambientes convergentes heterogêneos

São Paulo

2012

ROGÉRIO GARCIA DUTRA

Seleção adaptativa e interativa de serviços móveis em ambientes convergentes heterogêneos

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Doutor em Ciências.

São Paulo

2012

ROGÉRIO GARCIA DUTRA

Seleção adaptativa e interativa de serviços móveis em ambientes convergentes heterogêneos

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Doutor em Ciências.

Área de Concentração:
Departamento de Engenharia de Computação e Sistemas Digitais

Orientador:
Prof. Dr. Moacyr Martucci Júnior

São Paulo

2012

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 23 de Fevereiro de 2012.

Rogério Garcia Dutra

Prof. Dr. Moacyr Martucci Júnior

FICHA CATALOGRÁFICA

Dutra, Rogério Garcia
Seleção adaptativa e interativa de serviços móveis em ambientes convergentes heterogêneos / R.G. Dutra. -- São Paulo, 2012.
130 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Arquitetura orientada a serviços 2.Serviços móveis 3.Algoritmos de prospecção de padrões 4.Internet de serviços 5.Internet do futuro I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

DEDICATÓRIA

A minha esposa Rita e adoráveis filhas Cecília e Isabel.

À memória do meu pai Eurípedes.

A minha mãe Maria.

AGRADECIMENTOS

Ao meu orientador Moacyr Martucci, não somente pela orientação clara e inequívoca, mas pela parceria e confiança que tornaram este trabalho uma realidade.

A minha esposa Rita, com amor, admiração e gratidão por sua compreensão, carinho, presença e incansável apoio ao longo do período de elaboração deste trabalho.

As minhas adoráveis filhas Cecília e Isabel, por todo o tempo de convívio que foi subtraído delas.

A minha irmã Érica pela indicação, por todo o suporte e incentivo essenciais para o início e finalização deste trabalho.

Ao meu irmão Cristóvão e meus pais Eurípedes e Maria, pelo incentivo e suporte por toda a vida.

Aos professores e colegas da USP, que colaboraram direta ou indiretamente com idéias ou sugestões construtivas.

A própria Internet e aos colaboradores anônimos que contribuíram para a motivação deste trabalho.

“O que fazemos em vida, ecoa na eternidade”

Maximus Decimus Meridius

RESUMO

A crescente mobilidade de pessoas e recursos materiais exige um esforço constante na exploração das capacidades oferecidas pelas diferentes tecnologias disponíveis, que convergem para o fornecimento de informações e serviços em qualquer lugar com movimento, a qualquer tempo, e por meio de qualquer dispositivo, objetivos da rede mundial de cooperação e comunicação, conhecida como Internet. Em resposta a essa crescente demanda, a atual Internet está evoluindo do modelo de compartilhamento de informações para o modelo de contribuição e em um futuro próximo, para o modelo de colaboração entre provedores e consumidores, denominada de Internet do Futuro.

Embora a atual Internet seja extraordinariamente bem sucedida, como um meio ubíquo de comunicação, sua atual arquitetura impõe limites para o provisionamento de serviços em ambientes heterogêneos e convergentes, demandando novas soluções que superem os desafios tecnológicos necessários ao estabelecimento da Internet do Futuro. Estas novas soluções basear-se-ão nos princípios da computação orientada a serviços, formando os componentes de um novo arcabouço de serviços para a Internet do Futuro, denominada de Internet de Serviços. Na Internet de Serviços, redes de comunicação móveis, convergentes e heterogêneas serão criadas sob demanda, disponibilizando um elevado número de serviços, funcionalmente similares, porém distintos sob o ponto de vista não funcional, dificultando a seleção dos serviços que melhor atendem o nível qualidade de serviço acordado entre provedores e consumidores.

Este trabalho propõe uma nova solução para o problema de seleção de serviços, combinando algoritmos, usualmente empregados para prospecção de dados, para selecionar serviços de forma dinâmica e interativa, com base em atributos não funcionais, visando suprir as necessidades de mobilidade e colaboração em ambientes convergentes e heterogêneos, como a Internet de Serviços. Desta forma, este trabalho contribui para o projeto de pesquisa da Internet de Serviços, um dos pilares fundamentais para a elaboração da nova arquitetura orientada a serviços, que servirá de arcabouço à construção da Internet do Futuro, possibilitando inúmeras aplicações como Serviços Baseados em Localização e Computação em Nuvem.

Palavras Chave: Arquitetura Orientada a Serviços, Serviços Móveis, Qualidade de Serviço, Seleção Dinâmica e Adaptativa, Algoritmos de Prospecção de Padrões, Internet de Serviços, Internet do Futuro, Serviços Baseados em Localização e Computação em Nuvem.

ABSTRACT

The increasing mobility of people and resources demands additional efforts in exploring new capacities, offered by different technologies, which allows the supply of information and services in any place, any time and through any device, objective of the worldwide collaboration and communication web, known as Internet. Face to this increasing demand, current Internet is evolving from sharing to contribution model and in the near future, to collaboration model between providers and consumers.

Although the current Internet has been extraordinarily successful, as a ubiquitous and universal means for communication, its architecture imposes limits for services deployment in heterogeneous and convergent environments, demanding new solutions to overcome the technological issues for Internet of Future achievement. These new solutions will be based on service oriented computing principles, providing the components of the new service framework for Internet of Future, called Internet of Services. In Internet of Services, convergent and heterogeneous mobile communication networks will be created on demand, providing a huge numbers of services, similar from functional point of view, but very different from the nonfunctional point of view, creating challenges for service selection which fits the service level agreement between providers and consumers.

This work proposes a new solution for service selection process, combining algorithms commonly used for data mining, to perform dynamic and interactive service selection, fulfilling the mobility and collaborative requirements in a convergent and heterogeneous environment, such as Internet of Services. Therefore, this work contributes to Internet of Services research project, one of fundamental pillars to build the service oriented architecture, which will be used as framework for Internet of Future building, allowing many applications such as Location Based Services and Cloud Computing.

Keywords: Service Oriented Architecture, Mobile Services, Quality of Services, Adaptive and Dynamic Selection, Data mining Algorithms, Internet of Services, Internet of Future, Location Based Services, Cloud Computing.

SUMÁRIO

1. INTRODUÇÃO 16
1.1 Motivação 17
1.2 Objetivos 18
1.3 Justificativas 19
1.4 Metodologia 20
1.5 Estrutura do trabalho 21
2. DEFINIÇÕES E CONCEITOS APLICÁVEIS 22
2.1 Internet do Futuro (IoF) 22
2.2 Arquiteturas de *software* orientadas a serviços 26
2.3 Conceitos sobre soluções de intermediação (middlewares) de serviços 32
2.4 Conceitos sobre algoritmos para seleção de serviços 36
2.5 Comentários 44
3. SELEÇÃO DE SERVIÇOS 45
3.1 Apresentação do problema de seleção de serviços 45
3.2 Estado da arte para a seleção de serviços 47
3.3 Parâmetros de QoS percebidos pela rede 49
3.4 Parâmetros de QoS percebidos pelo usuário 53
3.5 Comentários 58
4. A SOLUÇÃO PROPOSTA 59
4.1 Arquitetura de referência da IoS 59
4.2 Subsistema *Descoberta de Serviços* 63
4.3 Algoritmos para seleção adaptativa de serviços 71
4.4 Comportamento do sistema *Descoberta de Serviços* 82
4.5 Processo de seleção adaptativo e interativo de serviços 85
4.6 Comentários 90
5. ESTUDO DE CASO E PROPOSTA PARA PROVA DE CONCEITO 92
5.1 Descrição do estudo de caso e proposta para Prova de Conceito 92
5.2 Artefatos de *software* utilizados na implementação do protótipo 95
5.3 Implementação do processo de seleção de serviços 97
5.4 Comparação dos resultados obtidos com os esperados 103
5.5 Comentários finais sobre a implementação do protótipo 104
6. CONSIDERAÇÕES FINAIS 107
6.1 Conclusões e contribuições 107
6.2 Ineditismo e contribuição ao conhecimento 109
6.3 Proposta de continuidade 110
REFERÊNCIAS BIBLIOGRÁFICAS 112

ANEXO – Serviços utilizados na prova de conceito 116
Consulta 1: economy/ car_price_service.owls 117
Consulta 2: economy/ book_price_service.owls 117
Consulta 3: economy/ dvdplayermp3player_price_service.owls 118
Consulta 4: economy/ 1personbicyclecar_price_service.owls 118
Consulta 5: economy/bookpersoncreditcardaccount_price_service.owls 119
Consulta 6: economy/ maxprice_cola_service.owls 119
Consulta 7: economy/ bookpersoncreditcardaccount__service.owls 120
Consulta 8: economy/ shoppingmall_cameraprice_service.owls 120
Consulta 9: economy/recommendedprice_coffeewhiskey_service.owls 121
Consulta 10: economy/ userscience-fiction-novel_price_service.owls 121
Consulta 11: economy/ preparedfood_price_service.owls 122
Consulta 12: food/ grocerystore_food_service.owls 122
Consulta 13: communication/ title_comedyfilm_service.owls 123
Consulta 14: communication/ title_videomedia_service.owls 123
Consulta 15: education/ researcher-in-academia_address_service.owls 124
Consulta 16: education/ university_lecturer-in-academia_service.owls 124
Consulta 17: education/ governmentdegree_scholarship_service.owls 125
Consulta 18: education/ publication-number_publication_service.owls 125
Consulta 19: education/ novel_author_service.owls 126
Consulta 20: education/ country_skilledoccupation_service.owls 126
Consulta 21: medical/ hospital_investigating_service.owls 127
Consulta 22: travel/ surfing_destination_service.owls 127
Consulta 23: travel/ surfinghiking_destination_service.owls 128
Consulta24:travel/ geographical-Regiongeographical-region_map_service.owls 128
Consulta 25: travel/ surfingorganization_destination_service.owls 129
Consulta 26: travel/ citycountry_hotel_service.owls 129
Consulta 27: travel/ geopolitical-entity_weatherprocess_service.owls 130
Consulta 28: weapon/ governmentmissile_funding_service.owls 130

LISTA DE FIGURAS

Fig. 2.1.1 Pilares de sustentação da Internet do Futuro 23
Fig. 2.4.1.1 Definição formal de um Autômato Adaptativo Finito 37
Fig. 2.4.3.1.1 Modelo de um sistema de inferência difusa do tipo TAKAGI-SUGENO 43
Fig. 3.1.1 Problemas para a seleção de serviços no contexto da Internet do Futuro 46
Fig.3.1.2 Requisitos necessários a etapa de seleção de serviços 47
Fig. 3.3.1.1 Parâmetros para cálculo da disponibilidade de nó da rede 50
Fig. 4.1.1.1 Requisitos funcionais para descoberta de serviços 60
Fig. 4.1.3.1 Subsistemas que compõem a arquitetura da IoS 63
Fig. 4.2.1 Componentes do subsistema *Descoberta de Serviços* 64
Fig. 4.2.4.1 Componentes do subsistema *Motor de Descoberta de Serviços* 67
Fig. 4.2.6.1 Componentes do subsistema *Motor de Descoberta de Serviços Especializado* 69
Fig.4.4.1.1 Diagrama de seqüência de componentes para a funcionalidade `startDiscovery` 83
Fig. 4.4.2.1 Diagrama de seqüência de componentes para a funcionalidade `findServices` 84
Fig. 4.4.3.1 Diagrama de seqüência de componentes para a funcionalidade `lookUp` 85
Fig. 4.5.1 Diagrama de seqüência entre componentes para procura e seleção de serviços para descoberta de serviços 86
Fig. 5.2.1 Solução de *Middleware para Seleção Adaptativa e Interativa* (MSAI) de serviços 96
Fig. 5.3.1 Atividades do processo de seleção proposto segundo componentes do protótipo 97
Fig. 5.3.2 Resultados do serviço SOM: (a) Índice de Davies-Bouldin (b) Matriz de distâncias e (c) Visualização de Agrupamentos 98
Fig. 5.3.3 Indução da Árvore de Decisão a partir das quatro classes de serviços 99
Fig. 5.3.4 Regras difusas obtidas através etapa (4) do processo de seleção 100
Fig. 5.3.5 Estrutura do ANFIS: Entradas, funções de pertinência, regras e saídas 101
Fig. 5.3.6 Avaliação dos resultados após o treinamento do ANFIS 102
Fig. 5.3.7 Superfícies difusas a partir da combinação dos atributos Preço e Atraso de Execução 102
Fig. 5.3.8 Superfícies difusas a partir da combinação dos atributos Disponibilidade e Preço 103

LISTA DE TABELAS

Tabela 2.3.1 Principais soluções de *middleware* adaptativas 34
Tabela 4.1.2.1 Requisitos não funcionais e resultados esperados do processo de descoberta de serviços 61
Tabela 5.1.1 Domínio dos serviços utilizados para prova de conceito 93
Tabela 5.1.2 Requisitos não funcionais e resultados esperados..... 95
Tabela 5.4.1 Comparação dos resultados obtidos com os esperados..... 104
Tabela 5.5.1 Classificação de serviços segundo critérios funcionais e não funcionais 105

LISTA DE ABREVIAÇÕES

AAF	Autômato Adaptativo Finito
ADAPTREE	*Adaptive Tree*
AF	Autômato de estados Finitos
AFTNT	*Adaptive Fuzzy Neural Tree Network*
ANFIS	*Adaptive Networked Fuzzy Inference System*
AOP	*Aspect Oriented Programming*
BMU	*Best Matching Unit*
CAMPE	*Context-Aware Middleware for Anytime, Anywhere Social Networks*
CBD	*Componente Based Development*
CORBA	*Common Object Request Broker Architecture* ®
DCOM	*Microsoft Distributed Component Object Model* ®
e-TOM	*Enhanced Telecom Operations Map*
EU	*European Unit*
EUA	Estados Unidos da América
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IoF	*Internet of Future*
IoS	*Internet of Services*
ISO	*International Organization for Standardization*
ITU	*International Telecommunication Unit*
LSA	Laboratório de Sistemas Abertos
MANET	*Mobile Ad-Hoc Network*
MH	*Mobile Host*
MLP	*Multi-Layered Perceptron*
MOBISOC	*Mobile Middleware for Social Computing Applications*
MPF	*Middleware de Procura Funcional*
MSAI	*Middleware de Seleção Adaptativa e Interativo*
NESSI	*Networked European Software and Services Initiative*
NEXOF-RA	*NESSI Open Framework – Reference Arquitecture*
OASIS	*Advancing Open Standards for the Information Society*
OMG	*Object Management Group*
PoC	*Proof of Concept*
QoS	*Quality of Service*
REF	*Reflection*
RFID	*Radio Frequency Identification*
RMI	*Java Remote Method Invocation* ®
RM-ODP	*Reference Model-Open Distributed Processing*
RMSE	*Root Mean Squared Error*
RNA	*Redes Neurais Artificiais*
SAMOA	*Self-organizing Group Management middleware for Mobile Ad-hoc Networks*
SDP	*Standard Developing Programming*
SLA	*Service Level Agreement*
SOA	*Service Oriented Architecture*

LISTA DE ABREVIAÇÕES (Continuação)

SOAP	*Simple Object Access Protocol*
SOM	*Self-Organizing Map*
UDDI	*Universal Description, Discovery and Integration*
UML	*Unified Modeling Language*
USLA	*User Service Level Agreement*
USP	Universidade de São Paulo
W3C	*World Wide Web Consortium*
WSDL	*Web Services Description Language*
WTA	Workshop de Tecnologia Adaptativa
WWW	*World Wide Web*

1. INTRODUÇÃO

A crescente mobilidade de pessoas e recursos, fator comum aos diversos mercados, exige um esforço constante por parte das empresas e pessoas no conhecimento e exploração das capacidades oferecidas pelas tecnologias disponíveis. O aproveitamento das oportunidades de negócio emergentes e a sua gradual consolidação em novos mercados ou setores de atividade são, cada vez mais, determinados pela adoção correta de estratégias de aliança e relações de parceria, que permitam agregar serviços e partilhar informações, tendo por base uma colaboração estreita entre provedores e consumidores, estimulados pela necessidade de:

- ***Criar empresas ágeis e inovadoras***, capazes de dimensionar e explorar as oportunidades de mercado, dentro de uma perspectiva global, porém adaptadas às demandas locais;
- ***Desenvolver uma gestão efetiva dos recursos***, tirando partido da tecnologia que se adapta a qualquer fase do ciclo de vida de produtos ou serviços;
- ***Promover cooperação entre pessoas e empresas*** locais, regionais ou globais, através da utilização de serviços em qualquer lugar, a qualquer tempo e por intermédio de qualquer dispositivo em movimento.

Em resposta a essa crescente demanda, a mais importante infra-estrutura global de cooperação, conhecida como Internet, foi concebida aproximadamente há 40 anos e desde então tem evoluído constantemente (EFII,2010). Como parte desta evolução contínua, a Internet está em processo de transição do modelo de “compartilhamento” de informações, conhecido como *WEB 1.0*, para o modelo de “contribuição”, conhecido como *WEB 2.0*.

Nos próximos anos, tem-se a expectativa de uma nova transição do modelo de contribuição para um novo modelo colaborativo, denominado de *WEB 3.0*, permitindo que consumidores e provedores colaborem entre si, através de um meio seguro, confiável e eficiente. Neste ambiente, denominado de Internet do Futuro, composto por tecnologias heterogêneas, porém convergentes, serviços deverão adaptar-se às necessidades personalizadas de seus consumidores, dentro de contextos abertos, descentralizados e dinâmicos.

Colaborar com pesquisa para a construção da Internet do Futuro, traduz a principal motivação deste trabalho.

1.1 Motivação

Embora a atual Internet seja extraordinariamente bem sucedida, como um meio ubíquo de comunicação, sua atual arquitetura impõe limites para o provisionamento de serviços em ambientes heterogêneos e convergentes (ZAHARIADIS;et al., 2011), demandando novas soluções que superem os desafios tecnológicos necessários ao estabelecimento da Internet do Futuro. Segundo o fórum de pesquisa da Comunidade Européia (TSELENTIS;et. al, 2009), os principais desafios são:

- Capacidade de interação com serviços, para melhor seleção de serviços, com suporte a qualidade fim-a-fim de serviços sob o ponto de vista da rede e do usuário final, com base em requisitos funcionais e não funcionais;
- Capacidade de adição de novas capacidades, incluído a ativação de novos serviços sob demanda;
- Capacidade de orquestração da segurança e confiabilidade de serviços, com gerenciamento de privacidade e proteção de dados;
- Mobilidade de redes, serviços e dispositivos, com suporte a gestão de energia.

Estas novas soluções basear-se-ão nos princípios da computação orientada a serviços, formando os componentes de um novo arcabouço de serviços para a Internet do Futuro, denominada de Internet de Serviços, visando atender aos principais requisitos (NESSI, 2009):

- **Acessibilidade e interoperabilidade**, garantindo acesso aos serviços disponíveis em qualquer lugar e por intermédio de qualquer tecnologia de acesso;
- **Adaptatividade e proatividade**, nas quais a oferta de serviços deve ser adaptada continuamente à demanda, atendendo a qualidade de serviço fim-a-fim, acordada entre consumidores e provedores de serviços;
- **Aprendizado contínuo,** através do aprendizado interativo de padrões de procura e seleção de um elevado número de serviços, similares do ponto de vista funcional, porém distintos sob o ponto de vista de qualidade de serviço.

A principal motivação deste trabalho é contribuir para o projeto de pesquisa da Internet de Serviços patrocinado pela Comunidade Européia (TSELENTIS;et al., 2009) um dos pilares fundamentais para a elaboração da nova arquitetura orientada a serviços, que servirá de arcabouço à construção da Internet do Futuro, possibilitando inúmeras aplicações, por exemplo, Serviços Baseados em Localização e Computação em Nuvem.

1.2 Objetivos

Visando atender aos principais requisitos necessários à Internet de Serviços, este trabalho propõe uma nova solução para seleção adaptativa e interativa de serviços móveis, em ambientes convergentes heterogêneos, como a Internet do Futuro. Esta solução tem por objetivos:

- Reduzir o universo de escolha de serviços, agrupando a oferta de múltiplos serviços funcionalmente similares, com base em requisitos não funcionais de Qualidade de Serviço (QoS);
- Gerar uma base de regras para seleção da melhor oferta de serviços, com base em requisitos não funcionais de QoS;
- Definir relações hierárquicas entre agrupamentos de serviços, através de um aprendizado supervisionado por máquina *(machine learning*), porém de forma adaptativa e incremental, simplificando a geração de regras que definem a formação destes agrupamentos;
- Tratar incertezas na definição difusa de atributos QoS, adaptando a oferta de serviços às requisições de consumo;
- Complementar o processo de procura de serviços, com base em atributos funcionais, melhorando o processo de *Descoberta de Serviços*.

Visando manter o foco na solução do problema de seleção de serviços, este trabalho não abordará os seguintes temas:

- Questões relacionadas à privacidade e segurança de acesso dos usuários, pressupondo a existência de um ambiente confiável e seguro para troca de informações entre consumidores e provedores de serviços;
- Questões relacionadas à tecnologia de telecomunicações, tais como *handover* vertical e horizontal, ou convergência entre redes móveis e a Internet;
- Questões relacionadas à coreografia e orquestração de serviços móveis.

1.3 Justificativas

Na Internet do futuro, redes de comunicação móveis, convergentes e heterogêneas serão criadas sob demanda, disponibilizando um elevado número de serviços, funcionalmente similares, porém distintos sob o ponto de vista não funcional, dificultando a descoberta dos serviços que melhor atendem o nível qualidade de serviço acordado entre provedores e consumidores.

A descoberta dos serviços disponíveis é composta por processos interdependentes e integrados, denominados de procura e seleção. Ao realizar-se o processo de procura de serviços com base em requisitos funcionais, descobre-se a existência de um conjunto de serviços que satisfaz a estes requisitos, similares do ponto de vista funcional. Simultaneamente ou posteriormente à procura, faz-se necessário a seleção do serviço ou conjunto de serviços que satisfaz requisitos não funcionais.

O processo de seleção é usualmente subestimado ou simplesmente ignorado em várias soluções de descoberta de serviços (MIAN; et.al., 2009), uma vez que a arquitetura de *software* da atual Internet pressupõe a existência de uma infra-estrutura de rede fixa e conhecida a priori, sem garantia da qualidade de serviço fim a fim desde o provisionamento até o consumo do serviço.

A realização inadequada do processo de seleção causa perda de eficiência do processo de procura e uma baixa percepção da qualidade do serviço, sob o ponto de vista do usuário final. Adicionalmente, prejudica-se a interoperabilidade dos sistemas e aplicações existentes, principalmente em ambientes de redes heterogêneas sem fio, dificultando a mobilidade do usuário.

A melhora do processo de descoberta de serviços, possibilitando um processo integrado entre a procura, realizada com base em atributos funcionais, e a seleção realizada com base em atributos de qualidade de serviço, adaptando dinamicamente a oferta de serviços à demanda, através do aprendizado interativo de padrões de provisionamento de serviços, traduz a principal justificativa para esta tese.

A seleção adequada de serviços possibilita a reutilização de serviços, viabilizando a composição de serviços individuais e desta maneira, reduzindo o custo e esforço para a criação e manutenção de novos serviços. Adicionalmente, aumenta-se a probabilidade de atendimento do nível de serviço acordado entre provedores e consumidores, uma vez que a seleção é realizada com base em atributos não funcionais que compõem o acordo de nível de serviço entre as partes.

1.4 Metodologia

A metodologia utilizada neste trabalho consiste em elaborar uma proposta para a prova de conceito, com base na captura de requisitos, projeto segundo o arcabouço da Internet de Serviços e a implementação de um protótipo, cuja viabilidade é comprovada através de um estudo de caso, comparando-se resultados obtidos e esperados.

Para avaliação dos resultados esperados, serão utilizados requisitos não funcionais estabelecidos pela arquitetura de referência da Internet de Serviços (NEXOF-RA-3,2010), provenientes de um cenário de negócios de compras de artigos e serviços, conforme descrito no anexo deste trabalho, abrangendo:

- Qualquer consumidor de serviço que possa definir diferentes necessidades de forma estrita ou difusa, possibilitando a procura e a seleção de serviços com base nestas definições;
- Qualquer provedor de serviço que publique sua oferta de serviços, buscando atender aos níveis de qualidade de serviço sob o ponto de vista dos consumidores.

As premissas, sob o ponto de vista de consumidores e provedores, que devem ser satisfeitas para tornar possível este estudo de caso são:

- Existe um entendimento comum entre consumidores e provedores, sobre os atributos que definem a qualidade de serviço negociado entre as partes;
- O processo de descoberta está baseado em procura semântica dos atributos funcionais e seleção adaptativa e interativa dos atributos não funcionais;
- Os atributos não funcionais conterão ambigüidades que fornecerão oportunidades para negociação do acordo do nível de serviço entre provedores e consumidores;
- Ambas as partes em acordo, devem ser capazes de acessar, se necessário, informações de desempenho do serviço, auxiliando no gerenciamento do acordo do nível de serviço para garantia da qualidade fim-a-fim do mesmo.

1.5 Estrutura do trabalho

Este trabalho foi estruturado em seis capítulos, descritos resumidamente a seguir:

- O capítulo 1 apresenta a introdução do trabalho no contexto da Internet do Futuro, evidenciando os desafios tecnológicos a serem superados pela solução proposta de serviços móveis, descrevendo as motivações, justificativas, objetivos e metodologia;
- O capítulo 2 apresenta a fundamentação teórica, necessária ao desenvolvimento da tese, apresentando uma pesquisa bibliográfica resumida sobre os temas abordados: arquitetura de *software* e arquitetura orientada a serviços (SOA), Internet do Futuro e de Serviços, soluções de *middleware* e algoritmos de prospecção de dados;
- O capítulo 3 apresenta o problema de seleção de serviços, o estado da arte das soluções e propostas encontradas na literatura, bem como os parâmetros de QoS utilizados na solução proposta;
- O capítulo 4 apresenta a solução proposta, descrevendo como os componentes que constituem a solução proposta no contexto da arquitetura de referência da Internet de Serviços (NEXOF-RA-1, 2010).
- O capítulo 5 apresenta o estudo de caso e a proposta para prova de conceito, realizada por intermédio de um protótipo da solução proposta, comparando os resultados obtidos com os resultados esperados;
- O capítulo 6 apresenta a conclusão do trabalho, descrevendo as considerações finais e a verificação das contribuições deste trabalho, face aos resultados obtidos no capítulo anterior. Adicionalmente serão descritas as melhorias e a proposta de continuidade futura da tese.

Este trabalho encerra-se com as referências bibliográficas, citadas ao longo do mesmo, e anexo contendo os serviços e respectivos valores dos atributos de qualidade, utilizados na elaboração do protótipo.

2. DEFINIÇÕES E CONCEITOS APLICÁVEIS

As definições e conceitos apresentados nesta seção fornecem embasamento conceitual sobre os seguintes temas:

- A seção 2.1 - Internet de Serviços – conceitua os pilares constituintes da Internet do Futuro, detalhando o pilar Internet de Serviços, como plataforma de serviços que viabilizará a Internet do Futuro;
- A seção 2.2 - Arquiteturas de *Software* Orientada a Serviços – conceitua os termos Arquitetura Orientada a Serviços (SOA) e Serviços, direcionando o conjunto de decisões sobre o projeto da solução de intermediação de serviços proposta neste trabalho;
- A seção 2.3 - Conceitos sobre soluções de intermediação (*middlewares*) – conceitua as soluções existentes, suas aplicações e deficiências para suporte a arquiteturas abertas e distribuídas, como redes de comunicação móveis;
- A seção 2.4 - Conceitos sobre algoritmos de prospecção de dados – conceitua os algoritmos de prospecção de dados utilizados para a seleção adaptativa de serviços, usando técnicas baseadas em árvores de decisão adaptativas, redes neurais artificiais e lógica difusa (*fuzzy*);
- A seção 2.5 encerra este capítulo, resumindo os principais conceitos necessários à elaboração da proposta da solução de seleção adaptativa de serviços.

2.1 Internet do Futuro (IoF)

A mais importante infra-estrutura mundial de rede de acesso, que conecta pessoas, dados e serviços, conhecida como Internet, foi concebida aproximadamente há 40 anos e desde então tem evoluído constantemente (EFII,2010). Segundo os mesmos autores, observa-se a transição do modelo de “compartilhamento” de informações, conhecido como *WEB 1.0*, para o modelo de “contribuição”, conhecido como *WEB 2.0*, no qual seus usuários (pessoas) geram conteúdos, por exemplo, através de redes sociais (O'REILLY,2009). Nos próximos anos, tem-se a expectativa de uma nova transição para um modelo colaborativo, denominado de *WEB 3.0*, no qual pessoas, máquinas e objetos “colaboram” entre si.

Neste novo ambiente, denominado de Internet do Futuro (IoF), soluções de *software* deverão adaptar-se às necessidades de seus usuários, dentro de contextos abertos, descentralizados e dinâmicos sem perder escalabilidade e desempenho.

Segundo (PAPADIMITRIOU; et al.,2009), a IoF terá como suporte, quatro pilares fundamentais de sustentação, além da infra-estrutura física de rede, como ilustrado na figura 2.1.1:

- *Internet Através e Para as Pessoas* (*Internet By and For People*);
- *Internet de Conteúdo e Conhecimento* (*Internet of Contents and Knowledge*);
- *Internet das Coisas* (*Internet of Things*);
- *Internet de Serviços* (*Internet of Services*).

Fig. 2.1.1 Pilares de sustentação da Internet do Futuro (PAPADIMITRIOU; et al.,2009)

O termo *"Internet Através e Para as Pessoas"* refere-se ao fato da Internet do Futuro buscar atender à crescente demanda por parte de seus principais usuários, ou seja, as pessoas, consolidando o modelo colaborativo da WEB 2.0 (O'REILLY,2009)

O termo *"Internet de Conteúdo e Conhecimento"* refere-se à disseminação de conhecimento local e global, através da geração de conteúdo capaz de ser lido ou processado por pessoas e, principalmente, por máquinas. A WEB 3.0, a geração, o armazenamento e o processamento por máquinas, utilizando anotações semânticas sobre os conteúdos gerados por máquinas ou pessoas permitirá a prospecção de um grande volume de dados, transformando-os em informação útil ao usuário final.

Segundo (CONNER,2010), o termo Internet das Coisas *(Internet of Things, IoT)* refere-se a um conjunto de objetos triviais, ou normalmente utilizados no dia a dia das pessoas, conectados entre si através de redes sem fio, porém utilizando o protocolo padrão da Internet (*Internet Protocol, IP*).

Estes objetos, embora normalmente não tenham a capacidade de comunicar-se entre si, formam uma rede sob demanda (Ad-Hoc) de nós identificados com emissores de rádio freqüência de baixo alcance (*Radio Frequency IDentification*, RFID). Esta rede pode conter uma infinidade de objetos, como livros, calçados, alimentos, entre outros, facilitando sua localização estática ou em movimento.

Cria-se desta forma a possibilidade sensoriamento e gerenciamento de uma infinidade de objetos, como verificação de estoques de forma quase instantânea, acompanhamento de movimentações desde a origem até o seu destino, provisão de abastecimento mais acurado e a conseqüente a geração de novos serviços disponibilizados através de aplicações acessíveis através da Internet de Serviços, como descrito a seguir.

2.1.1 Internet de Serviços (IoS)

Segundo (PAPADIMITRIOU; et al.,2009), o termo “Internet de Serviços” trata, de forma abrangente, como prover e operar serviços de forma pervasiva e ubíqua, através da Internet.

Por tratar-se de um termo abrangente, utiliza-se três domínios para classificar o desenvolvimento da IoS:

- Computação orientada a serviços em larga escala na Internet;
- Serviços proativos e personalizados baseados em contexto;
- Orquestração de serviços.

A computação orientada a serviços tem sido utilizada em escala crescente em diversos campos de aplicações na Internet, como por exemplo, o comércio eletrônico de bens e o processamento distribuído em nuvem (*cloud-computing*).

Baseada nos conceitos da arquitetura orientada a serviços (SOA), apresentados na próxima seção, recursos computacionais físicos ou virtuais, dados e funcionalidades de *software* são encapsulados na forma de serviços, acoplados de forma fraca (*loose coupling)*.

A IoS permitirá um acesso proativo e personalizado a estes serviços, baseado no contexto, na qual o usuário está presente, e não mais de forma reativa ou não personalizada como ocorre usualmente na Internet. Como contexto, entende-se qualquer interação com o meio físico ou virtual, no qual informações a respeito do usuário estão disponíveis, tais como (JONES;et al.,2004):

- Preferências com base no histórico de transações;
- Interações com locais, capturando informações de coordenadas geográficas;
- Interações com outros usuários, através de redes sociais.

A proatividade e personalização de serviços dependerão, fundamentalmente, da detecção dos serviços disponíveis em um determinado contexto, auxiliado pela procura de funcionalidades com base em descrições semânticas, ou seja, no significado ou conceituação dos atributos que descrevem as capacidades destes serviços.

Face à heterogeneidade da infra-estrutura (redes de acesso, plataformas e dispositivos) utilizada pela Internet atual e, conseqüentemente, a Internet do Futuro, torna-se inviável estabelecer uma única ontologia, isto é, uma representação formal dos conceitos que descrevem serviços de uma forma globalmente aceita na Internet, segundo (HERNAN,2001).

Apesar deste fato, o *World Wide Web Consortium* (W3C) criou o conceito de Web Semântica (W3C,2010), cujo objetivo é prover um arcabouço para compartilhar e reutilizar informações (dados) entre diferentes aplicações, empresas ou comunidades de usuários. Este arcabouço permitirá que descrições semânticas de serviços, em um determinado domínio de aplicação, sejam compartilhadas de forma local e não global.

Como conseqüência, a procura de serviços com base em atributos que descrevem funcionalmente estes serviços, resultará em um crescente número de serviços similares. O processo de decisão sobre a escolha do serviço, que melhor satisfaz às necessidades do usuário final, tornar-se-á cada vez mais complexo. Essa complexidade será refletida nas diferentes camadas de serviços distribuídos através da IoS, compreendendo vários tipos de aplicações de *software*, como por exemplo:

- Aplicações em ambientes ubiquos;
- Aplicações em ambientes industriais como manufatura e logística de transportes;
- Aplicações em ambientes financeiros, meios eletrônicos de pagamento e comércio eletrônico;
- Aplicações em saúde e governo eletrônicos.

Com o objetivo de propor uma visão de alto nível desta plataforma global, foi criado um padrão de arquitetura de *software* (AGASSI, et al.,2010), pelo grupo *Networked European Software and Services Initiative* (NESSI, 2009), o qual reúne cerca de 430 organizações de pesquisa privada e acadêmica. Este padrão descreve os principais

componentes que constituem a plataforma de *software*, seus papéis funcionais e os principais requisitos não funcionais que direcionarão o projeto da IoS.

Segundo a agenda de pesquisa do grupo NESSI (2009), para os próximos anos, essa crescente complexidade demandará novas arquiteturas de *software* que possibilitem:

- **Serviços proativos sensíveis ao contexto,** a interação com o usuário será personalizada com base no contexto onde o mesmo se encontra, selecionando serviços segundo a localização do usuário, tempo e sua interação através de ambientes heterogêneos, porém convergentes;
- **Orquestração de serviços**, que permitirá a composição de serviços simples para atender a demandas complexas, combinando fontes de informações heterogêneas como informações sociais, econômicas e geográficas;
- **Aplicação de arquiteturas de *software* orientadas a serviços em escala global:** a Internet orientada a serviços permitirá o acesso a recursos computacionais simples ou complexos, dados ou funcionalidades de *software* na forma de serviços, conforme descrito na próxima seção.

2.2 Arquiteturas de *software* orientadas a serviços

De acordo com a norma *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems* (IEEE1471,2007), o termo "Arquitetura de *Software*" foi definido formalmente como :

> *"Arquitetura é a organização fundamental de um sistema incorporada em seus componentes, seus relacionamentos com o ambiente, e os princípios que conduzem seu projeto e evolução."*

Segundo esta definição, arquitetura de *software* é uma coleção de decisões fundamentais sobre o projeto ou evolução de uma solução de *software*, incluindo seus componentes e colaboração entre os mesmos (interações e comportamentos), visando atender requisitos de qualidade.

Esta norma cita o padrão ISO *Reference Model of Open Distributed Processing* (RM-ODP,1998), que prove conceitos para a definição da arquitetura de *software* para sistemas abertos e distribuídos, escopo deste trabalho, sendo composto por quatro partes:

- Parte 1(ITU-T Rec. X.901 | ISO/IEC 10746-1): Visão geral e guia de uso do modelo de referência, contendo o escopo, justificativas e explicação dos

conceitos chaves, além de diretrizes da arquitetura ODP. Esta parte não é normativa.

- Parte 2 (ITU-T Rec. X.902 | ISO/IEC 10746-2): Fundamentos, os quais contêm a definição para conceitos e um arcabouço analítico para descrições normalizadas de sistemas distribuídos arbitrários. Esta parte introduz os princípios de conformidade, segundo os padrões do ODP, e a forma os quais são aplicados. Esta parte é normativa.
- Parte 3 (ITU-T Rec. X.903 | ISO/IEC 10746-3): Arquitetura, a qual contém as especificações dos requisitos de qualidade que sistemas distribuídos e abertos devem satisfazer. Esta parte utiliza os fundamentos da parte 2, sendo igualmente normativa.
- Parte 4 (ITU-T Rec. X.904 | ISO/IEC 10746-4): Definições semânticas, contendo a formalização dos conceitos de modelagem descritos na parte 2, sendo igualmente normativa.

Adicionalmente foi adotado a linguagem UML (*Unified Modeling Language*™), desenvolvida pelo *Object Management Group*™ (OMG™), como linguagem padrão para documentação e descrição dos artefatos arquitetônicos da arquitetura de *software* (ITU-T Rec. X.906 | ISO/IEC 19793).

Um sistema aberto e distribuído é definido, no contexto destas normas, como:

> *"Um conjunto compreensivo e consistente de padrões internacionais, sob o ponto de vista tecnológico e funcional, que especifica interfaces, serviços e formatos de suporte para alcançar interoperabilidade e portabilidade entre aplicações, dados e pessoas."*

Uma interface é definida, no contexto destas normas, como:

> "*Interface é uma fronteira compartilhada entre dois sistemas, ou entre componentes destes sistemas, ou entre sistemas e usuários."*

Com base na norma (ITU-T Rec. X.906 | ISO/IEC 19793), o modelo de referência ODP influenciou a criação de arquiteturas orientadas a serviços, que herdaram capacidades essenciais, descritas nas normas anteriores:

- **Interoperabilidade**: capacidade de dois ou mais serviços trocar e utilizar informações entre si;

- **Heterogeneidade**: capacidade de operar com diferentes tecnologias de sistemas abertos para implementação, acesso e comunicação de serviços;
- **Federação**: capacidade de operar com diferentes sistemas abertos, estando preparado para negociar a interoperação entre diferentes serviços;
- **Transparência**: capacidade do usuário solicitar serviços, não se preocupando com o comportamento do sistema para atender sua solicitação.

2.2.1 A Arquitetura Orientada a Serviços (SOA)

O termo SOA foi citado, pela primeira vez, em pesquisa publicada pelo instituto de pesquisa *Gartner Group* (SHULTE; NATIS, 1996), definindo este termo como:

> *"SOA é uma abordagem arquitetural corporativa que permite a criação de serviços de negócio interoperáveis que podem facilmente ser reutilizados e compartilhados entre aplicações e empresas."*

O modelo de referência do OASIS (MACKENZIE; et al., 2006), define a essência da arquitetura orientada a serviços, independente da tecnologia utilizada para sua construção:

> *"Um paradigma para organizar e utilizar capacidades distribuídas, que podem ficar sob o controle de diferentes domínios. Este paradigma fornece um meio uniforme para oferecer, descobrir, interagir e utilizar capacidades para produzir os efeitos desejados, de forma consistente com as pré-condições e expectativas mensuráveis*."

Segundo (MACKENZIE; et al., 2006), entidades (pessoas e organizações) criam funcionalidades para resolver ou suportar a solução de problemas, oriundos do relacionamento entre elas, em virtude dos negócios que as mesmas realizam para prover suas necessidades.

Normalmente, não existe uma relação unívoca entre necessidades e funcionalidades, levando por um lado ao agrupamento de várias funcionalidades para satisfazer uma necessidade, e por outro lado, a utilização de uma funcionalidade para satisfazer várias necessidades. A arquitetura SOA traz, como principal vantagem, um arcabouço para relacionar necessidades e funcionalidades, visando o atendimento das mesmas.

Este arcabouço apóia-se em conceitos fundamentais que descrevem os componentes e artefatos da arquitetura SOA, com o conceito de *"serviço"* descrito na próxima seção.

2.2.2 Conceitos sobre serviços

Segundo o modelo de referência da arquitetura SOA do OASIS (MACKENZIE; et al., 2006), um serviço é definido por:

> *"Um mecanismo para permitir acesso a uma ou mais capacidades, nas quais o acesso é provisionado usando uma interface definida e satisfaz a restrições e políticas especificadas em sua descrição."*

Em geral, as entidades (pessoas e organizações) que provisionam capacidades, são denominadas de *provedores de serviços*, ou simplesmente, *provedores* para satisfazer as necessidades de outras entidades denominadas de *consumidores de serviços*, ou simplesmente *consumidores*. Neste contexto, *usuário final* representa a última instância como consumidor de serviços. Ambos, provedores e consumidores, são denominados como *participantes dos serviços*, quando referenciados em conjunto.

No contexto da arquitetura SOA, adota-se os seguintes princípios:

- Serviços são **reutilizáveis**, permitindo o reuso por diferentes sistemas;
- Serviços compartilham um **contrato formal**, definido como uma coleção de um ou mais documentos que descrevem suas funcionalidades;
- Serviços possuem **acoplamento fraco**, minimizando dependências no relacionamento com demais serviços;
- Serviços **abstraem a lógica de implementação,** exibindo somente informações descritas no contrato formal;
- Serviços são **capazes de composição,** formando um serviço composto por intermédio de serviços simples;
- Serviços são **autônomos**, pois controlam de forma independente sua lógica de implementação;
- Serviços devem possuir a **capacidade de serem descobertos**, através de mecanismos de procura e seleção.

A implementação mais usual de serviços, segundo a arquitetura SOA, é através de *Web Services*, cujas definições são dados pelo *World Wide Web Consortium* (W3C,2010), que trata-se de um consórcio para pesquisa e definição de padrões para a Internet.

A definição de *Web Services*, segundo o documento informativo (não normativo) "*Web Service Architecture*" (BOOTH;et al.,2004):

> "*Web Service é um sistema de software projetado para suportar a interação máquina-para-máquina de forma interoperável sobre a rede (Internet).*"

Complementando esta definição, *Web Services* são implementados através de padrões definidos pelo W3C:

- *Web Services Description Language* (WSDL): segundo (BOOTH;et al,2007), trata-se uma linguagem que descreve a interface pública destes serviços;
- *Universal Description, Discovery and Integration* (UDDI): segundo (CLEMENT; et al.,2007), trata-se um repositório de descrições das interfaces públicas utilizando a linguagem WSDL;
- *Simple Object Access Protocol* (SOAP): segundo (GUDGIN;et al.,2007), trata-se de um protocolo de troca de mensagens para acesso aos métodos e funcionalidades dos serviços, através da interface pública dos mesmos.

Segundo (BOOTH; et al.,2004), *Web Services* são mais apropriados para aplicações projetadas segundo a arquitetura SOA, na qual:

- Confiabilidade e desempenho não podem ser garantidos;
- Não é possível garantir a distribuição de serviços para todos os consumidores e provedores de uma única vez;
- Os componentes de sistemas distribuídos executam suas funções em diferentes plataformas;
- As aplicações existentes devem ser expostas à Internet, encapsuladas como serviços.

2.1.3 Qualidade de Serviço (QoS)

O *Reference Model Open Distributed Processing* (RM-ODP,1998) refere-se à Qualidade de Serviço (QoS) como:

"Um conjunto de requisitos de qualidade no comportamento coletivo de um ou mais objetos."

Segundo (SERRA, 2007), QoS é definida sob o ponto de vista de sistemas de informação e sob a percepção do usuário final em um ambiente de convergência tecnológica, como a qualidade que deve ser apresentada ao usuário final pelos serviços com conteúdo multimídia, interativos e móveis oferecidos por uma provedora de serviços. Sendo que essa qualidade é obtida através de um conjunto de atributos que devem atender a especificação do serviço. Ainda segundo (SERRA, 2007), são apresentadas definições para os seguintes termos utilizados nesta tese:

- **Usuário final** é a pessoa que contratou o serviço;
- **Provedor de serviço** é responsável por disponibilizar serviços ao usuário final;
- **USLA (*User Service Level Agreement*)**: trata-se um documento formal de acordo entre o usuário e a provedora de serviço, que deve definir os níveis de qualidade de serviço percebida pelo usuário final, os tipos de serviços e o que mais for pertinente à contratação do serviço, de acordo com a legislação e regulamentação vigente.

As definições acima pressupõem a existência de um provedor de serviço, conhecido a priori, possibilitando o estabelecimento de um USLA. Em (SAKURAI,2010), por exemplo, foi proposto um modelo de negócios com foco no usuário final, no qual:

- O provedor de serviços é responsável pela comercialização dos serviços ao usuário final, cujo objetivo é oferecer serviços dentro da necessidade, qualidade e custo-benefício esperado pelo usuário final, sem que o mesmo se preocupe com a tecnologia a ser utilizada.
- O provedor de serviços deverá realizar contratos de fornecimento de serviços com as demais provedoras, como provedor de conteúdo, acesso e infra-estrutura.

Usualmente segundo (SAKURAI,2010), utiliza-se uma solução de mediação (*middleware)* distribuída entre as aplicações, permitindo a integração das mesmas, conforme apresentado na próxima seção.

2.3 Conceitos sobre soluções de intermediação (middlewares) de serviços

Informalmente, um *middleware* ou mediador tem por objetivo facilitar ou intermediar a comunicação entre dois componentes, potencialmente distribuídos por diferentes redes de acesso.

Tradicionalmente, sob a ponto de vista de arquitetura de *software*, um *middleware* é um componente de *software* que encapsula um conjunto de serviços, dentro do paradigma de programação orientada a objetos, que suportam interoperabilidade entre diferentes tecnologias e plataformas de *software*. Exemplos de arquiteturas de componentes distribuídos, baseadas no paradigma de orientação a objetos, que proveram soluções de middleware são:

- *OMG Common Object Request Broker Architecture* ® (CORBA,2007);
- *Microsoft Distributed Component Object Model* ® (DCOM,2007);
- *Java Remote Method Invocation* ® (RMI,2007).

Entretanto, estas soluções de *middlewares* tradicionais possuem uma capacidade de adaptação bastante limitada, segundo (SADJADI, 2009), requerendo diferentes paradigmas de programação, tais como:

- **Reflexão computacional** (*Computacional Reflection*, REF), segundo (MAES,1987) refere-se à capacidade de um programa alterar seu próprio comportamento, expondo parte das funcionalidades que permitirão adaptação em tempo de execução do *middleware*;
- **Baseado em componentes** (*Component Based*, CBD), segundo (SZYPERSKI,1999) refere-se à capacidade de utilizar componentes para projetar e produzir um novo *middleware* em tempo de projeto do mesmo;
- **Orientado a aspectos** (*Aspect Oriented Programming*, AOP), segundo (KICZALES,1997) refere-se à capacidade de segregar os requisitos que norteiam o projeto de *middleware*, como por exemplo: adaptação a restrições de qualidade, consumo de energia, segurança, tolerância a falha;
- **Baseado em padrões de *software*** (*Standard Development Based, SDP*), segundo (SCHMIDT,2001) refere-se à reutilização de um conjunto de padrões de código de programação, previamente testados em várias implementações, para compor uma solução mais complexa a medida da necessidade.

Ainda segundo (SADJADI, 2009), estes paradigmas de programação deram origem a inúmeras soluções de *middleware*, conforme tabela 2.4.1, na qual as abreviações na coluna compatibilidade significam: C (CORBA), R (RMI) e D (DCOM) e as abreviações dos paradigmas de computação significam: REF (Reflexão), CDB (Baseado em Componentes), AOP (Baseado em Aspectos) e SPD (Baseado em padrões).

Segundo (SADJADI, 2009), as soluções adaptativas de *middleware* foram subdivididas, segundo o domínio de aplicação, em três grupos:

- ***Middlewares* Orientados a Qualidade de Serviço** (*QoS Oriented*), suportando aplicações que estabelecem e procuram garantir níveis aceitáveis de QoS;
- ***Middlewares* Seguros** (*Dependable*), suportando aplicações distribuídas de missão crítica, que não podem falhar, tais como aplicações militares e médicas;
- ***Middlewares* Embarcados** (*Embedded*), suportando aplicações que requerem pouca memória para sua execução, como aparelhos de telefonia móvel, controladores industriais sem fio.

A solução proposta neste trabalho pode ser classificada no grupo de soluções de *middleware* orientadas a QoS, que será subdivido em quatro subgrupos, descritos a seguir:

- **Tempo Real** (*Real-Time*), suportando aplicações que demandam ações instantâneas, permitindo falhas durante ações de adaptação (*soft middleware*) ou não (*hard middleware*);
- **Orientada ao Fluxo de Dados** (*Stream-Oriented*), suportando aplicações que demandam fluxo contínuo de dados, tais como: aplicações multimídia, telefonia sobre IP (Internet Protocol);

- **Orientada à Reflexão** (*Reflection-Oriented*), suportando aplicações orientadas à qualidade de serviço, segundo o paradigma de reflexão computacional, como mencionado anteriormente;
- **Orientada ao Aspecto** (*Aspect-Oriented*), suportando aplicações orientadas a qualidade de serviço, segundo o paradigma de orientação a aspectos, como mencionado anteriormente.

Tabela 2.3.1 Principais soluções de *middleware* adaptativas (SADJADI, 2009)

Middlewares Adaptativos			REF	CBD	AOP	SDP	Compatibilidade
Orientados a Qualidade de Serviço (*QoS Oriented*)	Tempo Real (*Real-Time*)	ACE				X	
		TAO				X	C
		DynamicTAO	X			X	C
		CIAO		X		X	C
		Orbix	X			X	C
	Orientada ao Fluxo de Dados (*Stream-Oriented*)	OpenORB	X	X			
		OpenORB v2	X	X			D
		Squirrel		X		X	C
		MetaSockets	X	X	X	X	
	Orientada a Reflexão (*Reflection-Oriented*)	FlexNET	X	X			C
		OpenCorba	X				C
	Orientada a Aspecto (*Aspect-Oriented*)	QuO			X	X	C,R
		AspectIX			X		C,R
Seguros (Dependable)	Comunicação Confiável (*Reliable Communication*)	Isis					
		Horus					
		Ensable					
		Totem					
		Rocks					
		Racks					
	Tolerante a Falha (*Fault-Tolerant*)	Electra					C
		Orbix+Isis	X			X	C
		OGS				X	C
		Eternal				X	C,R
		IRL	X			X	C
		FTS	X			X	C
		FRIENDS	X				C
	Equalizador de Carga (*Load-Balancer*)	TAO-LB	X			X	C
Embarcados (*Embedded*)	Mínimo (*Minimum*)	Orbix/E	X				C
		PersonalJava	X				R
		EmbeddedJava	X				R
	Intercambiável (*Swappable*)	UIC	X	X		X	C,R,D
		ZEN	X	X		X	C

As demais descrições dos subgrupos pertencentes aos tipos Seguro (*Dependable)* e Embarcado (*Embedded)*, são encontrados em (SADJADI, 2009), na qual os *middlewares* do tipo Seguro (*Dependable)*, foram ainda subdivididos em três grupos, descritos a seguir:

- ***Middleware* de Comunicação Confiável *(Reliable Communication)***, provêem serviços de comunicação confiáveis, suportando domínios de aplicações de computação móvel, com alta disponibilidade e tolerância a falhas;
- ***Middleware* Tolerante a Falha *(Fault-Tolerant)***, permite que aplicações continuem a operar mesmo após a ocorrência de falhas, através da replicação de serviços. Para manter estas réplicas consistentes, utiliza-se serviços de *middlewares* confiáveis, como mencionado anteriormente;
- ***Middleware* Equalizador de Carga *(Load-Balancer)***, permite que aplicações distribuídas continuem a operar sobre alta demanda, buscando o balanço de carga.

Soluções de *middlewares* pertencentes ao tipo Embarcado (*Embedded)*, são subdivididos em dois subgrupos, como descrito a seguir:

- ***Middleware* Mínimo *(Minimum)***, permite execução do código fonte ocupando o mínimo da capacidade computacional do dispositivo que o executa;
- ***Middleware* Intercambiável *(Swappable)***, permite que partes opcionais do *middleware* sejam executadas ou não, conservando a capacidade computacional do dispositivo que o executa.

Apesar do aumento da capacidade adaptativa em ambientes distribuídos, estas soluções não foram projetadas para atender a requisitos de procura de serviços móveis, demandando novas soluções de *middleware*, tais como:

- **MOBISOC** (*Mobile Middleware for Social Computing Applications*): segundo (GUPTA;et al.,2009), trata-se de um *middleware* cuja estratégia de procura utiliza o relacionamento entre pessoas e seu contexto;
- **SAMOA** (*Self-organizing Group Management middleware for Mobile Ad-hoc Networks*): segundo (BOTAZZI; et al.,2008), trata-se de um *middleware* cuja estratégia se baseia na técnica de auto-organização de grupos de serviços segundo atributos funcionais previamente definidos;
- **CAMPE** (*Context-Aware Middleware for Anytime, Anywhere Social Networks*): segundo (BOTAZZI; et al.,2007), trata-se de um *middleware* cuja estratégia de procura se baseia na percepção do contexto do usuário.

Essas soluções, entretanto, não foram projetadas para realizar o processo de seleção de serviços, atendendo aos requisitos de adaptabilidade, proatividade e aprendizado contínuo, necessários à seleção dinâmica de serviços móveis em ambiente convergentes heterogêneos, como as redes móveis Ad-Hoc.

Nas redes móveis Ad-Hoc (*Mobile Ad-Hoc Networks, MANET*), diferentemente das redes com estrutura estática conectada através de fios, novas dificuldades são criadas para suportar o USLA acordado com o usuário final, tais como:

- Imprevisibilidade das propriedades das ligações dos nós da rede, na qual a qualidade do sinal, bem com a largura de banda e atrasos de propagação na rede não podem ser previstos;
- Mobilidade dos nós que constituem a rede criam uma topologia mutável, através de ligações intermitentes e oportunísticas;
- Recursos limitados dos nós que a constituem, tais como: capacidade de processamento e autonomia da bateria do dispositivo móvel.

Estas dificuldades implicam no aumento da complexidade computacional para selecionar um conjunto de serviços que, apesar de atender a requisitos funcionais, dificilmente satisfazem a requisitos não funcionais.

2.4 Conceitos sobre algoritmos para seleção de serviços

Nas próximas subseções deste capítulo, serão descritos algoritmos de prospecção de dados, os quais serão utilizados como componentes de uma solução de *middleware* para seleção dinâmica e adaptativa de serviços móveis, conforme a arquitetura proposta neste trabalho.

2.4.1 Árvores de Decisão

O problema de indução incremental de árvores de decisão para atributos discretos pode ser resolvido através da aplicação da tecnologia adaptativa, utilizando um dispositivo adaptativo descrito no algoritmo ADAPTREE (PISTORI;NETO,2002).

Um dispositivo adaptativo (PISTORI; et al.,2006) é constituído por duas partes:

- A primeira por um dispositivo definido através de regras não adaptativas, em particular, algum autômato, gramática ou qualquer outro dispositivo descrito

através de um conjunto finito de regras estáticas, denominado de dispositivo subjacente;

- A segunda por um dispositivo adaptativo, cuja conexão ao formalismo subjacente proporciona-lhe todos os recursos complementares, necessários para a obtenção das propriedades responsáveis pela modificação autônoma, que caracteriza a adaptatividade.

Um autômato adaptativo (PISTORI; et al.,2006) de estados finitos (AAF) é um dispositivo adaptativo, que estende o poder de expressão do autômato de estados finitos (AF), através da capacidade de modificar a sua própria estrutura, por intermédio da aplicação de regras adaptativas. Formalmente, um autômato adaptativo de estados finitos $\boldsymbol{M}$ foi definido (PISTORI;NETO,2002) pelas expressões descritas no quadro 2.4.1.1, a seguir:

$\mathrm{M} = \langle Q, \Sigma, q_0, F, \delta, Q_\infty, \Gamma, \Pi \rangle$

onde:

$Q \subseteq Q_\infty$ é um subconjunto finito de um conjunto, possivelmente infinito, de estados.

Σ é o alfabeto de entrada, finito e não vazio.

$q_0 \in Q$ é o estado inicial do autômato.

$F \subseteq Q$ é o conjunto de Estados Finais.

$\delta \subseteq (Q_\infty \times (\Sigma \cup \{\epsilon\}) \times Q_\infty)$ é a relação de transição.

Os três últimos elementos de M, responsáveis pela adaptabilidade do sistema, são definidos a seguir:

Q_∞ é um conjunto, possivelmente infinito, de estados.

$\Gamma \subseteq (\{+,-\} \times (Q_\infty \times (\Sigma \cup \{\epsilon\}) \times Q_\infty))$ é o conjunto de ações primitivas que podem ser executadas pelo AAF: incluir (+) ou excluir (-) transições [4].

$\Pi : (Q_\infty \times (\Sigma \cup \{\epsilon\}) \times Q_\infty) \rightarrow \Gamma^*$ é uma função que associa a cada transição em δ uma seqüência de ações adaptativas primitivas. Podemos associar uma seqüência vazia a uma determinada transição para indicar que esta é uma transição "normal", não adaptativa.

Fig. 2.4.1.1 Definição formal de um Autômato Adaptativo Finito

Neste Autômato Adaptativo Finito, para toda a transição temos:

$$x \in \delta, \; na\ qual\ \Pi(x) = \epsilon \qquad \text{(eq. 2.4.1.1)}$$

O autômato adaptativo se reduz, funcionalmente, a um simples autômato de estados finitos. Nos outros casos, antes de executar a transição, o AAF deverá efetuar as alterações

adaptativas sugeridas pelas ações adaptativas associadas à mesma. Após executar as ações adaptativas, o AAF passa a operar com sua nova estrutura.

O dispositivo adaptativo, no qual o *ADAPTREE* se baseia, pode ser visto como um AAF classificador, estendido para trabalhar com mais de duas classes de serviços. Neste contexto, a cada estado final é associado um elemento, que corresponderá a uma das classes possíveis. Ao receber um exemplo de treinamento, o *ADAPTREE* cria um caminho ligando o estado inicial do autômato ao estado final, correspondente à classificação deste exemplo.

Formalmente, dado ***n*** conjuntos disjuntos:

$A_1, A_2, \dots, A_n$

O conjunto de treinamento é composto por:

$$T = \{w_1, w_2, \dots, w_m\} \qquad \text{(eq. 2.4.1.2)}$$

No qual:

$$|w_i| = n,\ w_i = \alpha_1\alpha_2 \dots \alpha_n\ com\ \alpha_i \in A_i \qquad \text{(eq. 2.4.1.3)}$$

Para:

$1 \leq j \leq n$

$1 \leq i \leq m$

O último símbolo da cadeia é utilizado para representar a classe do exemplo, sendo que |***An***|, determina a quantidade de classes do problema e (***n-1)*** é o total de atributos (ou variáveis) a serem considerados no processo de aprendizagem.

Os conjuntos $A_1, A_2, \dots, A_n$, representam os domínios, isto é, os valores permitidos para cada atributo. Para alternar o classificador do modo de treinamento para o modo de classificação, basta fornecer cadeias de caracteres de tamanho ***(n-1)***, excluindo dessa forma as classificações de treinamento conhecidas a priori.

Caso não seja possível determinar, sintaticamente, a classe de uma cadeia de entrada, o *ADAPTREE* utiliza o mecanismo estatístico **ID3** (CHENG;et al.,1988), que fornecerá como resposta, uma estimativa baseada no ganho de entropia de informação, definindo a ordem dos atributos utilizada nesta classificação.

Porém, segundo (QUINLAN,1990), um atributo com muitos valores possíveis terá uma dispersão maior na distribuição de probabilidades desses valores, conseqüentemente maior ganho de entropia de informação. Para evitar esta distorção, que favorece o atributo

com maior número de valores, introduziu-se no algoritmo **C4.5** (QUINLAN,1992), sucessor do **ID3**, o conceito de Razão do Ganho de Entropia de Informação.

O método **C4.5** tem sido empregado para construir árvores de decisão (*Decision Tree*) que implementam classificadores de elevado desempenho. Contudo, este algoritmo só permite classes previamente definidas para classificação de atributos na fase de treinamento, não tendo a capacidade de interpolar ou deduzir novos padrões por inferência ou tratar dados imprecisos ou incertos. Como mostrado em (BRODLEY;UTGOFF,1992), o **C4.5** somente pode delimitar hiperplanos paralelos aos eixos coordenados, fato que, em um espaço de atributos contínuos, implicaria em crescimento exponencial da árvore de decisão resultante.

Diferentemente do **C4.5**, o ADAPTREE não particiona o conjunto de exemplos, ou seja, o princípio de otimização é global e não local. As árvores de decisão geradas pelo ADAPTREE não possuem um número excessivo de ramificações, devido a esta característica.

Adicionalmente, como o ADAPTREE é um algoritmo incremental, no qual novos exemplos podem ser incorporados durante o processo de classificação, resolvendo adequadamente o problema de variação temporal de padrões, muito comum em atividades de prospecção de dados. A combinação do ADAPTREE e do mecanismo **C4.5**, permite maximizar a vantagem de ambos algoritmos, minimizando as desvantagens citadas.

No entanto, ambos ADAPTREE e **C4.5** necessitam de um conjunto de treinamento para geração de uma árvore de decisão, cujo número de nós entre a raiz e folhas não seja excessivamente grande. Dessa forma, o ADAPTREE e o **C4.5** são algoritmos que trabalham em modo supervisionado durante a fase de treinamento, ou seja, não são capazes de definir o número de classes que particionam o conjunto de treinamento.

Por outro lado, *Redes Neurais Artificiais* (RNA) têm sido empregadas em tarefas de classificação para determinação de padrões em modo supervisionado ou não supervisionado. Dessa forma, classes complexas podem ser prospectadas através dos dados, gerando árvores de decisão de forma mais rápida, eficiente e de simples visualização.

Evidentemente que o desempenho e qualidade dos padrões adquiridos estão diretamente correlacionados com o tipo e a arquitetura escolhida para a modelagem da RNA, como descrevem os estudos realizados comparando-se a arquitetura M*ulti-Layered Perceptron* (MLP) com árvores de decisão (QUINLAN,1990).

A seguir, será apresentado um algoritmo de RNA adaptativo, porém não baseado no paradigma de autômatos adaptativos, cuja vantagem é não necessitar de um conjunto de treinamento classificado a priori, como o ADAPTREE.

2.4.2 Redes Neurais Artificiais

O algoritmo *Self-Organizing Map* (SOM), desenvolvido por Teuvo Kohonen (KOHONEN,1995), é um dos modelos mais populares de RNA. O algoritmo SOM utiliza aprendizado competitivo e não supervisionado, implicando em um treinamento direcionado exclusivamente pelos dados, sendo que os neurônios que constituem o mapa competem entre si para adquirir padrões dos dados, se aproximando deles.

Segundo (KOHONEN,1995), algoritmos supervisionados, como o *Multi-Layered Perceptron* (MLP), requerem uma classificação pré-definida para cada vetor de treinamento, além de depender fundamentalmente do número de camadas internas (*hidden units*) para um aprendizado com baixo erro de classificação e bom desempenho, limitações que não ocorrem no algoritmo SOM.

No algoritmo interativo SOM (VESANTO;et al.,1999), cada neurônio ***i*** é representado por um vetor *n*-dimensional de pesos, conhecido como vetor protótipo ou vetor de referência $\boldsymbol{m_i}$, segundo eq. 2.4.2.1:

$$m_i = [m_{i1}, \dots, m_{in}]^T \qquad \text{(eq. 2.4.2.1)}$$

No qual:

- *n* é igual ao número de dimensões do vetor de entrada;
- m_{in} é o *n*-ésimo peso do vetor m_i.

No algoritmo SOM, as relações topológicas e o número de neurônios são previamente fixados durante a fase de estruturação da rede. O número de neurônios deve ser usualmente escolhido como o maior possível, assim como o tamanho da vizinhança de forma a buscar um equilíbrio entre desempenho computacional e um bom grau de generalização e granularidade.

Antes da etapa de treinamento não supervisionado do algoritmo SOM, valores são escolhidos para os pesos componentes dos vetores de referência.

Em cada passo do treinamento, um vetor de entrada de amostra ***x*** dos dados é escolhido aleatoriamente e uma medida de similaridade é calculada entre este e todos os vetores de referência do mapa do algoritmo SOM. A unidade do mapa que mais se assemelha, também conhecida na literatura como BMU (*Best Matching Unit*), denominada aqui como

unidade vencedora ***c***, é a unidade que possui a menor medida distância, tipicamente distância Euclidiana, segundo a equação 2.4.2.2:

$$\|x - m_c\| = \min\|x_i - m_i\| \qquad \text{(eq. 2.4.2.2)}$$

Na qual o símbolo "|| ||" denota a medida de distância Euclidiana.

A regra de atualização para o vetor de referência de uma unidade ***i*** do algoritmo SOM será expressa pela eq. 2.4.2.3:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \qquad \text{(eq. 2.4.2.3)}$$

Na qual:

- ***t*** denota o instante de tempo ou passo do treinamento;
- ***x(t)*** o vetor de entrada selecionado aleatoriamente dos dados em um instante ***t***;
- $\boldsymbol{h_{ci}(t)}$ a relação de vizinhança ao redor de uma BMU ***c*** em um instante ***t***.

Apesar do algoritmo de treinamento ser relativamente simples, o tratamento matemático para obtenção de demonstrações de convergência é bastante complexo. Segundo (VESANTO;et al.,1999), a convergência em um mapa unidimensional ocorre na maioria dos casos.

Entretanto, a presença de imprecisão, incerteza ou ruído nos dados pode reduzir drasticamente o desempenho do treinamento não supervisionado do algoritmo SOM, da mesma forma que induz árvores excessivamente extensas, utilizando os algoritmos ADAPTREE e **C4.5**. Para tratar de forma eficaz a incerteza nos dados, torna-se necessário a utilização de um mecanismo de inferência com base em lógica difusa, denominado simplesmente de inferência difusa.

2.4.3 Lógica Difusa

A lógica difusa (*fuzzy*) possibilita que seja abordado, de forma mais adequada, o problema referente à representação e manipulação de imprecisão ou incerteza.

Os sistemas baseados em lógica nebulosa foram criados por Zadeh (ZADEH,1965), fundamentando-se na representação e manipulação de informações incertas e imprecisas tão

comuns no cotidiano humano. Expressões tais como “quase”, “muito” e “pouco” representam este tipo de imprecisão, que usualmente não pode ser tratada pelos sistemas da lógica clássica, de forma simples.

Os sistemas especialistas *fuzzy* utilizam um conjunto de regras do tipo “*Se-Então*”, baseadas em variáveis nebulosas. Primeiramente as variáveis de entrada sofrem um processo de “*fuzzificação”*, ou seja, os conjuntos nebulosos das variáveis lingüísticas de entrada são ativados. Terminado este processo, efetua-se a inferência sobre o conjunto de regras nebulosas obtendo os valores das variáveis de saída. Finalmente, as variáveis de saída sofrem um processo de “*defuzzificação*”.

Este processo consiste em converter os dados nebulosos para valores numéricos precisos. Para isto são utilizadas várias técnicas, tais como valor máximo, média dos máximos, média local dos máximos, centro de gravidade, ponto central da área, entre outros. Neste trabalho, foi utilizado o método de chamado *Takagi-Sugeno* (SUGENO,1985), ou simplesmente *Sugeno*, cuja saída representará um valor constante, representando uma classe definida pela RNA do tipo SOM.

Na tese, o algoritmo *Adaptive Network-based Fuzzy Inference Systems* (ANFIS) (JANG, 1993) do tipo Takagi-Sugeno (SUGENO,1985), terá sua formulação descrita.

2.4.3.1 Sistemas de Inferência Difusa

Inferência difusa é o processo de modelagem do relacionamento entre variáveis de entrada e saída, utilizando lógica difusa. Sistemas baseados em inferência difusa têm sido utilizados em diferentes áreas de aplicação: automação e controle, sistemas especialistas e suporte a decisão, análises de decisão e classificação de dados.

Estes sistemas podem ser classificados em dois tipos equivalentes: MAMDANI (MAMDANI; ASSILIAN,1975) e TAKAGI-SUGENO (SUGENO, 1977). O primeiro tipo de sistema de inferência difusa tem por objetivo sintetizar um conjunto de regras obtidas através de operadores humanos, que controlavam uma caldeira a vapor. Apesar do tipo MAMDANI representar sistemas de inferência difuso de forma bastante intuitiva, o tipo TAKAGI-SUGENO é computacionalmente mais eficiente, permitindo melhor adaptatividade que o tipo MANDAMI, segundo os autores. O modelo de um sistema de inferência difusa do tipo TAKAGI-SUGENO para duas entradas é ilustrado na figura 2.4.3.1.1 a seguir:

Nesta figura, a regra de inferência difusa será expressa como a expressão a seguir:

SE "entrada 1" = x E "entrada 2" = y, ENTÃO saída z = ax+by+c (eq. 2.4.3.1.1)

Nesta expressão ***x*** e ***y*** representam possíveis valores para entradas e ***z*** a resposta linear as possíveis entradas.

Um modelo de ordem "zero" tem como parâmetros:

$$a = b = 0 \qquad \text{(eq. 2.4.3.1.2)}$$

E conseqüentemente uma saída constante ***c***. Neste trabalho, cada constante ***c*** representará uma possível classificação de serviços, prospectada pela rede neural artificial.

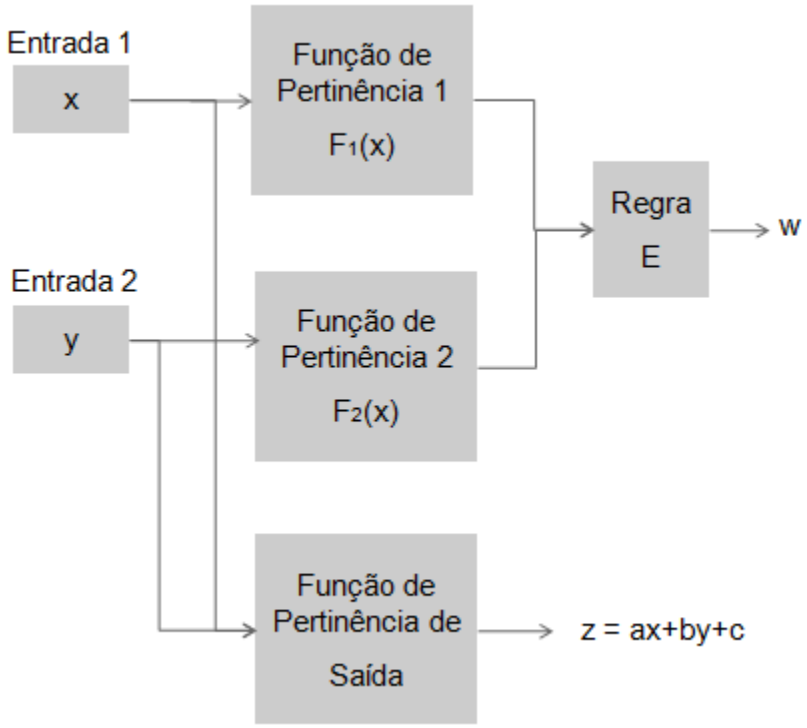

Fig. 2.4.3.1.1 Modelo de um sistema de inferência difusa do tipo TAKAGI-SUGENO (MATLAB,2010)

O nível de saída ***z*** de cada regra é ponderado através do parâmetro ***w*** de cada regra. Por exemplo, para uma regra do tipo ***E***, com entradas ***x*** e ***y***, o parâmetro ***w*** é calculado por:

$$w_i = regraE(F_1(x), F_2(x)) \qquad \text{(eq. 2.4.3.1.3)}$$

Na qual F_1 e F_2 são funções de pertinência das entradas 1 e 2, respectivamente.

A saída final é estimada pela média ponderada das saídas $\boldsymbol{z_i}$ de todas a regras, segundo eq. 2.4.3.1.3:

$$z = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \qquad \text{(eq. 2.4.3.1.4)}$$

Na qual N é o número de regras.

2.5 Comentários

O capítulo forneceu o embasamento teórico para a compreensão dos desafios a serem superados para o estabelecimento da arquitetura de referência para a Internet de Serviços, segundo os princípios da arquitetura orientada a serviços.

Adicionalmente, foram apresentadas as soluções de intermediação de serviços, encontradas na literatura, fundamentais para a construção de arquiteturas orientadas a serviços, segundo os vários paradigmas de computação conhecidos.

Estas soluções demandarão novos paradigmas para a seleção de serviços em ambientes heterogêneos e convergentes como a Internet do Futuro. Este novos paradigmas são representados por algoritmos de prospecção de dados, que constituem soluções usualmente empregadas para o aprendizado supervisionado e não supervisionado de padrões extraídos a partir de dados distribuídos em bases de dados heterogêneas.

A combinação destes algoritmos, base da solução proposta nesta tese, possibilitará a seleção adaptativa e dinâmica de serviços, visando atender os requisitos da arquitetura de referência da Internet de Serviços.

O próximo capítulo apresenta o problema de seleção de serviços em ambientes heterogêneos e convergentes, considerando a necessidade de avaliação de atributos de qualidade de serviços.

3. SELEÇÃO DE SERVIÇOS

Este capítulo tem por objetivo apresentar o problema de seleção de serviços móveis em ambientes heterogêneos e convergentes. Este capítulo é composto pelas seguintes seções:

- A seção 3.1 descreve o problema de seleção de serviços no contexto da Internet do Futuro;
- A seção 3.2 descreve o estado da arte das soluções de seleção de serviços;
- As seção 3.3 e 3.4 descrevem os parâmetros de QoS percebidos pela rede e pelo usuário, respectivamente, os quais serão utilizados para a seleção de serviços com base em atributos não funcionais;
- A seção 3.5 encerra o capítulo com comentários.

3.1 Apresentação do problema de seleção de serviços

Um fator chave de sucesso, para as arquiteturas orientadas a serviços (SOA), está relacionado com capacidade de descobrir os serviços disponíveis. Quando os serviços dispõem de descrições precisas, armazenadas em registros de serviços públicos, premissa para a descoberta nos sistemas SOA baseados na atual arquitetura de Internet, a tarefa de descoberta manual é provavelmente factível.

Um usuário ao realizar o processo de procura ou busca de serviços, com base em requisitos funcionais, descobre a existência de um conjunto de serviços que satisfaz estes requisitos, similares do ponto de vista funcional. Simultaneamente ou posteriormente ao processo procura, faz-se necessário a seleção do serviço ou conjunto de serviços que satisfaça um ou mais requisitos não funcionais. A definição e ordenação dos critérios de seleção, sejam eles perfeitamente definidos ou não, é parte integrante do problema.

No contexto da Internet de Serviços, na qual um elevado número de serviços estará disponível, a descoberta de serviços tornar-se-á uma atividade bastante complexa para realização apenas manual. A figura 3.1.1 ilustra os problemas relacionados à seleção de serviços, como parte do processo de descoberta de serviços.

O processo de seleção é usualmente subestimado ou simplesmente ignorado em várias soluções de descoberta de serviços (MIAN; et.al., 2009), uma vez que, pressupõem a existência de uma infra-estrutura fixa, na qual serviços são registrados em diretórios conhecidos a priori.

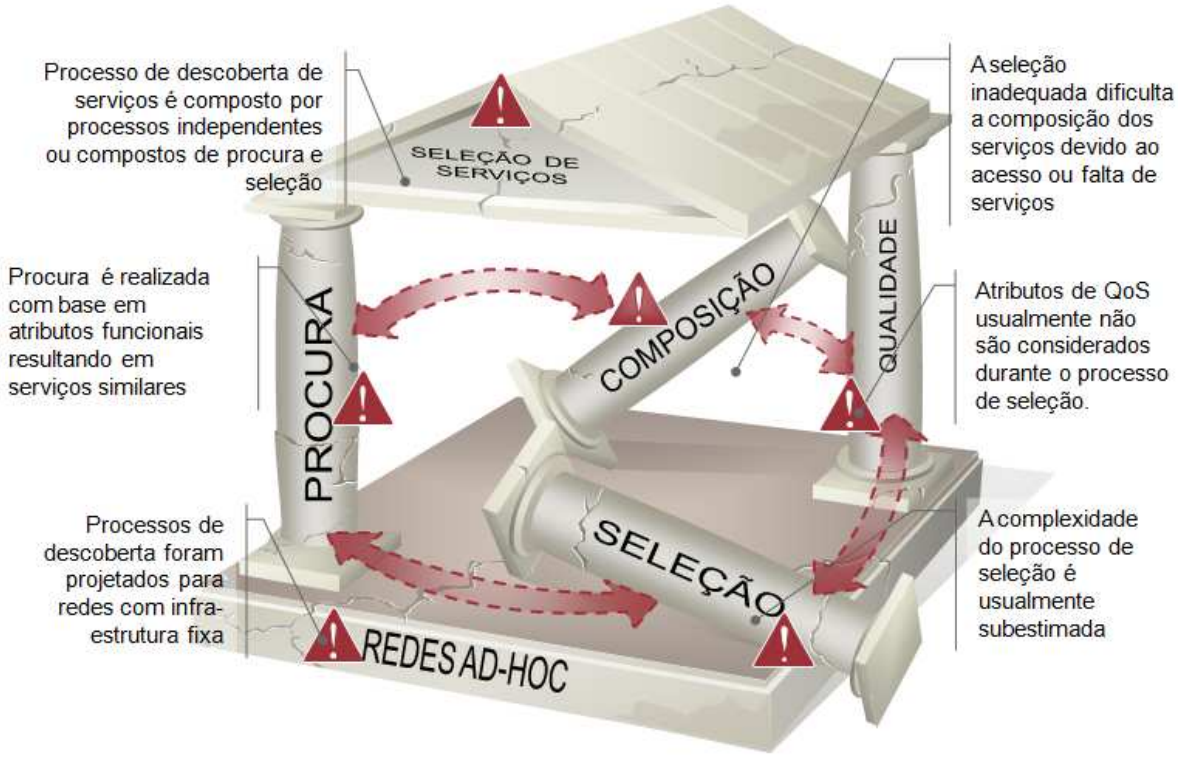

Fig. 3.1.1 Problemas para a seleção de serviços no contexto da Internet do Futuro (MIAN; et.al.,2009)

Segundo os mesmos autores, a realização inadequada do processo de seleção, causa perda de eficiência do processo de procura, com base em atributos funcionais, gerando um elevado número de serviços similares sob o ponto de vista funcional, porém distintos sob o ponto de vista não funcional.

A seleção inadequada de serviços dificulta a composição de mesmos, pois o excesso ou falta de serviços adequados, usualmente desconsiderando requisitos de qualidade dos mesmos, gera baixa percepção da qualidade do serviço, sob o ponto de vista do usuário final.

A solução destes problemas demandará um novo processo de descoberta dinâmico e adaptativo, que propicie procura e seleção eficiente e acurada em ambientes distribuídos, heterogêneos e convergentes. A etapa de seleção deverá atender aos requisitos ilustrados na figura 3.1.2, apresentados a seguir.

A redução do universo de escolha de serviços, agrupando a oferta de múltiplos serviços funcionalmente similares, com base em atributos não funcionais de Qualidade de Serviço, deverá permitir o refinamento da etapa de procura, integrando as etapas de seleção e composição, melhorando o processo de descoberta de serviços.

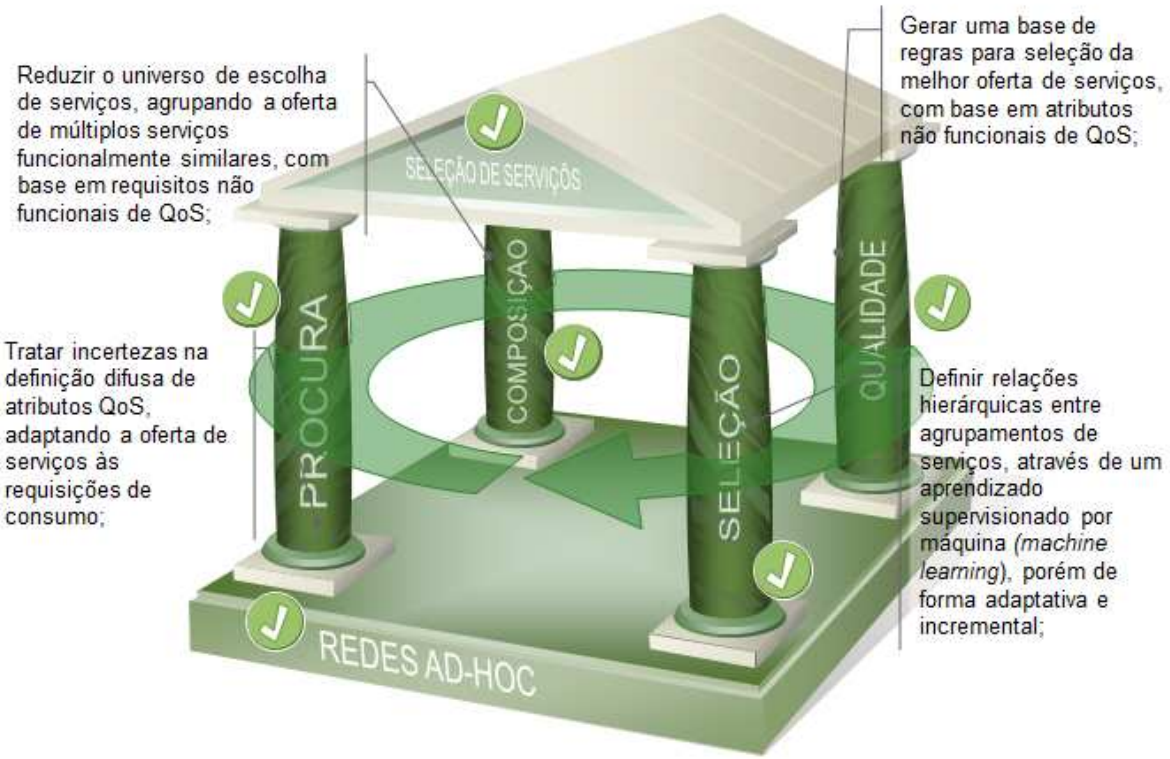

Fig.3.1.2 Requisitos necessários a etapa de seleção de serviços (NEXOF-RA-1,2010)

Durante a etapa de procura, segundo (NEXOF-RA-1,2010) o tratamento de incertezas, na definição difusa dos atributos de QoS, deverá possibilitar a adaptação das ofertas de serviços às requisições de consumo dos mesmos, gerando uma base de regras que apoiarão a decisão de seleção da melhor oferta de serviços, com base em atributos não funcionais.

A oferta de múltiplos serviços deverá ser agrupada, definindo relações hierárquicas de classificação destes agrupamentos, através de um aprendizado supervisionado e não supervisionado, porém de forma adaptativa e incremental, à medida que novos serviços são disponibilizados por redes móveis Ad-Hoc, como ocorre na Internet de Serviços.

3.2 Estado da arte para a seleção de serviços

O problema de seleção de serviços, baseado em múltiplos requisitos não funcionais de QoS, foi classificado como um problema "*NP-hard*" (ZENG; et al.,2004), e usualmente utilizam seleção manual ou algoritmos baseados em soluções heurísticas, como descrito a seguir:

- ZENG et al.(2004) propuseram três algoritmos com base em soluções heurísticas, nas quais a solução que utilizava a técnica "*integer programming*" apresentou os melhores resultados entre as três soluções avaliadas, entretanto é computacionalmente complexa para implementação em redes móveis Ad-Hoc;
- CANFORA (2005) propos uma solução heurística baseada em algoritmos genéticos e compararam com a solução baseada em "*integer programming*", sem resolver a questão da alta complexidade computacional de implementação em redes móveis Ad-Hoc;
- GU et al.(2006) propuseram uma solução heurística baseada no algoritmo de otimização de seccionamento de conjuntos, porém considerando uma infra-estrutura de rede fixa, inviabilizando sua utilização em redes móveis Ad-Hoc;
- YU et al. (2007) propuseram um algoritmo de seleção, baseado na disponibilidade dos nós da rede, porém necessitando uma visão global da rede, premissa não satisfeita em redes móveis Ad-Hoc.

Entre os doze trabalhos avaliados por MIAN et al. (2009), para descoberta de serviços em redes Ad-Hoc, apenas quatro mencionam algum tipo de solução de seleção:

- ZHU et al. (2003) mencionam que a seleção de serviço seja realizada pelo dispositivo cliente, sem apresentar maiores detalhes da solução. Não foram consideradas as restrições do dispositivo cliente, bem como requisitos não funcionais;
- REID et al. (2005) propuseram a seleção de serviços mais próximos, ou seja, considerando a disponibilidade dos serviços dos nós de rede mais próximos. Não foi considerada a disponibilidade de múltiplos serviços similares do ponto de vista funcional;
- TYLAN et al. (2004) propuseram que a seleção seja realizada por agentes móveis baseada em informação de contexto, sem considerar requisitos não funcionais;
- LENDERS et.al. (2005) propuseram que a seleção seja realizada baseada na distância dos nós de rede e na capacidade do serviço, sem considerar outros requisitos de QoS.

Recentemente, YANG et al. (2009) propuseram uma avaliação de requisitos não funcionais de QoS que combina requisitos dos serviços percebidos pelos usuários (USLA), tais como: disponibilidade, atraso de execução, preço e confiabilidade; com atributos

percebidos pelos nós que compõem a rede sob demanda (Ad-Hoc), tais como disponibilidade do nó, atraso e confiabilidade da rede. Os requisitos não funcionais percebidos pelos usuários serão utilizados neste trabalho, porém os mesmos estão relacionados com os requisitos não funcionais percebidos pelos nós da rede.

3.3 Requisitos não funcionais de QoS percebidos pela rede

Esta seção descreve os requisitos não funcionais percebidos pela rede móvel Ad-Hoc, composta por nós de dispositivos móveis, cujos atributos de qualidade foram utilizados pela solução de seleção de serviços proposta nesta tese.

3.3.1 Disponibilidade de nó de rede

Em redes sem fio, a disponibilidade do nó de rede em relação a seus vizinhos está diretamente relacionada à mobilidade dos nós que compõem a rede, assumindo que não há restrição no consumo de energia nos dispositivos móveis em cada nó da rede.

Se um nó ***i*** moveu-se para fora da área de transmissão de um nó ***j***, então ***i*** não é acessível por ***j*** e, como conseqüência, ***i*** torna-se não disponível para o nó ***j*** também.

A disponibilidade de um nó expressa, de certa forma, a disponibilidade da rede pois se este nó estiver indisponível, a rota que conecta o requisitante ao provedor da qual o nó faz parte, também estará indisponível. Dessa forma, a mobilidade do nó é utilizada para calcular a disponibilidade do nó, conforme ilustrado na figura 3.3.1.1.

Na figura 3.1.11, dois nós de rede (*Mobile Host*) $\boldsymbol{MH_i}$ e $\boldsymbol{MH_j}$ possuem o mesmo espaço de transmissão ***R***. Cada nó move-se aleatoriamente e assume-se que o espaço de movimentação é delimitado por um círculo de raio ***r*** e ***d*** representa a distância entre $\boldsymbol{MH_i}$ e $\boldsymbol{MH_j}$. Estes três parâmetros serão utilizados para o cálculo da disponibilidade de nó da rede.

Assume-se, por simplicidade, que:

- O raio ***R*** de transmissão é conhecido a priori, sendo uma constante ou variando conforme uma função conhecida.
- A posição de ambos os nós é conhecida, por exemplo, através de um sistema de coordenadas geográficas referenciadas (*Global Positioning System, GPS*).

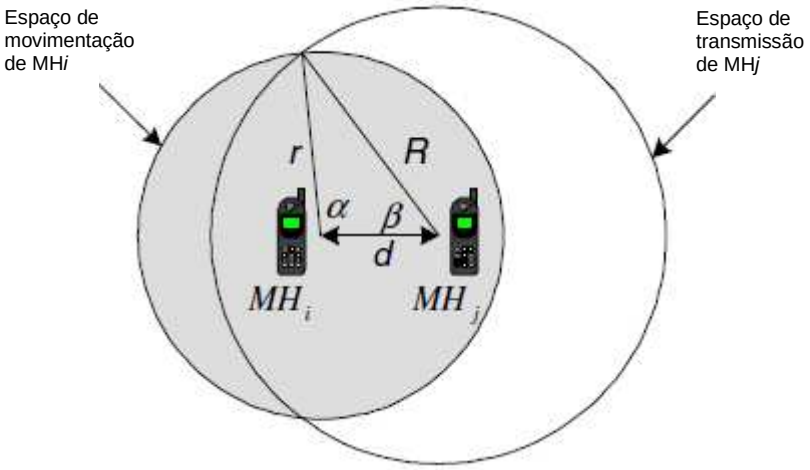

Fig. 3.3.1.1 Parâmetros para cálculo da disponibilidade de nó da rede

Considerando que (x_i,y_i) e (x_j,y_j) são as coordenadas de MH_i e MH_j, respectivamente, então a distância ***d*** pode calculado usando o cálculo de distância Euclidiana.

O raio ***r***, que define o espaço de movimentação de um dispositivo móvel, será calculado pelo produto entre velocidade ***s*** e o tempo médio de serviço ***t***:

$$r = \mathrm{s} \times \mathrm{t} \qquad \text{(eq. 3.3.1.1)}$$

Na qual:

- O tempo médio de serviço ***t*** pode calculado como o valor médio dos últimos ***n*** acessos que o serviço:

$$t = \sum_{i=1}^{n} (t_i/n) \qquad \text{(eq. 3.3.1.2)}$$

- A velocidade média ***s*** pode calculada com base na distância ***d*** percorrida entre os intervalos de tempo t_1 e t_2:

$$s = \frac{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}}{t_2-t_1} \qquad \text{(eq. 3.3.1.3)}$$

Conhecendo os valores dos parâmetros ***R***, ***r***, e ***d***, a probabilidade de MH_i permanecer dentro do alcance de transmissão de MH_j, pode ser calculada por :

$$P_i^{IN} = \frac{A_i^{IN}}{A_i^T} \qquad \text{(eq. 3.3.1.4)}$$

A probabilidade P_i^{IN} representa o espaço de $\boldsymbol{MH_i}$ movendo-se dentro da área de transmissão de $\boldsymbol{MH_j}$, denotada de A_i^{IN} , dividido pela área total de movimentação de $\boldsymbol{MH_i}$, denotada de A_i^T .

Do triângulo ilustrado na figura 3.3.1.1, temos:

$$\alpha = \arccos\left(\frac{r^2+d^2-R^2}{2r \times d}\right) \qquad \text{(eq.3.3.1.5)}$$

$$\beta = \arccos\left(\frac{R^2+d^2-r^2}{2R \times d}\right) \qquad \text{(eq. 3.3.1.6)}$$

Combinando ambas as equações, temos:

$$A_i^{IN} = \beta R^2 + \alpha r^2 - (R^2 sen(\beta)\cos(\beta) + r^2 sen(\alpha)\cos(\alpha)) \qquad \text{(eq. 3.3.1.7)}$$

Calculando-se a área do círculo ilustrado na figura 3.3.1.1, temos:

$$A_i^T = \pi r^2 = \pi \times (s \times t)^2 \qquad \text{(eq.3.3.1.8)}$$

Desta forma:

$$P_i^{IN} = \frac{A_i^{IN}}{\pi s^2 t^2} \qquad \text{(eq.3.3.1.9)}$$

A probabilidade de $\boldsymbol{MH_i}$ estar dentro da área de transmissão de $\boldsymbol{MH_j}$, pode ser calculada supondo que o serviço $\boldsymbol{s}$, provido por $\boldsymbol{MH_i}$ é um candidato para suprir uma requisição de $\boldsymbol{MH_j}$, e a disponibilidade deste nó é denotada de $q_{av}^n(s)$, conforme eq. 3.3.1.10:

$$q_{av}^n(s) = P_i^{IN} = \frac{A_i^{IN}}{\pi s^2 t^2} = \frac{\beta R^2 + \alpha r^2 - (R^2 sen(\beta)\cos(\beta) + r^2 sen(\alpha)\cos(\alpha))}{\pi s^2 t^2} \qquad \text{(eq. 3.3.1.10)}$$

3.3.2 Atraso de rede

O atraso de um pacote de dados trafegando entre nós da rede é definido pelo intervalo de tempo necessário para o mesmo deixar o nó de origem e alcançar o nó de destino. Similar ao que ocorre em redes com fio, o atraso de uma rede sem fio é definido pela soma dos atrasos na rota que contêm vários nós, porém dependente da velocidade ***v*** de cada nó.

Segundo GEIHS et al. (2004), o atraso médio de um pacote de dados para uma rede com ***n*** nós é estimado pela eq. 3.3.2.1:

$$D(n) = O(\sqrt{n}/v(n) \qquad \text{(eq. 3.3.2.1)}$$

Na qual:

- ***v(n)*** descreve o comportamento da velocidade dos nós em função de ***n***.

Este trabalho utiliza-se do resultado final desta equação, na qual os detalhes adicionais podem ser encontrados em GEIHS et al. (2004).

3.3.3 Confiabilidade da rede

Em redes sem fio, a confiabilidade da rede está diretamente relacionada à qualidade dos canais de comunicação sem fio (YANG; et al.,2009) e a confiabilidade de cada nó que compõem a rede. Com base nesta premissa, a disponibilidade de rede de um serviço ***s*** será calculada pela taxa de perda de pacotes de comunicação, considerando que o receptor tem uma capacidade limitada para receber pacotes, acumulados em sua fila.

Seja $\boldsymbol{P_d}$ a probabilidade de perda de um pacote e $\boldsymbol{P_e}$ a probabilidade de erro do canal de comunicação sem fio. Um pacote é corretamente recebido nas seguintes condições:

- Condição 1: O pacote de dados é corretamente recebido através de canal de comunicação sem fio (com probabilidade $\boldsymbol{1- P_e}$);
- Condição 2: O pacote de dados não é perdido na fila de recepção (com probabilidade $\boldsymbol{1- P_d}$).

Então a taxa de perda de pacotes $\boldsymbol{\eta}$ é estimada pela eq. 3.3.3.2:

$$\eta = 1 - (1 - P_e)(1 - P_d) \qquad \text{(eq. 3.3.3.2)}$$

Utilizando esta equação, quanto mais longe um nó provedor de serviço está, a confiabilidade de rede do serviço s é estimada pela eq. 3.3.3.3:

$$q_{re}^{n}(s) = \eta = 1 - (1 - P_e)(1 - P_d) \qquad \text{(eq. 3.3.3.3)}$$

Seja $\boldsymbol{\eta^i}$ a taxa de perda de pacote de um nó especifico ***i***. Considerando também os nós intermediários, desde um nó consumidor (receptor) ***r*** até um nó provedor ***p***, temos a eq. 3.3.3.4:

$$q_{re}^{n}(s) = 1 - \prod_{i=r}^{p}\left(1 - \eta^i\right) = 1 - \prod_{i=r}^{p}(1 - P_e^i)(1 - P_d^i) \qquad \text{(eq. 3.3.3.4)}$$

Na qual $\boldsymbol{P^i_e}$ e $\boldsymbol{P^i_d}$ referem-se à probabilidade de erro do canal de comunicação do nó ***i*** e a probabilidade de não recebimento de pacotes na fila do nó ***i***, respectivamente.

A estimativa das probabilidades $\boldsymbol{P_e}$ e $\boldsymbol{P_d}$, dependem de características do canal de comunicação, do meio de acesso e da forma de cálculo, como por exemplo, em (YANG; et al.,2009) foi utilizado método baseado em cadeias de *Markov*.

O método de seleção de serviços, adotado neste trabalho, independe do método de estimativa destas probabilidades e por este motivo, o método de cálculo não será detalhado.

3.4 Requistos não funcionais de QoS percebidos pelo usuário

A seguir serão descritos os requisitos não funcionais de QoS percebidos pelo usuário, cujos atributos foram utilizados para a solução de seleção de serviços proposta neste trabalho.

3.4.1 Disponibilidade do Serviço

A disponibilidade $\boldsymbol{q_{av}^{u}(s)}$ de um serviço, denotado como ***s***, é avaliada como a probabilidade do serviço ser utilizável no momento em que foi requisitado. Para calcular esta probabilidade, serão considerados dois eventos:

- Evento com probabilidade $\boldsymbol{q_{av}^{n}(s)}$: o serviço $\boldsymbol{s}$ pode estar funcionando bem em seu provedor, denominado como ***h(s)***, porém o provedor encontrar-se inacessível no momento em que o serviço foi requisitado;
- Evento com probabilidade $\boldsymbol{q_{av}^{s}(s)}$: o serviço pode não estar acessível ou não estar funcionando.

Com base em ambos os eventos, a probabilidade de um serviço estar disponível ao usuário é estimada pela eq. 3.3.4.1:

$$q_{av}^{u} = 1 - (1 - q_{av}^{n}) \times (1 - q_{av}^{s}) \qquad \text{(eq. 3.4.1.1)}$$

Na qual a probabilidade $\boldsymbol{q_{av}^{n}(s)}$ será estimada conforme eq. 3.3.1.10 e $\boldsymbol{q_{av}^{s}(s)}$ será estimada, segundo (ZENG, et al.,2004), como:

$$q_{av}^{s}(s) = \frac{T_a(s)}{\delta} \qquad \text{(eq. 3.4.1.2)}$$

Na qual $\boldsymbol{T_a(s)}$ é o intervalo de tempo total no qual o serviço estava disponível e $\boldsymbol{\delta}$ é o tempo transcorrido. A estimativa desta probabilidade é realizada pelo provedor e enviada ao consumidor sob demanda do mesmo. $\boldsymbol{\delta}$ é uma constante definida pelo provedor de serviço $\boldsymbol{s}$.

3.4.2 Preço do serviço

O preço $\boldsymbol{q_{pr}^{u}(s)}$ de um serviço $\boldsymbol{s}$ é o valor que um consumidor deve pagar para utilizar o serviço requisitado, sendo normalmente estipulado em acordo entre as partes, através do USLA. Caso não exista um acordo definido entre as partes, o preço será estipulado pelo provedor do serviço, sendo armazenado com um valor constante nos atributos do serviço.

3.4.3 Confiabilidade do serviço

A confiabilidade de um serviço, denotada de $\boldsymbol{q_{re}^{u}(s)}$ **,** percebida pelo usuário **u,** é estimada como a probabilidade de uma requisição de serviço seja atendida corretamente, isto é, o consumidor receba o serviço que atenda suas expectativas, dentro de um período máximo de tempo estipulado pelo USLA. Caso o USLA ainda não esteja definido, a interface pública

do serviço deverá indicar este valor de tempo máximo, em função da disponibilidade do serviço no provedor.

A estimativa de confiabilidade depende da configuração $\boldsymbol{c}$ de *hardware* e *software* $\boldsymbol{q_{re}^{c}(s)}$ do provedor de serviço $\boldsymbol{s}$ e da qualidade da rede $\boldsymbol{q_{re}^{n}(s)}$, entre provedor e consumidor, conforme eq. 3.4.3.1:

$$q_{re}^{u}(s) = q_{re}^{c}(s) + q_{re}^{n}(s) \qquad \text{(eq. 3.4.3.1)}$$

A estimativa de $\boldsymbol{q_{re}^{c}(s)}$, segundo ZENG et al. (2004), normalmente é realizada com base em histórico de execução do serviço conforme eq. 3.4.3.2:

$$q_{re}^{c}(s) = N_f/T(s) \qquad \text{(eq. 3.4.3.2)}$$

Na qual $\boldsymbol{N_f(s)}$ é o número de vezes no qual o serviço $\boldsymbol{s}$ foi executado com sucesso dentro do período de tempo esperado e $\boldsymbol{T(s)}$ é o número total de requisições do serviço $\boldsymbol{s}$.

Combinando as eq. 3.4.3.2 e 3.3.3.4, temos a eq. 3.4.3.3:

$$q_{re}^{u}(s) = q_{re}^{c}(s) + q_{re}^{n}(s) = \frac{N_f}{T(s)} + 1 - \prod_{i=r}^{p}\left(1 - P_e^i\right) \times \left(1 - P_d^i\right) \qquad \text{(eq. 3.4.3.3)}$$

3.4.4 Atraso na execução do serviço

O atraso $\boldsymbol{q_{de}^{u}(s)}$ na execução de um serviço $\boldsymbol{s}$, sob a perspectiva de um usuário $\boldsymbol{u}$, é a medida do intervalo de tempo entre a requisição de um serviço é emitida pelo usuário, até o mesmo receba o resultado da sua requisição. Os fatores que contribuem para o atraso na execução de um serviço são o tempo de processamento local $\boldsymbol{T_{local}(s)}$ e o tempo de transmissão da rede $\boldsymbol{T_{net}(s)}$, conforme eq. 3.4.4.1:

$$q_{de}^{u}(s) = T_{local}(s) + T_{net}(s) \qquad \text{(eq. 3.4.4.1)}$$

Considerando o aumento do poder computacional dos dispositivos móveis consumidores de serviços, o $\boldsymbol{T_{local}(s)}$ tende a zero, e o atraso na execução do serviço pode ser estimado pelo tempo de transmissão da rede.

3.4.5 Definição do vetor de qualidade

Combinando os atributos de qualidade de serviço, sob o ponto de vista do usuário, temos um vetor de qualidade ***QV(s)*** , conforme eq. 3.4.5.1:

$$QV(s) = (q_{av}(s), q_{pr}(s), q_{re}(s), q_{de}(s)) \qquad \text{(eq. 3.4.5.1)}$$

Considerando que um serviço é composto por um conjunto de tarefas $\{t_i | i = 1, ..., n\}$, e uma tarefa t_i ,necessária para satisfazer a requisição do consumidor do serviço, necessita de seleção, uma vez que, foram encontrados ***m*** serviços candidatos $S(t_i) = \{ s_{i,1}, s_{i,2}, ..., s_{i,m} \}$, que potencialmente executam a tarefa t_i.

Cada serviço candidato:

$s_{i,j} | \, j = 1, ... ,m$

Possui pelo menos quatro atributos de QoS, conforme vetor de qualidade definido pela eq. 3.4.5.1, cuja expansão vertical para cada valor de ***j*** é representada pela matriz ***M***, segundo eq. 3.4.5.2:

$$M = \begin{pmatrix} q_{av}(s_1), q_{av}(s_2), \dots, q_{av}(s_m) \\ q_{pr}(s_1), q_{pr}(s_2), \dots, q_{pr}(s_m) \\ q_{re}(s_1), q_{re}(s_2), \dots, q_{re}(s_m) \\ q_{de}(s_1), q_{de}(s_2), \dots, q_{de}(s_m) \end{pmatrix} \qquad \text{(eq. 3.4.5.2)}$$

Observa-se que o índice ***i*** foi omitido, pois o objetivo é selecionar o grupo de serviços que satisfaçam a uma única tarefa, sem perda de generalidade, uma vez que, a matriz ***M***, pode ser expressa por:

$$M = (q_{i,j}(s_i); 1 \leq i \leq m, 1 \leq j \leq 4) \qquad \text{(eq. 3.4.5.3)}$$

Em ***M***, cada coluna q_i corresponde a um serviço componente s_i, na qual cada linha q_j corresponde a um vetor de qualidade.

O vetor QoS de um de serviço composto, formado pela composição de um ou mais serviços componentes (s_i), será estimado pela eq. 3.4.5.4 :

$$QV(s^C) = (q'_1(s^C), \dots, q'_r(s^C)$$ (eq. 3.4.5.4)

Na qual:

- $\boldsymbol{s^C = \{s_i | \ i = 1,\dots,n\}}$**; n >1** e $\boldsymbol{s_i}$ é um serviço elementar ou componente;
- $\boldsymbol{q_i'(s^C)}$ representa a estimativa do valor de QoS de um serviço composto, agregando o valor de qualidade de seus componentes.

Considerando a medição de qualidades dos quatro parâmetros de qualidades citados na seção 3.3, temos a eq. 3.4.5.5:

$$QV(s^C) = (q_{av}(s^C), q_{pr}(s^C), q_{re}(s^C), q_{de}(s^C))$$ (eq. 3.4.5.5)

Na qual os atributos de qualidade serão estimados por:

- Disponibilidade de um serviço composto $\boldsymbol{s^C}$ será estimada pelo produto dos valores de disponibilidades dos serviços componentes $\boldsymbol{s_i}$, segundo eq. 3.4.5.6:

$$q_{av}(s^C) = \prod_{\boldsymbol{i-1}}^{\boldsymbol{n}} q_{av}(s_i)$$ (eq. 3.4.5.6)

- Preço de execução de um serviço composto $\boldsymbol{s^C}$ será estimada pela soma dos valores de preço dos serviços componentes ($\boldsymbol{s_i}$), segundo eq. 3.4.5.7:

$$q_{pr}(s^C) = \sum_{\boldsymbol{i-1}}^{\boldsymbol{n}} q_{pr}(s_i)$$ (eq.3.4.5.7)

- Confiabilidade de um serviço composto $\boldsymbol{s^C}$ será estimada pelo produto dos valores de confiabilidade dos serviços componentes ($\boldsymbol{s_i}$), segundo eq. 3.4.5.8:

$$q_{re}(s^C) = \prod_{\boldsymbol{i-1}}^{\boldsymbol{n}} q_{re}(s_i)$$ (eq. 3.4.5.8)

- Atraso na execução de um serviço composto $\boldsymbol{s^C}$ será estimada pela soma dos valores de atraso dos serviços componentes ($\boldsymbol{s_i}$), segundo eq. 3.4.5.9:

$$q_{de}(s^C) = \sum_{\boldsymbol{i-1}}^{\boldsymbol{n}} q_{de}(s_i)$$ (eq.3.4.5.9)

Similar a (ALRIFAI;et.al.,2009), as equações 3.4.5.6 e 3.4.5.8 podem ser transformadas em soma dos valores individuais, através a aplicação da operação de logaritmo. Dessa forma, a equação 3.4.5.10 permite a estimativa agregada dos valores individuais de cada serviço ***j*** para qualquer parâmetro ***k*** de QoS:

$$q_k^{'}(s^C) = \sum_{j=1}^{n} q_k(s_j) \qquad \text{(eq.3.4.5.10)}$$

3.5 Comentários

Este capítulo apresentou o problema de seleção de serviços e o estado da arte das poucas soluções encontradas na literatura, corroborando com a tese que o problema de seleção é usualmente subestimado ou simplesmente ignorado (MIAN,2009).

Adicionalmente, nenhuma destas soluções avaliadas satisfaz aos requisitos não funcionais necessários à implementação da arquitetura de referência proposta para a Internet de Serviços (AGASSI,et al.,2010).

Estes requisitos demandam a avaliação de atributos de qualidade de serviços dos serviços em adição aos requisitos funcionais, como proposto em YANG et al. (2009), que combina requisitos não funcionais dos serviços percebidos pelos usuários (USLA), tais como: disponibilidade, atraso de execução, preço e confiabilidade; com atributos percebidos pelos nós que compõem a rede Ad-Hoc, tais como disponibilidade do nó, atraso e confiabilidade da rede.

Ambos tipos de requisitos não funcionais foram detalhados, pois a avaliação quantitativa dos mesmos será utilizada para a construção da proposta de solução apresentada no próximo capítulo.

4. A SOLUÇÃO PROPOSTA

Este capítulo tem por objetivo descrever os requisitos e componentes da solução de seleção adaptativa de serviços em ambientes heterogêneos e convergentes, encapsulados como serviços complementares à arquitetura de *software* de referência da IoS (AGASSI,et al.,2010).

Esta arquitetura de referência, proposta gerada a partir de inúmeros trabalhos de pesquisa do grupo de pesquisa para a Internet do Futuro - *NESSI Open Framework – Reference Architecture* (NEXOF-RA-1, 2010), é descrita como um sistema distribuído, composto de subsistemas e funcionalidades.

O capítulo é composto pelas seguintes seções:

- A seção 4.1 descreve os requisitos funcionais e não funcionais, bem como os componentes que compõem a arquitetura de referência da Internet de Serviços (IoS);
- A seção 4.2 descreve o subsistema para descoberta de serviços, no contexto da IoS;
- A seção 4.3 descreve os algoritmos para seleção adaptativa de serviços propostos neste trabalho;
- A seção 4.4 descreve a interação entre componentes do subsistema descoberta de serviços;
- A seção 4.5 descreve o processo de seleção adaptativo e interativo de serviços proposto;
- A seção 4.6 encerra o capítulo com comentários.

4.1 Arquitetura de referência da IoS

Segundo a arquitetura de referência da Internet de Serviços (NEXOF-RA-3,2010), os requisitos funcionais e não funcionais, para a descoberta de serviços, serão descritos a seguir.

4.1.1 Requisitos Funcionais

Os requisitos funcionais para a descoberta de serviços são definidos por um conjunto de funcionalidades, encapsuladas como serviços segundo (NEXOF-RA-3; 2010):

- **Serviço de Descoberta (**`discoverServices()`): esta funcionalidade possibilita a procura de um conjunto de serviços para satisfazer um conjunto de requisitos funcionais;
- **Seleção de Serviços (**`matchServices()`**):** esta funcionalidade compara os requisitos solicitados com as capacidades dos serviços disponíveis, para selecionar aqueles serviços que melhor atendem aos requisitos;
- **Ordenação de Serviços (**`rankServices()`**):** a ordenação de serviços consiste em estabelecer uma pontuação para os serviços selecionados;
- **Gestão de Algoritmos (**`manageAlgorithms()`**):** esta funcionalidade permite a gestão de algoritmos, permitindo o acréscimo (`addAlgoritm()`), remoção (`deleteAlgorithm()`) e atualização (`updateAlgorithm()`) de algoritmos.

Estes serviços e os atores que interagem com os mesmos, são ilustrados na figura 4.1.1.1:

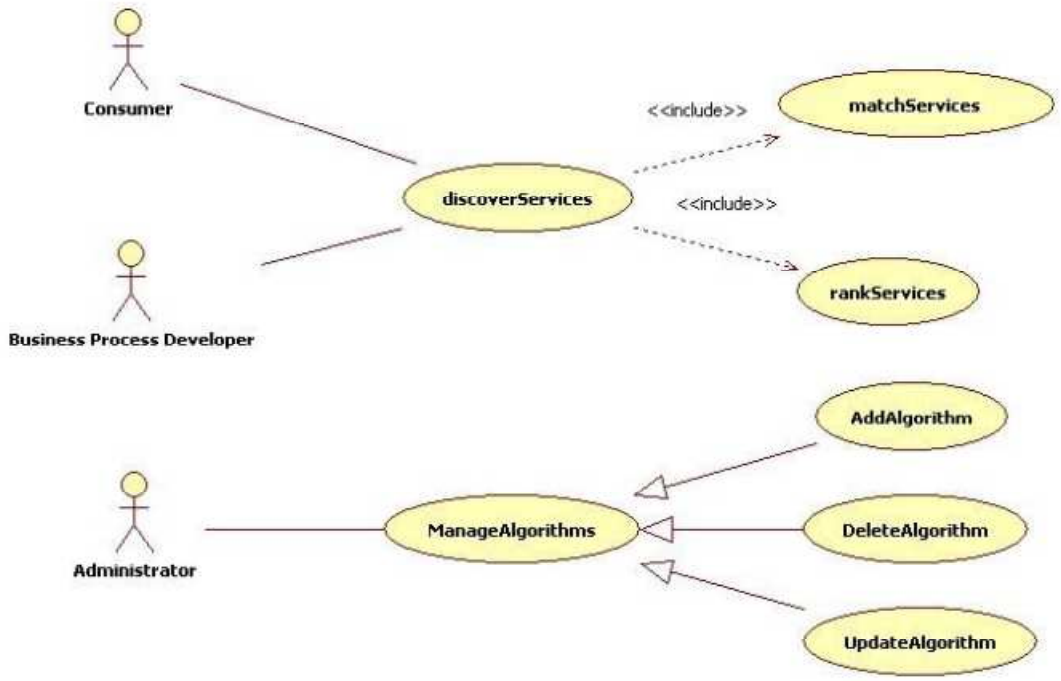

Fig. 4.1.1.1 Requisitos funcionais para descoberta de serviços (NEXOF-RA-3, 2010)

Na figura 4.1.1.1, a arquitetura proposta para a IoS (NEXOF-RA-3,2010) segundo a arquitetura SOA, pressupõe a existência de um ator, denominado de Desenvolvedor de Processos de Negócios (*Business Process Developer*), que solicitará a funcionalidade de

Descoberta de Serviços (`discoverServices()`), buscando o atendimento das necessidades definidas pelo ator Consumidor (*Consumer*), incluindo as funções `matchServices()` e `rankServices()`, descritas neste capítulo.

Para o gerenciamento dos algoritmos (`manageAlgorithms()`) de seleção adaptativa e interativa de serviços, propostos neste trabalho, foi nominado o ator Administrador (*Administrator*).

4.1.2 Requisitos não funcionais (atributos de qualidade)

Segundo o modelo de qualidade do (NEXOF-RA-3,2010), os atributos de qualidades utilizados para da solução de descoberta de serviços para a IoS, são definidos através de capacidades, conforme tabela 4.1.2.1.

Tabela 4.1.2.1 Requisitos não funcionais e resultados esperados do processo de descoberta de serviços (NEXOF-RA-3,2010)

Requisitos não funcionais	**Critérios de avaliação**	**Resultados esperados**
Capacidade de integração	Critério subjetivo, tratando-se de um protótipo	Alta
Capacidade de interoperação	Critério subjetivo, tratando-se de um protótipo	Alta
Acuracidade	Número de casos classificados com sucesso em relação ao total de casos avaliados	Alta
Desempenho	Tempo de resposta desde a requisição até a proposta de um ou mais serviços	Baixo
Escalabilidade	Relação entre a capacidade computacional do sistema e o tamanho do sistema para atender uma maior demanda futura	Alta
Segurança	Não será avaliado na PoC	Alta

Nesta tabela, os requisitos disponibilidade, interoperabilidade, capacidade de recuperação, confiabilidade e segurança, não são afetados pelas decisões necessárias ao

projeto e execução do processo de descoberta de serviços, uma vez, que as mesmas não afetam as funcionalidades e serviços deste processo.

Adaptabilidade, isto é a capacidade de modificação e portabilidade, também não é afetado pelo processo de descoberta de serviços, uma vez que a arquitetura orientada a serviços possibilita a capacidade de modificação e portabilidade necessárias para adaptação das interfaces dos serviços de descoberta e implementação das funcionalidades destes serviços em qualquer linguagem.

Capacidade de construção, integração e manutenção também não são afetadas pelas decisões tomadas para a definição do processo de descoberta, pois a arquitetura orientada a serviços possibilita a construção, integração e manutenção dos serviços de descoberta com um acoplamento fraco e reusabilidade inerentes à definição de serviços.

Acuracidade, desempenho e escalabilidade são requisitos de qualidade, para os quais espera-se uma melhora quando comparado com processos manuais de descoberta.

4.1.3 Componentes da arquitetura de referência da IoS

Esta seção tem por objetivo prover uma descrição de alto nível da arquitetura de referência da IoS, como sistema distribuído, através da decomposição em subsistemas e respectivos componentes, conforme ilustrado na figura 4.1.3.1, segundo (AGASSI,et al.,2010).

O principal subsistema é denominado de plataforma de serviços (*Service Plataform*), composto por subsistemas:

- **Interface de Usuário**, cujo objetivo é permitir a interação do usuário final com os demais subsistemas, adaptando suas necessidades e preferências aos serviços disponíveis na IoS;
- **Publicação e Descoberta**, contendo as funcionalidades de divulgação, procura e seleção de serviços, descritas na próxima seção;
- **Entrega e Consumo de Serviços**, contendo as funcionalidades para provisionamento e consumo de serviços;
- **Infra-estrutura**, contendo as funcionalidades para criação, gerenciamento e monitoramento de recursos computacionais.

Adicionalmente à Plataforma de Serviços, a arquitetura de referência da IoS é composta dos subsistemas:

- **Segurança**, responsável pela proteção de dados através da definição, criação e gerenciamento de controles de acesso;
- **Gerenciamento**, responsável pela negociação e monitoramento do USLA.

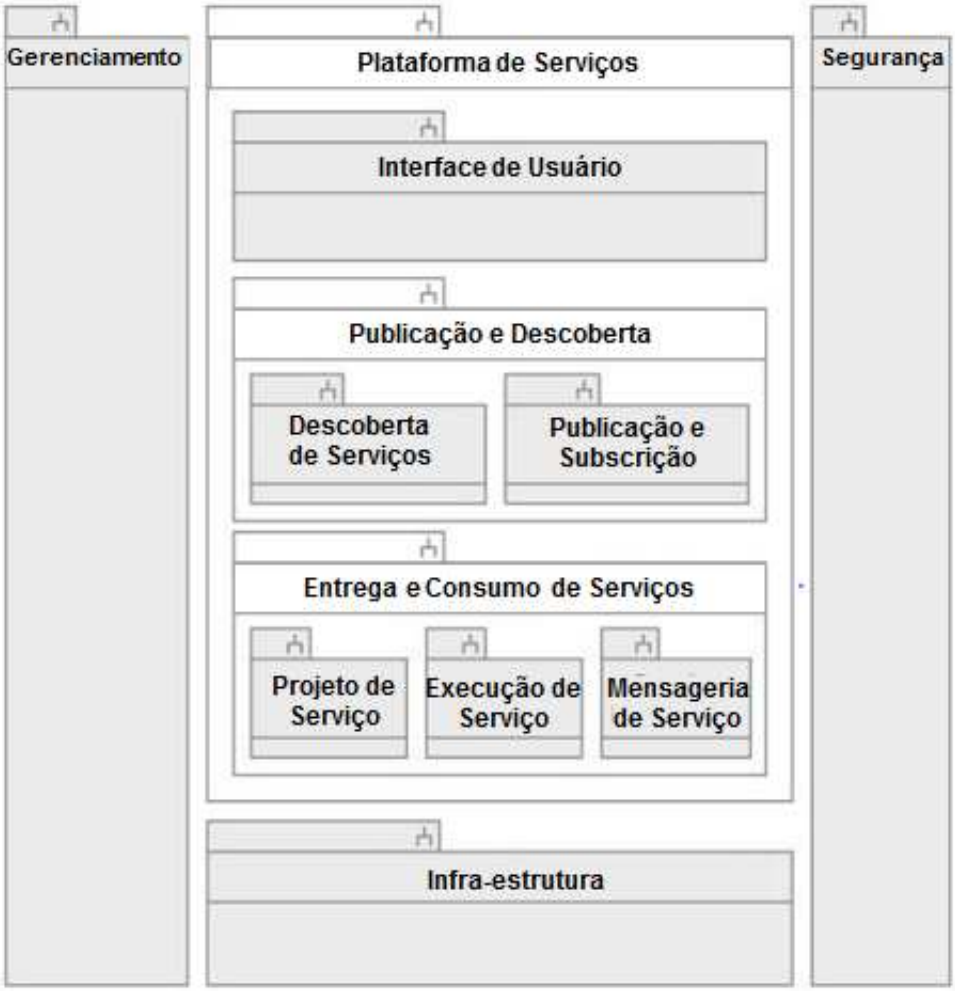

Fig. 4.1.3.1 Subsistemas que compõem a arquitetura da IoS (AGASSI,et al.,2010)

4.2 Subsistema *Descoberta de Serviços*

O subsistema *Descoberta de Serviços (ServiceDiscovery)*, dentro do contexto da IoS, é formado pelos componentes ilustrados na figura 4.2.1, segundo (NEXOF-RA-1,2010), apresentando as interfaces com os atores consumidores de serviços e desenvolvedores de processos de negócios, bem como interfaces com componentes externos a este subsistema.

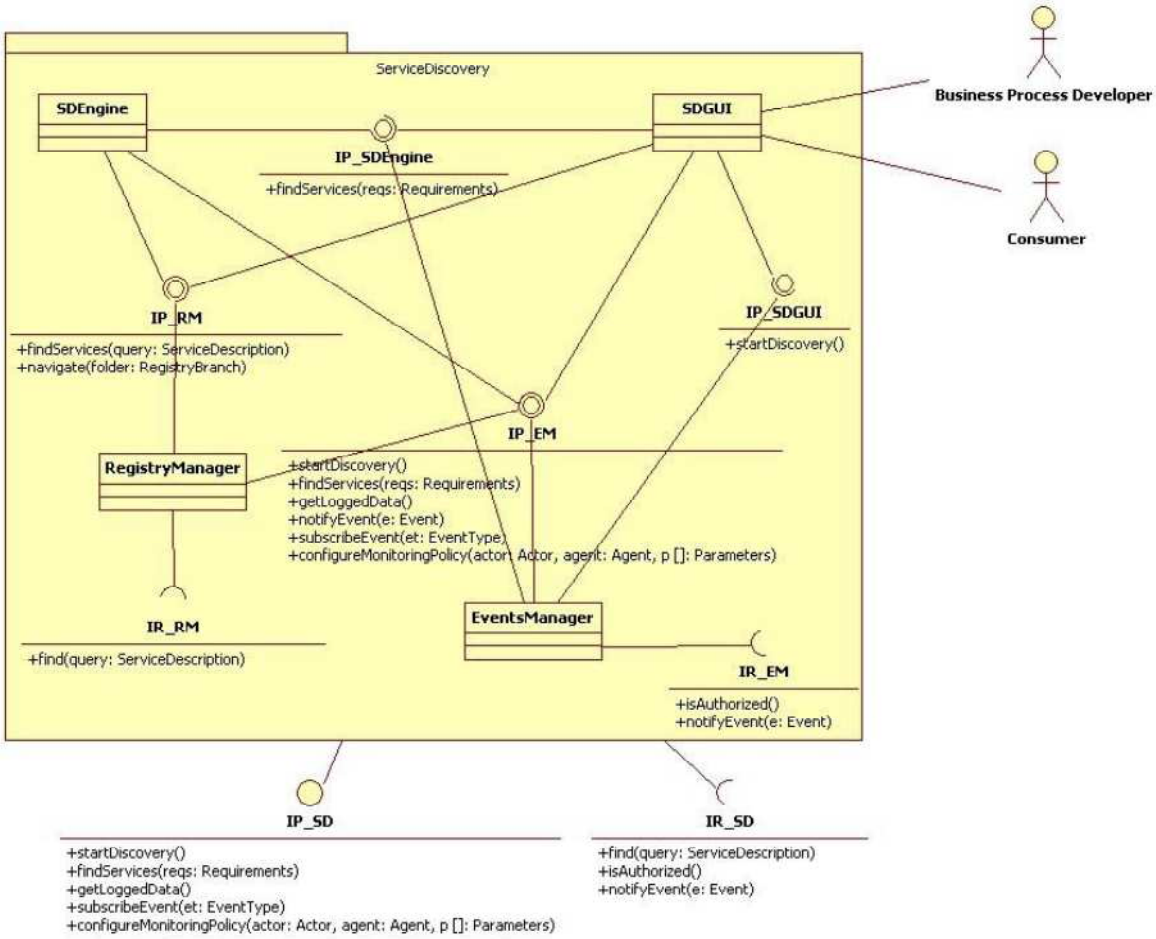

Fig. 4.2.1 Componentes do subsistema *Descoberta de Serviços*

4.2.1 Interface de Usuário para Descoberta de Serviços (SDGUI)

A interface de usuário permite que os atores acima realizem a descoberta interativa de serviços. Após a definição de requisitos, os usuários receberão uma lista ordenada de serviços que atendam a estes requisitos, através da funcionalidade:

- `startDiscovery()`

  Esta funcionalidade prepara os demais componentes deste pacote, para descoberta interativa de serviços, através do acesso a Interface de Usuário.

4.2.2 Gerenciador de Registro (RegistryManager)

Este componente é responsável por armazenar as definições dos serviços disponíveis e gerenciar conexões com os demais componentes do subsistema, através da provisão das seguintes funcionalidades:

- `findService(query: ServiceDescription) : AgentDescription`

  Procura serviços com base em descrições fornecidas através de agentes;

- `navigate(folder: RegistryBranch): BranchContent`

  Esta funcionalidade permite obter uma seção ou ramificação do registro de serviços, assumindo que o mesmo está estruturado em formato de árvore de decisão;

- `find(query: ServiceDescription)`

  Esta funcionalidade permite obter informação sobre descrição dos serviços armazenados no Registry Manager.

4.2.3 Gerenciador de Eventos

O Gerenciador de Eventos centraliza qualquer evento que ocorre no pacote *Descoberta de Serviços*, atuando como uma interface para notificar erros e eventos para componentes externos. Ele gerencia autorização e segurança de acesso dos usuários.

Funcionalidades providas:

- `startDiscovery()`

Esta funcionalidade prepara os demais componentes deste pacote, para descoberta interativa de serviços, através do acesso a Interface de Usuário;

- `findServices(reqs: Requirements): ServicesRanking`

Esta funcionalidade procura serviços, gerando uma lista ordenada segundo o grau de atendimento a requisitos;

- `getLoggedData()`

Esta funcionalidade retorna os dados acessados (*logged*) por um usuário;

- `notifyEvent(e: Event)`

Permite receber eventos internos do componente *Descoberta de Serviços*, de forma que as notificações de eventos podem ser reenviadas para os componentes subscritos a este serviço;

- `subscribeEvent(et: EventType)`

Permite que clientes de serviços subscrevam seus interesses a um tipo de evento. O cliente será notificado sempre que o evento de um determinado tipo ocorrer;

- `configureMonitoringPolicy(actor: Actor, agent: Agent, p[]: Parameters)`

Esta funcionalidade configura regras de monitoramento do componente. O parâmetro do agente permite especificar um componente concreto para monitorar se a operação é chamada em um determinado contexto. Os parâmetros permitem estabelecer diferentes políticas – no formato de métricas, qualidade de serviço, entre outros; e eventos para serem monitorados no componente. Por exemplo, pode-se definir o método de obtenção de dados, o intervalo de tempo, ou atributos de QoS.

Funcionalidades externas necessárias à operação deste componente são:

- `notifyEvent(e: Event)`

Esta funcionalidade é necessária para notificar eventos aos clientes que subscreveram seus interesses através do método `subscribeEvent()`.

- `isAuthorized()`

Permite verificar se o cliente tem os acessos necessários às funcionalidades providas pelo componente *Descoberta de Serviços*.

4.2.4 Motor de Descoberta de Serviços (SDEngine)

Este é o principal subsistema do pacote *Descoberta de Serviços*, possibilitando a descoberta assistida ou automática de serviços, com base em requisitos definidos, procurando e ordenando-os de diferentes formas. Adicionalmente, permite a procura iterativa, utilizando diferentes algoritmos em diferentes etapas do processo de descoberta, denominado de processo de descoberta multi-fases.

Os componentes que pertencem ao subsistema Motor de descoberta de serviços (*SDEngine*) são ilustrados na figura 4.2.4.1 (NEXOF-RA-2,2010), sendo descritos a seguir:

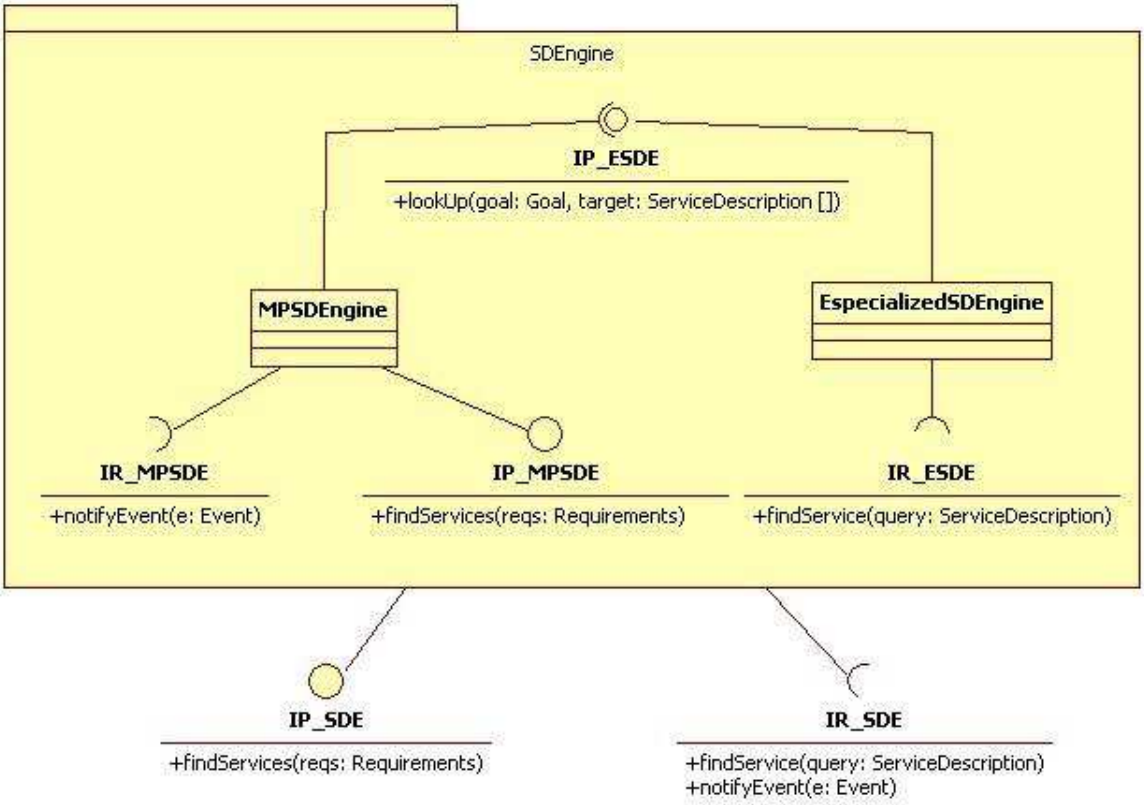

Fig. 4.2.4.1 Componentes do subsistema *Motor de Descoberta de Serviços*

4.2.5 Componente Motor de Descoberta Multi-fases (MPSDEngine)

Este componente é responsável por gerenciar as fases do processo de descoberta durante as atividades de procura e seleção. Ele realizará chamadas ao componente de *Descoberta de Serviços Especializado* (*EspecializedSDEngine*), de acordo com requisitos e a configuração recebida pelos demais componentes.

As funcionalidades providas por este componente são:

- `findServices(reqs: Requirements): ServicesRanking`

  Dado um conjunto de requisitos e indicadores sobre as fases do processo de *Descoberta de Serviços*, será realizado a ordenação de serviços chamando o componente *Motor Especializado de Descoberta de Serviços* (*EspecializedSDEngines*), descrito a seguir.

As funcionalidades necessárias à operação deste componente:

- `notifyEvent(e: Event)`

Este componente requer a funcionalidade de notificação de eventos para o componente responsável pelo gerenciamento dos mesmos.

4.2.6 Motor Especializado de Descoberta de Serviços (EspecializedSDEngine)

Este é o principal componente do subsistema *Motor de Descoberta de Serviços* e conseqüentemente do sistema de *Descoberta de Serviços*, uma vez que a operações de procura e seleção são realizadas neste componente.

Os componentes que compõem o subsistema *EspecializedSDEngine* são ilustrados na figura 4.2.6.1 (NEXOF-RA-3,2010), que serão descritos a seguir:

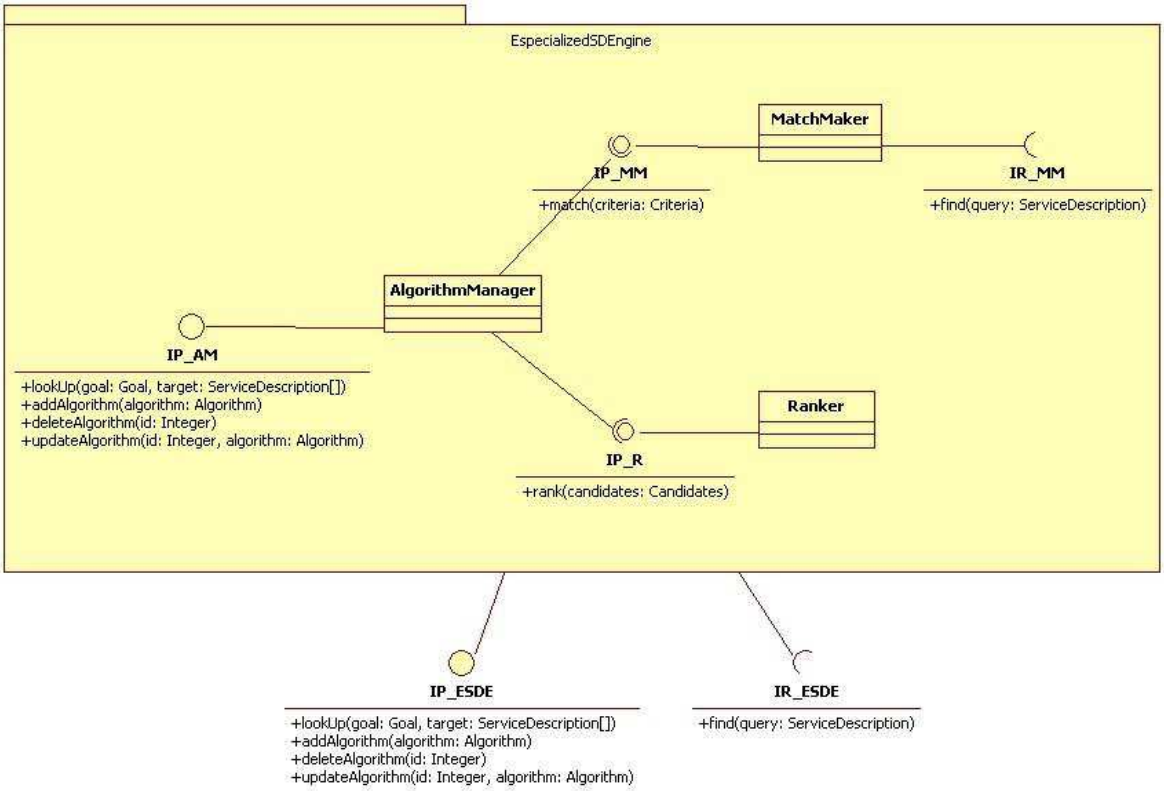

Fig. 4.2.6.1 Componentes do subsistema *Motor de Descoberta de Serviços Especializado*

4.2.6.1 Descobridor (Matchmaker)

Este componente é responsável pela descoberta de serviços, realizando as etapas de procura e seleção. Dependendo da estratégia selecionada, este componente utiliza parâmetros de procura, com base em descrições semânticas, palavras-chave ou qualquer outra estratégia. Ele acessará o *Registro de Serviços (Registry)* se necessário, realizando a procura segundo a lista de descrições de serviços.

Funcionalidades providas por este componente:

- `match(criteria: Criteria): Candidates`

Esta funcionalidade realiza a procura, segundo critérios definidos pelo algoritmo de procura, gerenciada pelo *AlgorithmManager*. Para evitar acessos desnecessários ao *Registro de Serviços*, o método exige uma lista de candidatos pré-selecionados.

Funcionalidades necessárias para operação do componente:

- `findService(query:ServiceDescription): AgentDescription[]`

Uma vez especificado a descrição dos serviços contidos no *Registro de Serviços* (*Registry*), serão apresentados os serviços que atendem ao conjunto de critérios de pesquisa.

4.2.6.2 Ordenador (Ranker)

Este componente, segundo a arquitetura de referência proposta pelo NEXOF (NEXOF-RA-3,2010), é responsável pela segunda etapa do processo de descoberta. Nesta etapa, os serviços descobertos pelo componente *Descobridor* (*Matchmaker*) são ordenados segundo uma estratégia definida pelo *Gerenciador de Algoritmos* (*AlgorithmManager*)

Este componente prove as seguintes funcionalidades:

- `rank(candidates: Candidates): ServicesRanking`

Esta funcionalidade ordena a lista de serviços candidatos, segundo um critério de ordenação.

4.2.6.3 Gerenciador de Algoritmos (AlgorithmManager)

Este componente é responsável pelo gerenciamento dos algoritmos de procura e seleção. As funcionalidades providas por este componente são:

- `lookUp(goal:Goal,target:ServiceDescription[]): ServicesRanking`

Esta funcionalidade realiza a descoberta de serviços com base nos requisitos recebidos e um filtro prévio do conjunto de candidatos de serviços. Estes requisitos

estão baseados descrições semânticas, objetivos ou palavras chaves que permitam definir critérios para especificação de serviços.

- `addAlgorithm(algorithm: Algorithm): Integer`

Esta funcionalidade permite incluir novos algoritmos necessários ao processo de descoberta, sob demanda, buscando o melhor desempenho deste processo.

- `deleteAlgorithm(id: Integer): Integer`

Esta funcionalidade permite excluir algoritmos não mais necessários ao processo de descoberta, sob demanda, buscando o melhor desempenho deste processo.

- `updateAlgorithm(id:Integer,algorithm:Algorithm): Integer`

Esta funcionalidade permite atualizar algoritmos existentes ao processo de descoberta, sob demanda, buscando o melhor desempenho deste processo.

Os algoritmos propostos neste trabalho serão descritos seguir.

4.3 Algoritmos para seleção adaptativa de serviços

Os algoritmos para seleção adaptativa de serviços, no contexto da IoS, são:

- Prospecção de padrões de utilização de serviços, baseado em requisitos não funcionais, utilizando o algoritmo de rede neural artificial S*elf-Organizing Map* (SOM);
- Árvore de decisão para interpretação e estruturação dos padrões em um formato hierárquico, utilizando o algoritmo de aprendizado por máquina ADAPTREE;
- Sistema de inferência baseado em lógica difusa, para tratamento de padrões inconsistentes através de regras ou políticas, utilizando o algoritmo *Adaptive*

Network-based Fuzzy Inference Systems (ANFIS), do tipo *Takagi-Sugeno* (SUGENO,1985).

A seguir serão descritos os parâmetros de entrada, funções e parâmetros de saída de cada algoritmo.

4.3.1 Algoritmo para prospecção de Padrões

O algoritmo SOM utiliza os serviços descobertos pelo componente *MatchMaker*, resultantes da procura com base em atributos funcionais, para formar um mapa bidimensional, que permite visualizar de forma qualitativa propriedades e relacionamentos dos dados que constituem classes ou agrupamentos de serviços.

4.3.1.1 Variáveis de Entrada

As variáveis de entrada são representadas pelo vetor de qualidade dos serviços, conforme definido pela eq. 4.3.1.1.1, descrita na seção 3.4.5 e apresentada novamente a seguir:

$$M = (q_{i,j}(s_i); 1 \leq i \leq m, 1 \leq j \leq 4) \qquad \text{(eq. 4.3.1.1.1)}$$

Em $\boldsymbol{M}$, cada coluna $\boldsymbol{q_i}$ corresponde a um serviço componente ($\boldsymbol{s_i}$), na qual cada linha $\boldsymbol{q_j}$ corresponde a um vetor de qualidade.

4.3.1.2 Funcionalidades providas

As seguintes funções, para treinamento não supervisionado do algoritmo SOM, utilizam os valores dos atributos do vetor de qualidade descritos anteriormente.

- `som_read_data(M): sD`
- `som_normalize(sD,'var'): sD`
- `som_make(sD): sM`
- `som_seqtrain(sM,sD): sM`

Estas funções são descritas a seguir:

- `som_read_data(M): sD`

Função responsável pela leitura da matriz ***M***. Fornece como saída, a matriz de treinamento $\boldsymbol{sD_{i\,x\,j}}$, na qual:

- ***i*** corresponde o número de serviços descobertos;
- ***j*** corresponde ao número de atributos de qualidade.

- `som_normalize(sD,'var'): sD`

Função responsável pela normalização dos dados de treinamento. A normalização tem por objetivo controlar a variação dos componentes do vetor de treinamento, evitando distorções. O método de normalização utilizada nesse caso é o método VAR (MATLAB,2010), isto é, *Normalização de Variança* (*Variance Normalization).* Por este método, transformam-se linearmente os dados resultando em uma variação unitária dos mesmos.

- `som_make(sD): sM`

Função responsável por criar, inicializar e treinar o algoritmo SOM. Utiliza as seguintes funções internas, descritas na biblioteca de programas SOM *Toolbox* (VESANTO;et al.,1999):

- `SOM_TOPOL_SCRUCT( );`
- `SOM_TRAIN_STRUCT( );`
- `SOM_DATA_STRUCT( );`
- `SOM_MAP_STRUCT( ).`

Inicialmente o número de unidades do mapa (M_{units}) é determinado. A função `SOM_TOPOL_STRUCT( )` é utilizada para determinar esse número, através da heurística ilustrada na eq. 4.3.1.2.1 (VESANTO;et al.,1999):

$$M_{units} = 5\sqrt{dlen} \qquad \text{(eq. 4.3.1.2.1)}$$

Na qual:

- ***dlen*** é um escalar que representa o volume de dados da base de treinamento, ou seja, o número *i* de serviços.

Após a definição do número de unidades do mapa, o tamanho do mapa é determinado. Basicamente, os dois maiores auto-valores do vetor ***sD*** de treinamento são calculados e a razão entre os comprimentos das laterais (comprimento e largura no caso de topologia plana da rede neural) é ajustada por essa razão, de forma que o valor do comprimento multiplicado pela largura do mapa seja aproximadamente igual ao número de unidades do mapa.

O algoritmo SOM é então linearmente inicializado, primeiramente ao longo de dois maiores auto-vetores se estes puderem ser determinados, caso contrário é utilizado inicialização randômica. Após a inicialização, o algoritmo SOM é treinado em duas etapas: macro treinamento (altos valores para a taxa de treinamento) seguido de ajuste fino (baixos valores para a taxa de treinamento).

Existem dois algoritmos para treinamento da SOM disponíveis na função `som_make()`:

- Treinamento Batch, no qual todos os dados de treinamento são utilizados ao mesmo tempo;
- Treinamento Seqüencial, no qual os dados de treinamento são utilizados de forma seqüêncial.

Se não for especificado o treinamento Seqüencial como argumento da função `som_make`, é utilizado o treinamento Batch. O modo Seqüencial será utilizado neste trabalho, permitindo maior adaptabilidade a possíveis mudanças na procura de serviços, utilizando a função de treinamento a seguir:

- `som_seqtrain(sM,sD): sM`

Dado um mapa `sM`, e os dados da matriz de treinamento `sD`, esta função realiza o treinamento de forma seqüência, resultando em um novo mapa de forma recursiva.

4.3.1.3 Variáveis de Saída

Do ponto de vista quantitativo, as variáveis de saída da etapa de treinamento da rede tipo SOM, são os próprios vetores protótipos (***sM***) ajustados topologicamente aos dados, de forma a preservar a posição relativa dos dados em um espaço n-dimensional reduzido a um espaço de bidimensional.

4.3.2 Algoritmo Agrupamento (clustering) e Sumarização

O objetivo do algoritmo de agrupamento e sumarização é fornecer informações quantitativas, no qual apenas os grupos de unidades do mapa, de interesse para formação dos agrupamentos, devem ser sumarizados. Cada unidade do mapa, a priori, pode ser considerada um agrupamento e sumarizada (quantificada) separadamente. Porém, esse procedimento tem um desempenho baixo para o contexto de seleção de serviços na IoS.

Por este motivo, segundo (VESANTO; et al.,1999), a etapa de agrupamento é realizada em duas etapas: primeiramente os serviços são agrupados pela SOM, resultando em vetores protótipos que compõem variáveis de entrada dessa etapa, denominados de *"protoclusters"*. Após isto, os mesmos são agrupados e quantificados. O maior benefício dessa abordagem é o ganho em desempenho computacional no tratamento dos serviços, que podem atingir um altíssimo número, dentro do contexto do IoS.

As variáveis de entrada, funcionalidades e variáveis de saída serão detalhadas a seguir:

4.3.2.1 Funcionalidades providas

- `som_kmeans(method, sD, k, epochs, verbose): [codes,clusters,err]`

Esta função implementa o algoritmo *k-means* (MATLAB,2010), com os seguintes parâmetros de entrada e saída, descritos a seguir.

Parâmetros de entrada:

- `method` : define o tipo de treinamento do algoritmo *k-means* - 'batch' ou 'sequencial';

- `sD` : matriz de treinamento *i x j* ;
- `k` : escalar que determina o número de centróides;
- `epochs`: escalar que determina o número de épocas de treinamento;
- `verbose`: escalar, se diferente de “0” exibe informações adicionais sobre o processo de agrupamento.

Parâmetros de saída:

- `Codes` : matrix de vetores de códigos (*codebook*) ;
- `Clusters`: escalar que representa o número de clusters de cada amostra;
- `err` : escalar que representa o erro total de quantização dos agrupamentos.

- `Davies_Bouldin_index(sD, cl, C, p, q) : [t,r]`

Esta função avalia o índice de Davies-Bouldin (DAVIES;BOULDIN,1979) para determinar o número de classes, com os seguintes parâmetros de entrada e saída:

Parâmetros de entrada:

- `sD` : Matriz de treinamento (*i x j*);
- `cl` : vetor unidimensional *n x 1* que representa o agrupamento correspondente a amostra de treinamento;
- `C` : matriz de vetores protótipo (*c x i*);
- `p` : escalar que representa a norma utilizada no algoritmo;
- `q` : escalar que representa a dispersão entre os agrupamentos.

Parâmetros de saída:

- `t` : escalar que representa o índice de Davies-Bouldin ;
- `r`: vetor (*c x 1*) que representa o maior valor do índice de Davies-Bouldin para cada agrupamento.

4.3.3 Algoritmo para interpretação e estruturação dos padrões em um formato hierárquico (árvore de decisão)

Como o algoritmo SOM não faz uso eficiente de classes pré-definidas, uma vez que o modo não supervisionado não considera os valores das variáveis de saída durante o aprendizado dos padrões, será utilizado o algoritmo ADAPTREE modificado para construção de uma árvore de decisão, que permita a interpretação e estruturação de padrões em um formato hierárquico.

Em uma árvore de decisão obtida pelo algoritmo ADAPTREE, cada nó corresponde a um atributo não classificado e cada ramo da árvore a um possível valor limiar para cada atributo. Os nós folhas especificam um valor esperado para classificar um determinado atributo segundo um caminho definido desde o nó raiz até o nó folha. Em uma árvore de decisão, cada nó deve estar associado aos atributos não classificados que maximizam a informação entre os atributos não considerados no caminho entre raiz e o nó folha. A métrica utilizada pelo ADAPTREE é a razão do ganho de entropia, modificado para a medida de informação, sendo diretamente proporcional a probabilidade do atributo pertencer à classe definida pelo nó.

Os valores da variável de saída são classes obtidas diretamente dos dados através da etapa de agrupamento e sumarização, através da função seguir:

- `adaptree (trainData,testSet,crossValidationSet, percentageSplit): [crispRuleSet, fuzzyRuleSet]`

4.3.3.1 Variáveis de entrada

As variáveis de entrada para o algoritmo ADAPTREE serão formadas de vetores de qualidade, contendo amostras aleatoriamente selecionadas da base de registro de serviços, as mesmas que foram utilizadas para o treinamento da SOM, acrescidas das classes obtidas na etapa de agrupamento e sumarização, compondo a base de treinamento estendida com suas respectivas classificações.

Conjunto de treinamento

A variável `trainData` contém os valores do vetor de qualidade acrescidos das classes resultantes do algoritmo de agrupamento e sumarização.

Dados de testes

A variável `testSet` contém os valores do vetor de qualidade, no qual segrega-se valores distintos para teste de acuracidade do ADAPTREE, comparando-se as classes obtidas através do algoritmo ADAPTREE com os classes dos conjunto de testes.

Conjunto de validação cruzada (Cross Validation)

A variável `crossValidationSet` contém os valores do vetor de qualidade, no qual segregou-se valores distintos para validação cruzada de acuracidade do ADAPTREE, comparando-se as classes resultantes do ADAPTREE com as classes resultantes do conjunto de validação cruzada.

Percentual de partição

A variável `percentageSplit` representa o valor percentual utilizado para segregar o conjunto de treinamento e compor de testes.

4.3.3.2 Variáveis de saída

Espera-se obter como variáveis de saída de uma árvore de decisão, os valores limiares em cada nó que definem a função de decisão do nó em questão, permitindo a indução de regras que, além de sumarizar os dados, permitiram classificar novas amostras.

Conjunto de regras estritas (crisp)

A variável `crispRuleSet` contém o conjunto de regras estritas e respectivas classificações segundo valores "*crisp*" ou valores dos atributos de qualidade. A funcionalidade

`ruleBuilding()`, adicional ao algoritmo ADAPTREE, é responsável pela leitura da árvore de decisão e pela construção da regras estritas.

Conjunto de Regras Difusas (Fuzzy)

A variável `fuzzyRuleSet` contém o conjunto de regras difusas e respectivas classificações segundo valores "*fuzzy*" ou valores dos atributos de qualidade. A funcionalidade `ruleFuzzyfication()`, adicional ao algoritmo ADAPTREE, é responsável pela leitura da árvore de decisão e pela construção da regras estritas.

4.3.4 Construção do motor de inferência difuso (fuzzy)

O motor de inferência difuso (*fuzzy*) é construído através do algoritmo ANFIS (SUGENO, 1985), sendo implementado através da função:

- `anfis(trnData,fismat,trnOpt,dispOpt,chkData,method): [fismat1,trnError,ss,fismat2,chkError,RuleSet]`

Nesta função, os parâmetros de entrada `trnOpt` (training options), `dispOpt` (*display options*), `chkData` (*checking data*), e `method` (*training method*) são opcionais e utilizaram valores *default*, descritos a seguir:

4.3.4.1 Variáveis de entrada

Dados de treinamento

A variável de entrada obrigatória `trnData` contém o vetor de qualidade com atributos difusos e as respectivas classes associadas, obtidos através das variáveis de saída da árvore de decisão.

Criação da estrutura do motor de inferência difusa

A variável de entrada obrigatória `fismat` define o número de número de atributos de qualidade e, conseqüentemente, o número de funções de pertinência de entrada e seu

formato de distribuição. O número de atributos de entrada é obtido através do vetor de qualidade e o formato de distribuição *default* das funções de pertinência é a distribuição "normal".

Opções de treinamento

A variável opcional `trnOpt` define os seguintes valores default para o treinamento do ANFIS:

- `trnOpt(1):` escalar que representa o número de épocas de treinamento, cujo valor default é "10";
- `trnOpt(2):` escalar que representa a tolerância ao erro, cujo valor default é "0";
- `trnOpt(3):` escalar que representa o tamanho do passo inicial, cujo valor default é "0,01" ;
- `trnOpt(4):` escalar que representa a taxa de decréscimo do passo de iteração do algoritmo de treinamento, cujo valor default é "0,9";
- `trnOpt(5):` escalar que representa a taxa de acréscimo do passo de iteração do algoritmo de treinamento, cujo valor default é "1,1".

Método de treinamento

A variável opcional `method` define o método de treinamento do ANFIS, para estimativa e ajuste das funções de pertinência:

- `method(1):` escalar que representa o método *Backpropagation*, cujo valor default é "1";
- `method(0):` escalar que representa o método híbrido *Backpropagation* com o método de *Least-Squares*, cujo valor default é "0";

O método *Backpropagation* será utilizado neste trabalho por apresentar um desempenho computacional melhor que o método híbrido, segundo (MATLAB,2010). A função interna `anfisTraining()` é responsável pela execução do treinamento segundo o método e opções de treinamento, conforme os valores default citados anteriormente.

4.3.4.2 Variáveis de saída

Estrutura do ANFIS para os dados de treinamento

A variável de saída `fismat1` corresponde ao ajuste das funções de pertinência das variáveis, que minimizam o erro de treinamento, quando não se utiliza dados para verificação.

Erro de treinamento

O erro de treinamento representa a diferença escalar entre o valor do dado de saída antes e após o treinamento. A variável `trnError` escalar representa a raiz quadrada da média do erro quadrático (*Root Mean Squared Error – RMSE*) (MATLAB, 2010) dos dados de treinamento em uma determinada época de treinamento.

Tamanho do passo de treinamento (step size - ss)

O valor inicial do passo de treinamento é representado pelo escalar “0,01” e a variável `ss` permite obter do passo de treinamento em uma determinada época de treinamento.

Dados de verificação (checking data)

Os dados de verificação, `chkData`, são utilizados para verificar a capacidade de generalização do ANFIS em uma determinada época de treinamento. Usualmente, segrega-se uma amostra distinta dos dados de treinamento para verificação.

A verificação de dados é importante para a atividade de aprendizado supervisionado, na qual o número de dados de treinamento é elevado e/ou existe ruído na medição dos valores dos atributos de qualidade.

Estrutura do ANFIS para os dados de verificação

A variável de saída `fismat2` corresponde ao ajuste das funções de pertinência das variáveis de entrada, que minimizam o erro de treinamento para os dados de verificação.

Erro de verificação

O erro de verificação representa a diferença escalar entre o valor do dado de saída antes e após a verificação. A variável `chkError` escalar representa a raiz quadrada da média do erro quadrático (*Root Mean Squared Error – RMSE*) (MATLAB,2010) dos dados de verificação em uma determinada época de treinamento.

Conjunto de regras (ruleSet)

A variável `ruleSet` contém o conjunto de regras após o treinamento do ANFIS. A função interna `ruleDefuzzyfication()`, é responsável por converter regras difusas em estritas, permitindo a comparação com os valores definidos pelo USLA.

4.4 Comportamento do sistema *Descoberta de Serviços*

Esta seção tem objetivo ilustrar a interação entre componentes do sistema *Descoberta de Serviços*, através de suas funcionalidades propostas pela arquitetura de referência para a IoS (NEXOF-RA-3, 2010).

4.4.1 Funcionalidade StartDiscovery

A figura 4.4.1.1 ilustra o comportamento da funcionalidade `startDiscovery()`, cujo primeiro passo é a confirmação de autorização para executar a operação, que depende da forma que o usuário deseja realizar a descoberta de serviços:

- Navegar através do registro de serviços, avaliando de forma ordenada a lista de serviços publicados;
- Definir algumas características simples que permitam filtrar serviços no registro, através de uma consulta a base de serviços;
- Definir requisitos funcionais e não funcionais e permitir que o *Motor de Descoberta de Serviços* procure e selecione os serviços, que melhor atendam aos mesmos.

A última opção ilustrada na figura 4.4.1.1, ou seja, a funcionalidade de procura (*find by search*) será descrita a seguir, na qual os componentes Registro (*Registry*) e Gerenciador

de Segurança (*Security Manager*) são componentes externos ao sistema *Descoberta de Serviços* e por este motivo não serão detalhados neste trabalho.

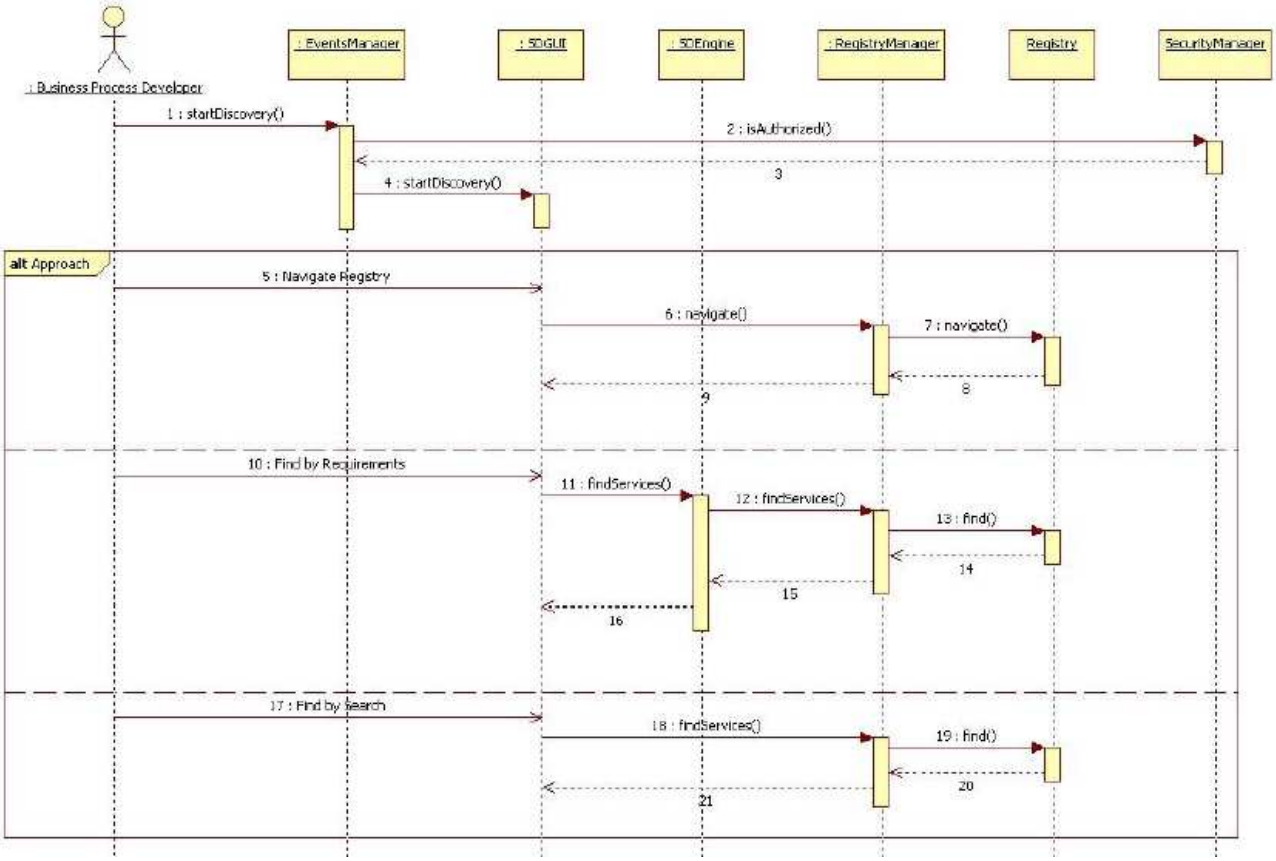

Fig.4.4.1.1 Diagrama de seqüência de componentes para a funcionalidade `startDiscovery()` (NEXOF-RA-3, 2010)

4.4.2 Funcionalidade de procura de serviço (findService)

A figura 4.4.2.1 ilustra como os componentes interagem para realização da funcionalidade de procura de serviços (`findServices()`). Estas interações não incluíram qualquer notificação de erros, a qual deve ser gerenciada pelo componente *Gerenciador de Eventos* (*EventsManager).*

Esta funcionalidade é orientada para componentes externos que requeiram procura de serviços, recebendo uma lista ordenada de melhores candidatos. Assim como na funcionalidade anterior, os componentes *Registro* e *Gerenciador de Segurança* são componentes externos ao sistema *Descoberta de Serviços*.

A realização da atividade de descoberta em três fases. O componente de *Descoberta de Serviços Multi-fases* (*MPSDEngine*) extraí os requisitos funcionais e realiza a chamada do

componente *Descoberta de Serviços Especializado* (*EspecializedSDEngine*), o qual implementa três abordagens para selecionar serviços. Durante as duas primeiras fases, este componente obtém serviços do registro (*Registry*) e seleciona alguns candidatos, enquanto na última fase ele utiliza um conjunto reduzido de candidatos, extraído do componente *MPSDEngine.*

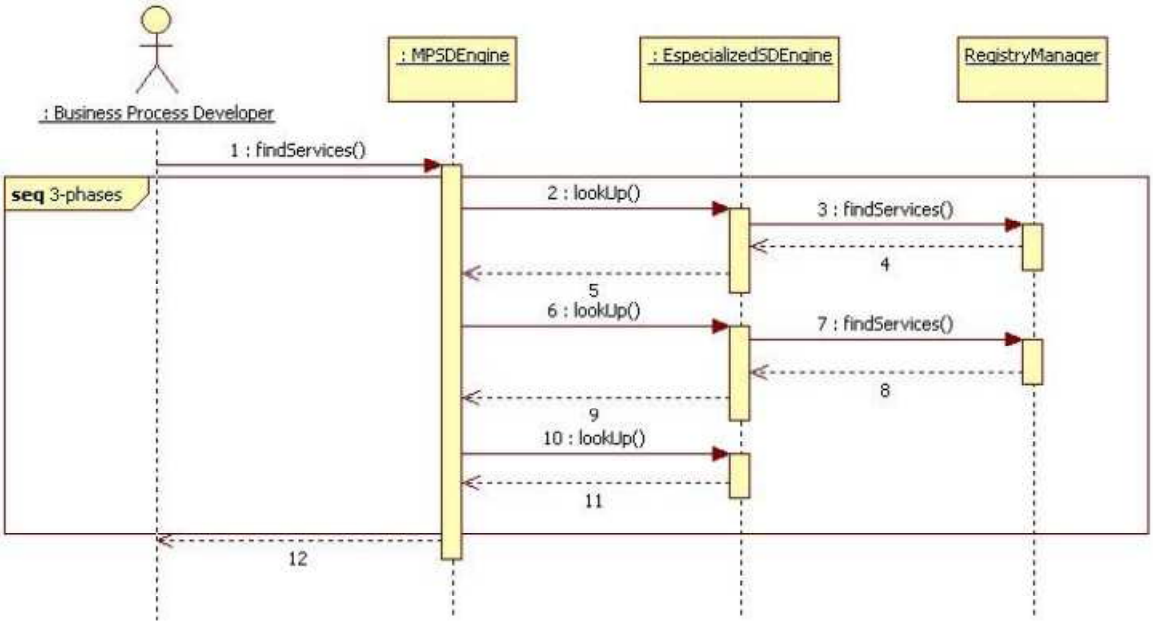

Fig. 4.4.2.1 Diagrama de seqüência de componentes para a funcionalidade `findServices` (NEXOF-RA-3, 2010)

4.4.3 Funcionalidade de busca de serviços (lookUp)

A figura 4.4.3.1 ilustra como os componentes interagem para realização da funcionalidade de busca de serviços (`lookUp()`). Segundo (NEXOF-RA-3, 2010) estas interações não incluíram qualquer notificação de erros, a qual deve ser gerenciada pelo componente *Gerenciador de Eventos* (*EventsManager*). Da mesma forma, nenhuma questão relacionada a segurança será gerenciada por este componente, segundo a mesma referência.

Durante a execução da funcionalidade de busca (`lookUp()`), o componente *Gerenciador de Algoritmos* extraí os requisitos e cria o componente de *Procura* (*MatchMaker*), segundo a estratégia de procura definida pelo mesmo. Após realizado a etapa de seleção, o componente *Gerenciador de Algoritmos* cria o componente *Ranker* para ordenar a lista de candidatos. No final do processo, a instância dos componentes *MatchMaker* e *Ranker* são eliminadas.

O componente *Gerenciador de Registros* (*RegistryManager*) é externo a esta funcionalidade e componentes para descoberta e ordenação de serviços, embora forneça acesso ao *Registro* (*Registry*).

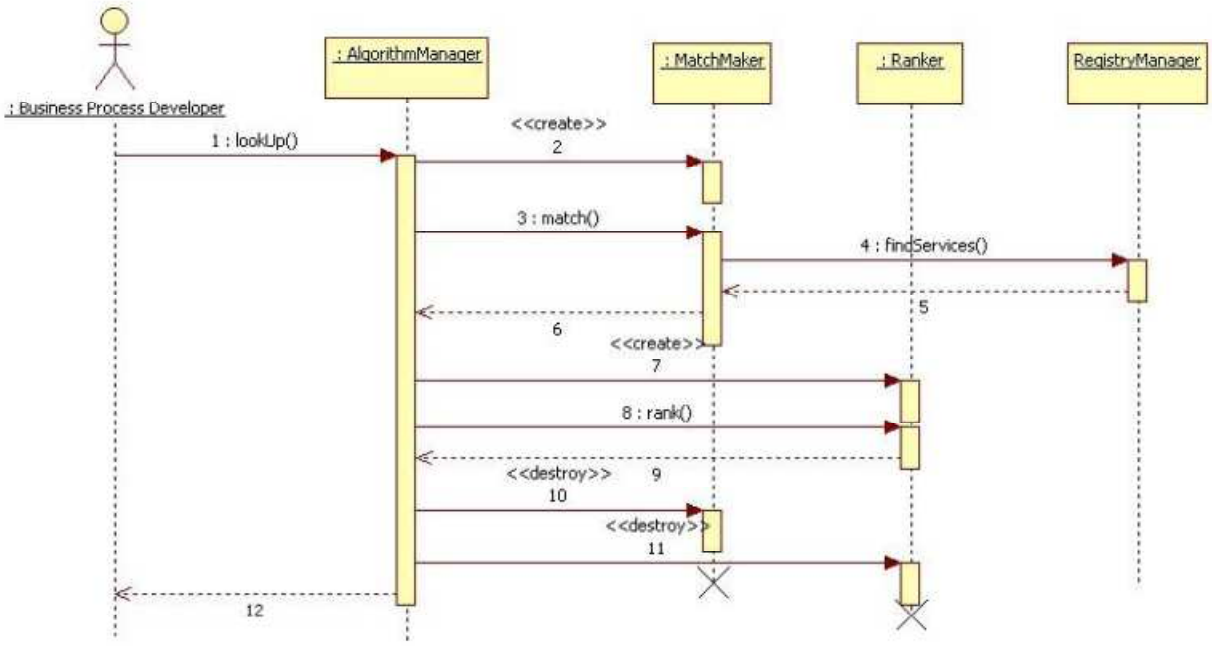

Fig. 4.4.3.1 Diagrama de seqüência de componentes para a funcionalidade `lookUp()` (NEXOF-RA-3, 2010)

4.5 Processo de seleção adaptativo e interativo de serviços

No processo de descoberta proposto para IoS (NEXOF-RA-3,2010) não foi detalhado a etapa de seleção de serviços, somente a etapa de procura e ordenação, corroborando com a hipótese que a etapa de seleção é usualmente subestimada ou simplesmente ignorada em várias soluções de descoberta de serviços (MIAN; et.al., 2009).

Na fig. 4.4.2.1, o processo proposto para seleção serviços inicia-se como uma requisição de procura serviços por parte de um consumidor de serviços, realizando um conjunto cíclico de etapas coordenadas, até o provisionamento de um ou mais serviços que satisfaçam a esta requisição, conforme ilustrado na figura 4.5.1.

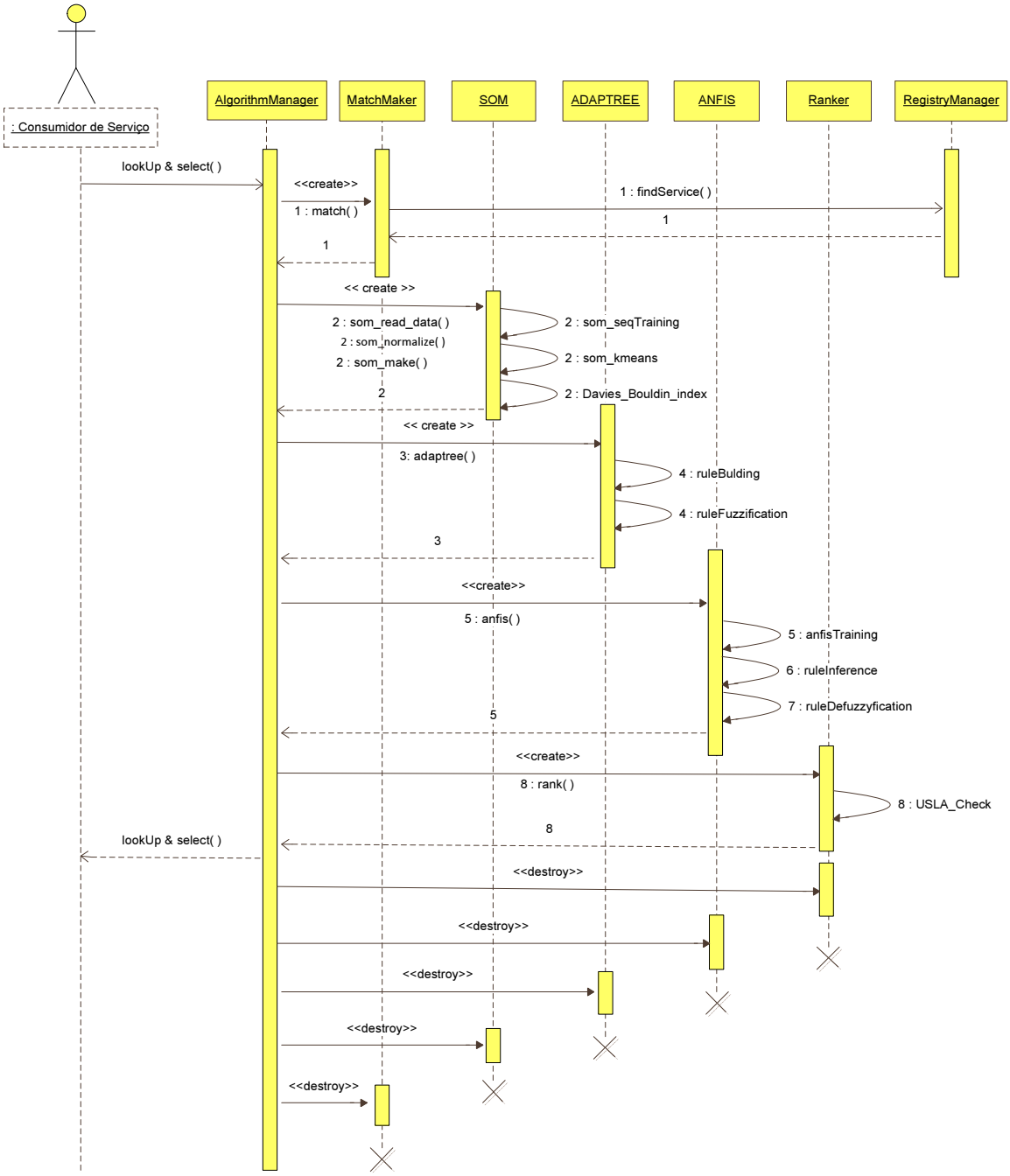

Fig. 4.5.1 Diagrama de seqüência entre componentes para procura e seleção de serviços para descoberta de serviços

A funcionalidade `lookUp()` foi modificada para incluir a seleção multi-fases de serviços, denominada de `lookUp&select()`, solicitando a criação dinâmica de uma instância do componente *Matcher* através do comando `<<create>>` na etapa (1).

Uma vez criado esta instância, o componente *AlgorithmManager* executa a funcionalidade `match`, que a realiza a procura com base em atributos funcionais. Esta procura é realizada através da função `findService()` que realiza a chamada ao componente gerenciador de registros (*RegistryManager*). O componente *RegistryManager* verifica em diferentes repositórios de serviços, quais são os candidatos que satisfazem a solicitação e retorna ao componente *Matcher*. Este por sua vez, consolida os resultados e comunica ao componente *AlgorithmManager* que a procura foi finalizada com sucesso.

Na etapa (2), o componente *AlgorithmManager* solicita a criação de uma instância do componente SOM, que implementa o algoritmo Self-Organizing Map, cuja estratégia de aprendizado é do tipo não supervisionada. O comando `<<create>>` realiza a chamada das funcionalidades `som_read_data()`, enviando os dados resultantes de procura funcional, bem como a normalização dos valores dos atributos de qualidade, através da funcionalidade `som_normalize()`. Uma vez finalizada a leitura e normalização dos atributos de qualidade dos serviços candidatos, a funcionalidade `som_make()` realiza a construção da grade de neurônios que realizarão o aprendizado não supervisionado dos padrões de agrupamento destes serviços, acionado a funcionalidade de treinamento seqüencial `som_seqTraining()`.

Finalizado a construção e treinamento da rede neural que compõem o componente SOM, são realizados as atividades de agrupamento de serviços, através da funcionalidade `som_kmeans(),` e sumarização dos agrupamentos, através da funcionalidade `Davies_Bouldin_index()`. Como resultado desta etapa, descobre-se o número de agrupamentos (classes) e quais serviços pertencem aos mesmos.

Na etapa (3), com base nos resultados da etapa anterior, o componente *AlgorithmManager* solicita a criação de uma instância do componente ADAPTREE, que implementa o algoritmo de indução de árvores de decisão adaptativa, através da funcionalidade `adaptree()`. Esta árvore é composta por atributos de QoS em nós intermediários, e por classes ou agrupamentos de serviços nos nós terminais, isto é, folhas da árvore de decisão.

Na etapa (4), realiza-se a extração das regras do tipo "***Se-Então***", lendo os valores dos atributos desde o nó raiz da árvore de decisão até os nós folhas. Estas regras contêm os valores dos atributos de QoS, geradas através da funcionalidade `ruleBuilding()`, segundo a linguagem adaptativa (DUTRA, et al.,2009), expressa por:

$$\textit{Crisp_ruleSet} = \{SE\ (AQ_{i,}(s_i)\ OPERADOR\ (q_{i,j}\ (s_i))\ E\ )^*\ ENTÃO\ s_i \in C_k\ \} \qquad \text{(exp. 4.5.1)}$$

Na qual:

- ***SE*** palavra reservada que corresponde ao operador lógico condicional para construção da regra "SE - ENTÃO";
- $AQ_i(s_i)$ corresponde ao Atributo de Qualidade do serviço (s_i);
- ***OPERADOR*** corresponde ao quantificadores "maior", "maior ou igual", "menor", "menor ou igual";
- $q_{i,j}\ (s_i)$ corresponde ao valor escalar do atributo de Qualidade do serviço (s_i), na qual cada linha q_j corresponde a um vetor de qualidade;
- ***E*** palavra reservada que corresponde ao operador lógico de conjunção para construção da regra;
- ******* símbolo que corresponde ao operador de *Kleene* (PAPADIMITRIOU; et al.,2004);
- ***ENTÃO*** palavra reservada que corresponde ao operador lógico condicional para construção da regra "SE - ENTÃO";
- $\in$ símbolo que corresponde ao operador pertinência de conjunto;
- C_k corresponde a classe ou agrupamento de serviços k, no qual $k \in \mathbb{N}, k > 1$.

Nesta expressão, a requisição de consumidores de serviços normalmente contém valores difusos, uma vez que o consumidor de serviços não conseguirá definir de forma inequívoca, todos os valores de atributos de QoS de serviços que ainda não utilizou ou que variam de acordo com os resultados da última procura de serviços.

Para tratar incertezas, as regras não difusas resultantes da etapa (4) são adaptadas para regras difusas, através da funcionalidade `ruleFuzzyfication`, expressa por:

$$\textit{Fuzzy_ruleSet} = \{SE\ (AQ_{i,}(s_i) \in FPe(q_{i,j}\ (s_i))\ E\ )^*\ ENTÃO\ s_i \in C_k\ \} \qquad \text{(exp. 4.5.2)}$$

Na qual:

- ***SE*** palavra reservada que corresponde ao operador lógico condicional para construção da regra "SE - ENTÃO";
- $AQ_i(s_i)$ corresponde ao Atributo de Qualidade do serviço (s_i);

- ***FPe($q_{i,j}$ (s_i))*** corresponde à Função de Pertinência de entrada "Alto" ou "Baixo", segundo o valor $q_{i,j}$ do atributo de Qualidade do serviço (s_i);
- ***E*** palavra reservada que corresponde ao operador lógico de conjunção para construção da regra;
- ******* símbolo que corresponde ao operador de *Kleene* (PAPADIMITRIOU; et al.,2004);
- ***ENTÃO*** palavra reservada que corresponde ao operador lógico condicional para construção da regra "SE - ENTÃO";
- $q_{i,j}$ corresponde ao atributo de qualidade do serviço candidato (s_i), na qual cada linha q_j corresponde a um vetor de qualidade;
- $\in$ símbolo que corresponde ao operador pertinência de conjunto;
- C_k corresponde a classe ou agrupamento de serviços k, no qual $k \in \mathbb{N}, k > 1$.

Através desta expressão, as regras difusas obtidas são utilizadas para treinar, de forma supervisionada na etapa (5), as funções de pertinência que definem os termos difusos "*alto*" ou "*baixo*" no componente ANFIS, através da funcionalidade `anfisTraining()`.

Uma vez completa a etapa (5), utilizando uma amostra de serviços, executa-se a inferência para os demais serviços, através da combinação das funções de pertinência, gerando uma superfície de decisão difusa como resultado da etapa (6), através da funcionalidade `ruleInference()`.

Na etapa (7), através da funcionalidade `ruleDefuzzyfication()`, calcula-se o centro de massa da superfície de decisão difusa, gerando valores de atributos de QoS, que podem ser comparados com as definições do USLA, na etapa (8). Nesta etapa, o componente *Ranker* é instanciado através da função `<<create>>` e o componente *AlgorithmManager* solicita a comparação dos atributos estritos ou difusos definidos no USLA, através da função `USLA_Check()`, gerando uma lista ordenada dos serviços candidatos através da funcionalidade `rank()`.

Se o grau de atendimento for alto, o serviço selecionado atende ao USLA contratado e a descoberta do serviço ou serviços está encerrada, gerando uma proposta de serviço para o consumidor. Neste caso, as instâncias dos componentes *Rank*, ANFIS, ADAPTREE, SOM e *MatchMaker* são eliminados através do comando `<<destroy>>`.

Se o grau de atendimento for baixo, repete-se a etapa (1), realizando uma nova procura e seleção de serviços. Os serviços resultantes são comparados com os padrões já

capturados pelo serviço de SOM, atualizando as classes de serviços, caso necessário na etapa (2). Esta atualização dispara uma indução incremental da árvore de decisão adaptativa na etapa (3), implicando em uma nova execução das etapas posteriores.

Caso não haja necessidade de atualizar novos padrões, realiza-se uma nova comparação com os parâmetros definidos no USLA da etapa (7), realizando-se interativamente as atividades (8) e (1) até alcançar um grau de atendimento alto do USLA.

Um "alto" grau de atendimento do USLA, segundo percepção do usuário final, implicará, neste caso, em:

- O "menor" preço do serviço;
- O "menor" tempo de execução do serviço;
- A "maior" disponibilidade e confiabilidade do serviço.

Sob o ponto de vista da rede de acesso, considerando a qualidade fim-a-fim do provisionamento ao consumo de serviços, um alto grau de atendimento do USLA implicará, neste caso, em:

- A "maior" disponibilidade de nós provedores de serviços;
- A "maior" confiabilidade de nós provedores de serviços;
- O "menor" atraso na comunicação entre nós provedores e consumidores.

4.6 Comentários

Este capítulo apresentou os principais requisitos funcionais e não funcionais que deverão ser atendidos pela arquitetura de referência da Internet de Serviços, cujos componentes e funções, bem com o comportamento dos mesmos foram descritos.

Inserida contexto desta arquitetura, foram apresentados os algoritmos SOM, ADAPTREE e ANFIS como os algoritmos para agrupamento e sumarização de padrões de serviços, cujas vantagens e desvantagens serão descritas a seguir:

Vantagens do algoritmo *Self-Organizing Map* (SOM):

- Independência de uma ontologia comum para classificar agrupamentos de serviços;
- Agrupamento baseado em um número irrestrito de atributos de QoS;
- Algoritmo iterativo, permitindo classificação de padrões sem a necessidade da leitura de todos os dados simultaneamente.

Desvantagens do algoritmo *Self-Organizing Map* (SOM):

- Algoritmo opaco, não permitindo a extração de regras que definem os agrupamentos de serviços;
- Não trata incertezas nos dados.

Vantagens do algoritmo *Árvore de Decisão Adaptativa* (ADAPTREE):

- Relações de agrupamentos extraídos no formato “SE-ENTÃO” , facilitando o treinamento do ANFIS;
- Algoritmo adaptativo, permitindo aprendizado iterativo sem necessidade de conhecer todos os dados a priori;
- Permite a simplificação de regras, através da medida da razão do ganho de entropia.

Desvantagens do algoritmo *Árvore de Decisão Adaptativa* (ADAPTREE):

- Aprendizado supervisionado, requerendo treinamento;
- Não trata incerteza nos dados.

Vantagens do algoritmo *Adaptive Network-based Fuzzy Inference System* (ANFIS):

- Combina algoritmo de inferência difusa com redes neurais artificiais para ajuste das funções de pertinência;
- Permite a comparação de atributos difusos ou não dos atributos de QoS dos agrupamentos de serviços com requisitos difusos ou não de USLA;
- Permite análise multidimensional de atributos.

Desvantagens do algoritmo *Adaptive Networked Fuzzy Inference System* (ANFIS):

- Aprendizado supervisionado, requerendo treinamento;
- Desempenho do mecanismo de inferência difusa depende do número de regras a avaliar.

5. ESTUDO DE CASO E PROPOSTA PARA PROVA DE CONCEITO

O objetivo de capítulo é descrever o estudo de caso e uma proposta de prova de conceito (*Proof of Concept, PoC*), utilizada para implementar a solução proposta para a seleção de serviços, denominada de *Middleware* de Seleção Adaptativa e Interativo (MSAI).

O capítulo é composto pelas seguintes seções:

- A seção 5.1 descreve o estudo de caso e a proposta para a PoC: seus objetivos, critérios de avaliação e os resultados esperados;
- As seções 5.2 e 5.3 descrevem os artefatos de *software* concretos, que implementam os componentes abstratos descritos no capítulo 4, e como os mesmos interagem entre si para a realização do processo de procura e seleção de serviços segundo o estudo de caso;
- A seção 5.4 compara os resultados esperados com os resultados obtidos, com base nos critérios de avaliação da proposta de prova de conceito;
- A seção 5.5 encerra o capítulo com comentários.

5.1 Descrição do estudo de caso e proposta para Prova de Conceito

O objetivo do estudo de caso é implementar um protótipo utilizando artefatos de *software*, para avaliar requisitos não funcionais do processo de descoberta de serviços, proposto para arquitetura da IoS (NEXOF-RA-1,2010). O cenário de negócios, deste estudo de caso, é composto por serviços relacionados aos processos de compras, de diferentes tipos de produtos, conforme descrito no anexo deste trabalho, agrupados segundo os domínios funcionais descritos na tabela 5.1.1.

Nesta tabela, a classificação funcional agrupa os 578 serviços, através da medida de similaridade dos atributos funcionais destes serviços (DAMLS,2001), gerando 28 classificações funcionalmente distintas para os serviços resultantes da etapa de procura funcional.

Estes serviços serão providos e consumidos por atores que estão imersos em ambientes de computação ubíqua, permitindo a mobilidade necessária à realização das interações entre pessoas, entre pessoas e máquinas e entre máquinas. Na Internet do Futuro, segundo (NEXOF-RA-1,2010), pressupõe-se que estas interações sejam realizadas através de redes de comunicação móveis Ad-Hoc, dispensando a necessidade de uma infra-estrutura fixa.

Tabela 5.1.1 Domínio dos serviços utilizados para prova de conceito

	Serviços	**Classificação Funcional**
Domínio Funcional	**Quantidade**	**Quantidade**
Educação	135	6
Cuidados com Saúde	52	1
Alimentação	25	1
Viagem	106	6
Comunicação	29	2
Economia	206	11
Armamentos	25	1
Total	**578**	**28**

A busca de melhor serviço, sob o ponto de vista do consumidor, objetiva aumentar suas possibilidades de escolha de provedores que melhor atendam suas necessidades. Desta forma, caberá ao consumidor formular uma descrição dos serviços que necessita, a qual deve incluir requisitos funcionais e não funcionais, tais como melhor preço, maior disponibilidade do serviço, maior confiabilidade e menor atraso na execução do mesmo. Estes requisitos devem possibilitar a procura, seleção e comparação dos respectivos provedores destes serviços.

Tendo escolhido um provedor de serviços, o consumidor solicita os detalhes do nível de qualidade do serviço (*Service Level*, SL) o qual o provedor garante cumprir. Com estas informações, o consumidor completará um modelo de SL, o qual especifica os níveis de qualidade que mais se assemelham aos seus requisitos.

Este modelo de nível de serviço (SL) completo é enviado ao provedor com o objetivo de obter um acordo sobre o nível de serviço sob o ponto de vista do consumidor (*User Service Level Agreement*, USLA) (SERRA,2007). Confirmado ou acordado o USLA, o provedor envia ao consumidor a referência do ponto de localização do serviço, para que o mesmo possa iniciar o consumo do mesmo.

Durante o consumo do serviço, ambos provedores e consumidores, deverão coletar e analisar informações de desempenho do mesmo. Isto permitirá que o serviço atenda aos requisitos do usuário, na busca do atendimento do SLA acordado entre as partes.

Durante o uso do serviço ou ao final do período de utilização do serviço, o consumidor poderá realizar o pagamento pelo uso do serviço. A forma de pagamento e os pagamentos propriamente ditos, também estão definidos no SLA acordado entre o

consumidor e provedor, antes do início da utilização do mesmo, não se aplicando ao cenário de negócios proposto.

O cenário de negócios deste estudo de caso considera:

- Qualquer consumidor de serviço que possa definir diferentes necessidades de forma estrita ou difusa, possibilitando a procura e a seleção de serviços com base nestas definições;
- Qualquer provedor de serviço que publique sua oferta de serviços, buscando atender aos níveis de qualidade de serviço sob o ponto de vista dos consumidores.

As premissas, sob o ponto de vista de consumidores e provedores, que necessitam ser satisfeitas para tornar possível este cenário são:

- Existe um entendimento comum entre as partes, consumidores e provedores, sobre o significado dos requisitos funcionais e não funcionais do serviço;
- O processo de descoberta está baseado em procura semântica dos atributos funcionais e seleção adaptativa e interativa dos atributos não funcionais;
- Os atributos não funcionais conterão ambigüidades que fornecerão oportunidades para negociação do USLA entre provedores e consumidores;
- Ambas as partes em acordo, devem ser capazes de acessar, se necessário, informações de desempenho do serviço, auxiliando no gerenciamento do USLA.

5.1.1 Requisitos funcionais para implementação da proposta da PoC

Para implementação e avaliação dos resultados esperados da PoC, segundo o cenário descrito anteriormente, serão utilizados os requisitos funcionais utilizados na arquitetura de referência para IoS (NEXOF-RA-3,2010) para descoberta de serviços, conforme descrito na seção 4.1.1.

5.1.2 Requisitos não funcionais e resultados esperados

Para avaliação dos resultados esperados da proposta da PoC, segundo descrito na seção 4.1.2, serão utilizados os critérios de qualidade definidos pela arquitetura de referência

para IoS (NEXOF-RA-3,2010), acrescidos dos critérios de avaliação segundo tabela 5.1.2 a seguir:

Tabela 5.1.2 Requisitos não funcionais e resultados esperados

Requisitos não funcionais	Critérios de avaliação	Resultados esperados
Capacidade de integração	Critério subjetivo, tratando-se de um protótipo	Alta
Capacidade de interoperação	Critério subjetivo, tratando-se de um protótipo	Alta
Acuracidade	Número de casos classificados com sucesso em relação ao total de casos avaliados	Alta
Desempenho	Tempo de resposta desde a requisição até a proposta de um ou mais serviços	Baixo
Escalabilidade	Relação entre a capacidade computacional do sistema e o tamanho do sistema para atender uma maior demanda futura	Alta
Segurança	Não será avaliado na PoC	Alta

Nesta tabela, “Alta” e “Baixo” representam valores difusos que permitem avaliar, relativamente, os valores dos requisitos não funcionais esperados como resultados da solução proposta.

5.2 Artefatos de *software* utilizados na implementação do protótipo

A implementação de um protótipo, para a seleção adaptativa e interativa de serviços no contexto da IoS, utilizará soluções de *middleware* conforme ilustrado na figura 5.2.1.

O *Middleware para Seleção Adaptativa e Interativa* (MSAI) de serviços permite abstrair o processo de seleção de serviços das camadas adjacentes, formada por provedores e consumidores, implementando os componentes concretos SOM, ADAPTREE, ANFIS, ALGORITHM MANAGER e RANKER, implementados através dos *softwares*:

- Para implementação dos componentes abstratos SOM, ANFIS e RANKER, foram utilizados funções do *Software Matlab*® 2010, disponíveis através das bibliotecas SOM Toolbox 2.0 (VESANTO;et al.,1999), FUZZY Toolbox (FUZZY, 2011) e RANKING Toolbox (RANKING,2011) respectivamente.

- Para implementação do componente abstrato ADAPTREE, foi utilizado as bibliotecas de funções do WEKA (WEKA,2010), que foram adaptadas segundo os autores PISTORI e NETO (2002);
- Para implementação do componente abstrato ALGORITHM MANAGER, foi utilizado a solução REDS (CUGOLA;NITTO,2008), denominado simplesmente como BROKER, no contexto do protótipo.

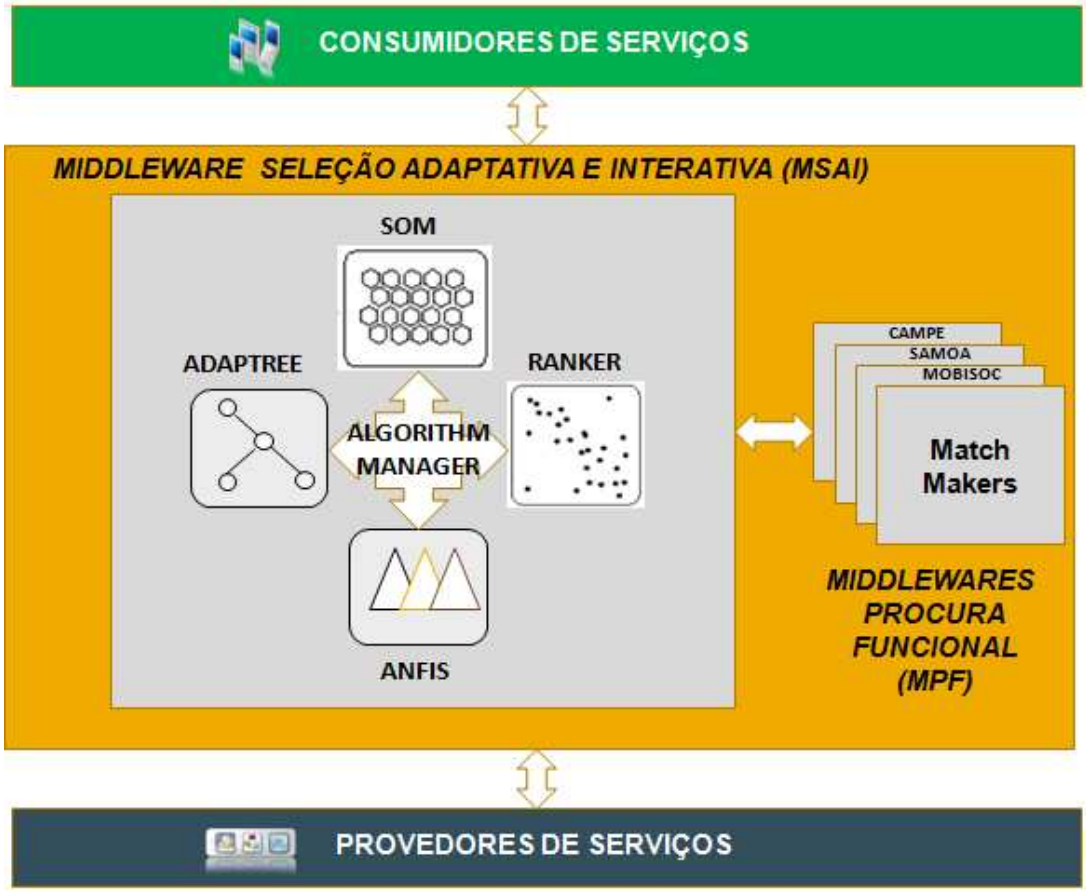

Fig. 5.2.1 Solução de *Middleware para Seleção Adaptativa e Interativa* (MSAI) de serviços

O *Middleware de Procura Funcional* (MPF) abstrai a captura e coleta de preferências e alterações de contexto de usuários ou consumidores de serviços, que interagem entre si através de redes móveis ou fixas, executando as funcionalidades do componente abstrato *MatchMaker* descrito no capítulo anterior. Além da percepção sobre o ambiente, cabe a este componente o gerenciamento, personalização e adaptação de serviços, que atendam às preferências dos usuários em um determinado contexto, através da utilização de componentes concretos para redes móveis, tais como: SAMOA (BOTTAZZI;et al.,2008), CAMPE (BOTAZZI;et al.,2007), e MOBISOC (GUPTA;et al.,2009), mencionados na seção 2.5.

A implementação do processo de seleção de serviços descrito na seção 4.5.2, será descrito na próxima seção, bem como os resultados obtidos através do protótipo.

5.3 Implementação do processo de seleção de serviços

O processo proposto para a seleção serviços móveis inicia-se como uma requisição de procura serviços por parte de um consumidor, realizando um conjunto cíclico de etapas coordenadas, até o provisionamento de um ou mais serviços que satisfaçam a esta requisição, conforme ilustrado na figura 5.3.1.

O tratamento da requisição de serviço, por parte de um consumidor de serviços, define dois modos distintos de atendimento: proativa e reativa.

No modo reativo, espera-se a ocorrência de uma requisição de serviços para o *Middleware de Procura Funcional* (MPF) iniciar o processo de procura funcional, com base em atributos funcionais, isto é, atividade (1) ilustrada na figura 5.3.1. No modo proativo, não é necessário a ocorrência de uma requisição de serviços, por parte do consumidor, para que o *Middleware de Procura Funcional* (MPF) inicie o processo de procura funcional.

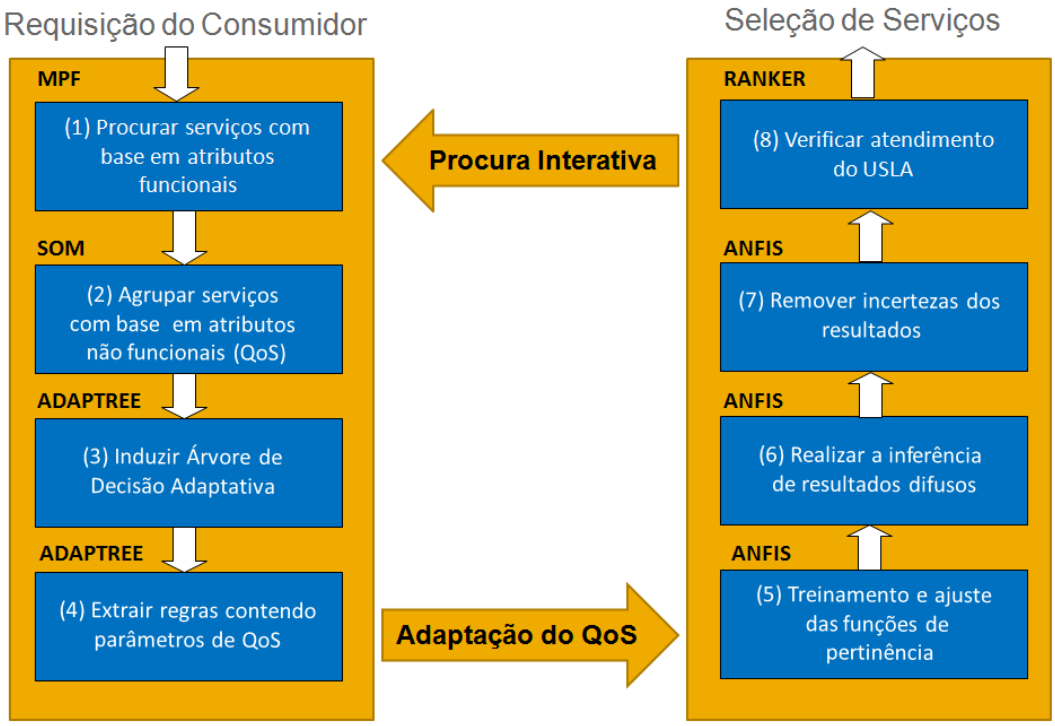

Fig. 5.3.1 Atividades do processo de seleção proposto segundo componentes do protótipo

Em ambos modos, o escopo de procura funcional baseia-se em buscas segundo estratégias definidas no MPF:

- Busca do contexto ou localização do usuário final, a exemplo da solução SAMOA;
- Busca por pessoas que fazem parte da rede contatos, ou rede social, do usuário final, a exemplo da solução CAMPE;
- Com base em uma combinação de ambas as estratégias, isto é, pessoas que fazem parte de rede de contatos e que estão fisicamente próximas à localização do usuário final, a exemplo da solução de MOBISOC.

Adotou-se como premissa, para implementação deste protótipo, que o MPF já realizou a atividade de pesquisa de serviços disponíveis, isto é, a funcionalidade de `findService()` retornou 578 serviços para serem avaliados, provenientes de um processo de procura funcional sobre um processo de negócios de compra (DAMLS,2011), descritos no apêndice deste trabalho.

Em modo de operação proativo, o componente SOM é invocado pelo componente BROKER para realizar o agrupamento de serviços, conforme etapa (2).

Segundo avaliação do índice de *Davies-Bouldin* (DAVIES;BOULDIN,1979), ilustrado na figura 5.3.2 (a), cujo valor mínimo indica a existência de quatro classes de serviços:

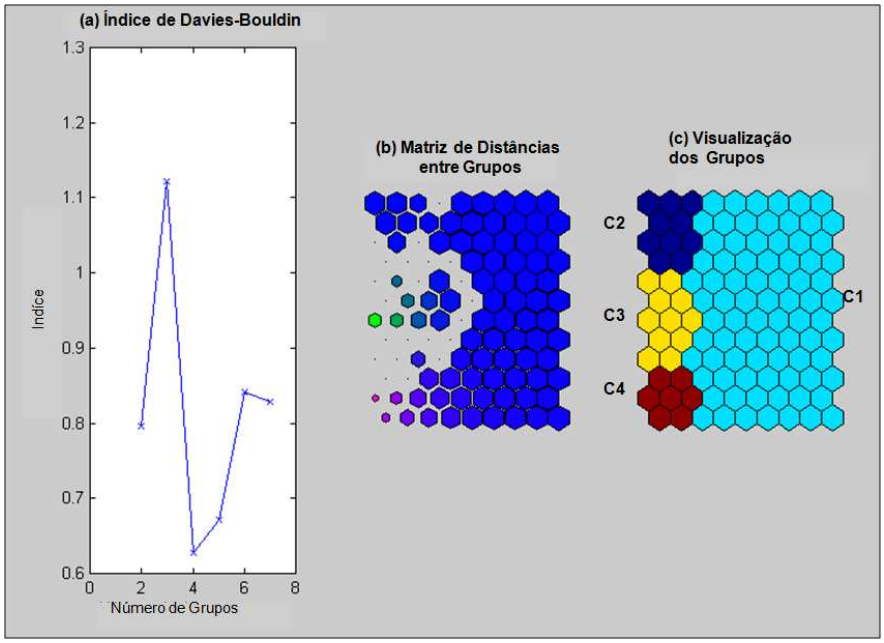

Fig. 5.3.2 Resultados do serviço SOM: (a) Índice de Davies-Bouldin (b) Matriz de distâncias e (c) Visualização de Agrupamentos

A figura 5.3.2 (b) ilustra a matriz de distâncias entre as unidades (neurônios) que constituem o mapa do serviço SOM, onde o tamanho de cada unidade representa o grau de similaridade dos serviços dentro de um agrupamento. A figura 5.3.2 (c) permite a visualização dos quatro agrupamentos ou classes de serviços, que servirão de base para o treinamento ou aprendizado supervisionado do serviço ADAPTREE, induzindo uma árvore de decisão como descrito na etapa (3) e ilustrado na figura. 5.3.3.

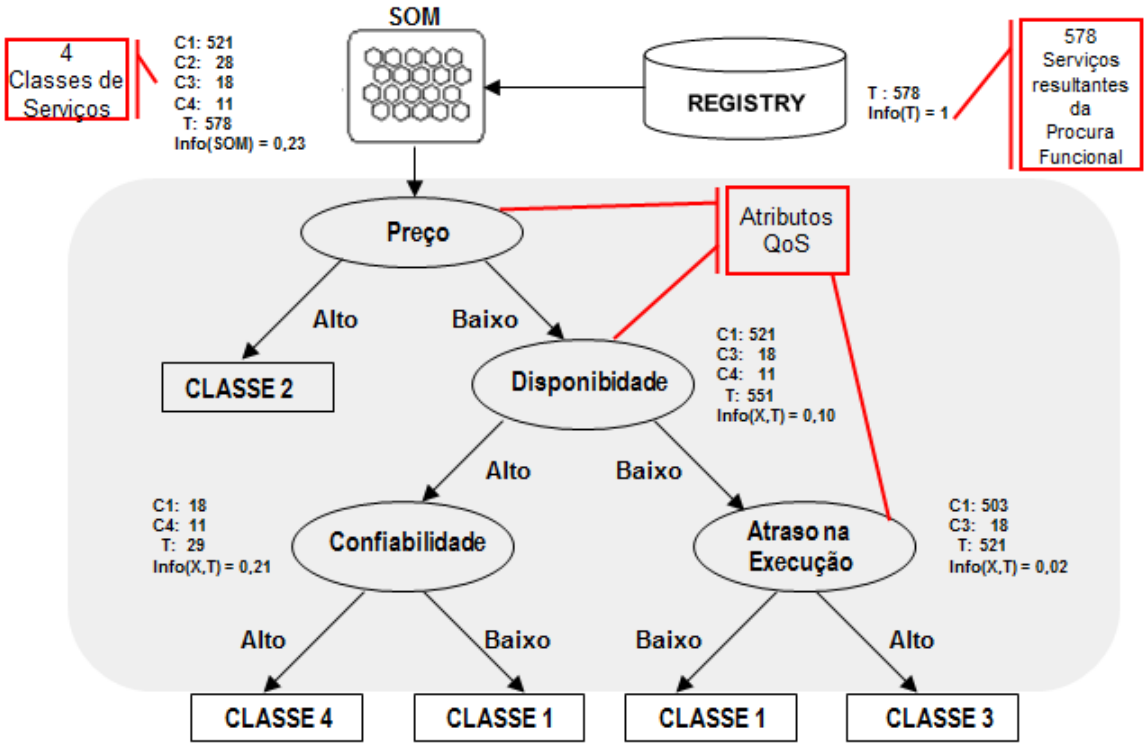

Fig. 5.3.3 Indução da Árvore de Decisão a partir das quatro classes de serviços

Na figura 5.3.3, "C_i:*X*" representa o número de serviços selecionados que pertencem a cada agrupamento "*i*". Os termos difusos (*fuzzy*) "Alto" e "Baixo" representam a transformação dos valores específicos dos atributos em valores incertos, para treinamento do ANFIS na etapa (5). *Info(X,T)* representa a medida do grau de entropia de informação em cada nó da árvore de decisão, na qual *X* é um agrupamento de serviços resultante da escolha "Alto" ou "Baixo" para o atributo em avaliação e *T* representa o total de serviços do repositório, em um dado momento da ciclo de seleção.

Para coletar as regras "*SE-ENTÃO*" a partir da árvore de decisão, basta solicitar ao serviço ADAPTREE a consulta a todos os possíveis ramos que partem do nó de origem e

conduzem até os nós folha, contendo as classes de serviços. Conforme ilustrado na figura 5.3.3, extrai-se as seguintes regras difusas:

Regra #3:
SE
Preço := "Baixo" E
Disponibilidade (serviçoi) := "Baixo" E
Atraso de Execução := "Baixo"
ENTÃO
serviço$_i$ PERTENCE A classe #1

Regra #2:
SE
Preço := "Baixo" E
Disponibilidade (serviçoi) := "Baixo" E
Tempo de Execução := "Alto"
ENTÃO
serviço$_i$ PERTENCE A classe #3

Regra #1:
SE
Preço := "Alto"
ENTÃO
serviçoi PERTENCE A classe #2

Regra #4:
SE
Preço := "Baixo" E
Disponibilidade (serviçoi) := "Alta" E
Confiabilidade (serviçoi) := "Baixa"
ENTÃO
serviço$_i$ PERTENCE A classe #1

Regra #5:
SE
Preço := "Baixo" E
Disponibilidade (serviçoi) := "Alta" E
Confiabilidade (serviçoi) := "Alta"
ENTÃO
serviço$_i$ PERTENCE A classe #4

Fig. 5.3.4 Regras difusas obtidas através etapa (4) do processo de seleção

Como resultado da etapa (4), temos a geração de cinco regras que delimitam o escopo de treinamento do ANFIS, determinando que apenas cinco, entre dezesseis regras, serão necessárias para o aprendizado supervisionado do ANFIS, conforme ilustrado na figura 5.3.5.

Conforme descrito na etapa (5) e ilustrado na figura 5.3.5, as funções de pertinência de entrada são ajustadas através de treinamento supervisionado, utilizando as regras geradas a partir da árvore de decisão. O operador lógico "E" foi utilizado para concatenar atributos, conforme definidos nas regras #1 a #5.

Foram selecionados de forma aleatória, uma amostra de 469 serviços do repositório de serviços resultante da procura funcional, divididos em 3 conjuntos para o treinamento do ANFIS, conforme ilustrado na figura 5.3.6, a seguir.

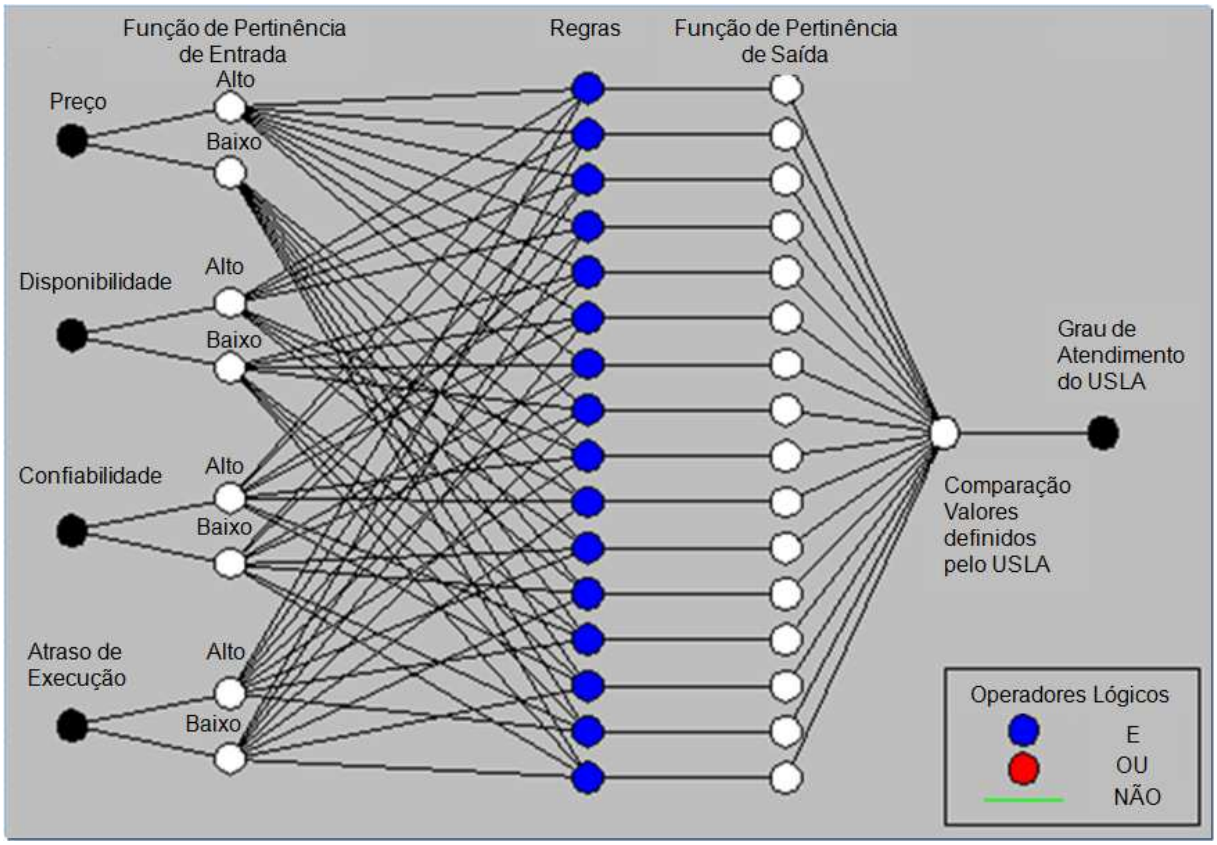

Fig. 5.3.5 Estrutura do ANFIS: Entradas, funções de pertinência, regras e saídas

- 200 amostras foram utilizadas para o treinamento das funções de pertinência de entrada e comparadas com a classe resultante do ANFIS após o treinamento, como ilustrado na figura 5.3.6 (a).
- 169 amostras foram utilizadas para testes dos resultados, como ilustrado na fig. 5.3.6(b). Avaliou-se a convergência do erro absoluto entre a classe esperada e a classe de serviço resultante do ANFIS, durante as 500 iterações de treinamento, como ilustrado na fig. 5.3.6 (d).
- 100 amostras foram utilizadas para verificação final dos resultados, como ilustrado na fig. 5.3.6 (c). Avaliou-se a convergência do erro absoluto entre a classe esperada e a classe de serviço resultante do ANFIS, durante as 500 iterações de treinamento deste conjunto amostral, como ilustrado na fig. 5.3.6 (d).

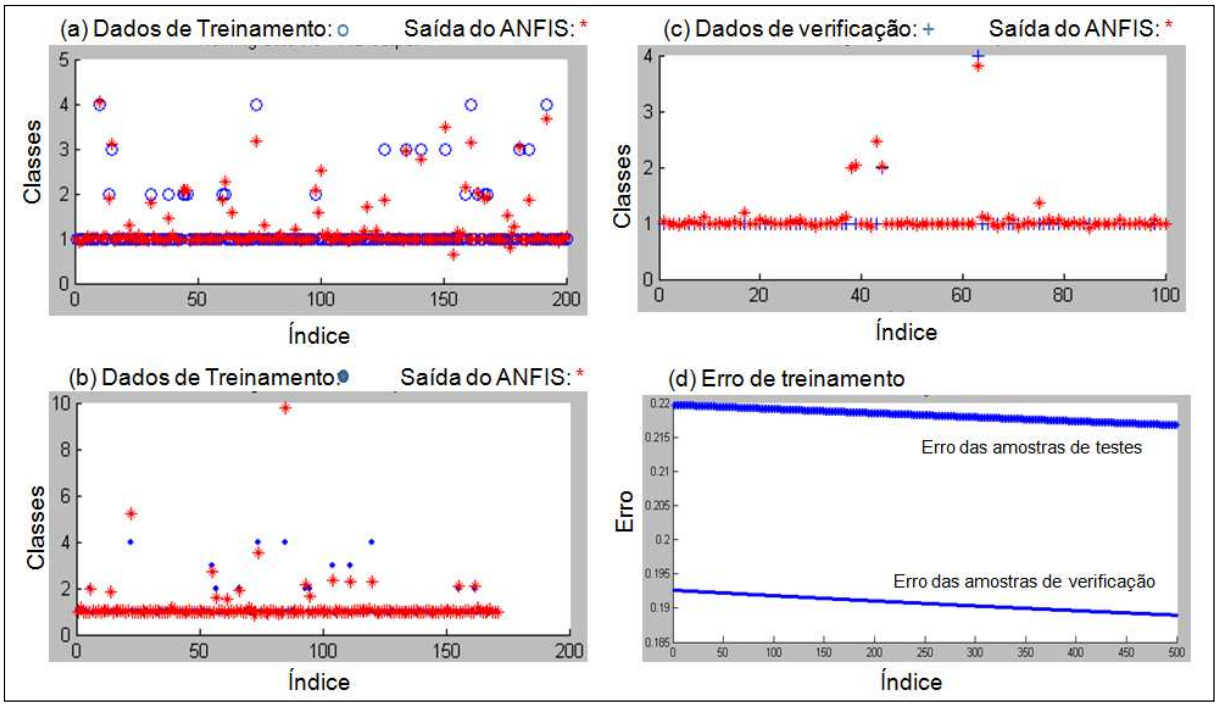

Fig. 5.3.6 Avaliação dos resultados após o treinamento do ANFIS

Adicionalmente, a Fig. 5.3.6 (d) ilustra a redução da taxa de erro, com o número de épocas de treinamento, não justificando um aumento do número de épocas de treinamento, para este conjunto de dados.

Uma vez ajustadas as funções de pertinência dos atributos de QoS dos serviços em análise, realiza-se a etapa de inferência difusa (6), gerando superfícies difusas a partir da combinação destes requisitos, como ilustrado na figura 5.3.7 e 5.3.8, a seguir.

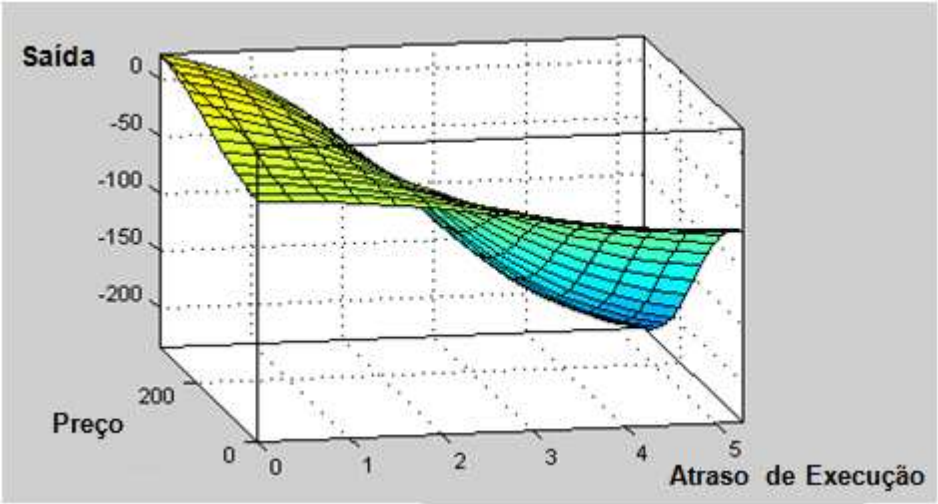

Fig. 5.3.7 Superfícies difusas a partir da combinação dos requisitos não funcionais Preço e Atraso de Execução

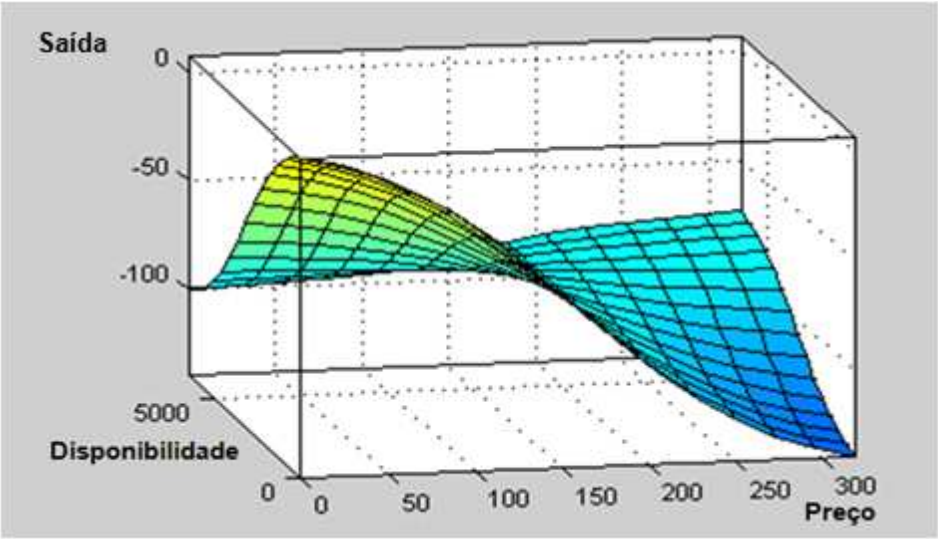

Fig. 5.3.8 Superfícies difusas a partir da combinação dos requisitos Disponibilidade e Preço

A combinação dos quatro requisitos de QoS forma uma região de decisão difusa de quatro dimensões, uma vez que os atributos são linearmente independentes. O valor de saída, após a "defuzzificação" de valores na etapa (7), permite a comparação com os valores dos parâmetros definidos pelo USLA.

Esta comparação realizada na etapa (8), permite o ajuste das funções de pertinência de saída para avaliar o grau de aderência dos requisitos de QoS do serviço selecionado ao USLA.

5.4 Comparação dos resultados obtidos com os esperados

A tabela 5.4.1 permite comparar os resultados esperados, segundo a arquitetura proposta para a Internet de Serviços (NEXOF-RA-3,2010), com os resultados obtidos através da proposta para prova de conceito (PoC).

A acuracidade apresentou um valor alto, como esperado, uma vez que o erro de verificação é inferior 0,19 em cada 100 amostras, isto é, 0,2% após 500 interações.

O desempenho apresentou um valor baixo, como esperado, uma vez que o tempo de resposta médio foi de cinco minutos, utilizando-se um computador com processador Intel de dois núcleos de 2 GHz e 4 GBytes de memória. Este tempo depende dos seguintes fatores:

- Número de épocas utilizadas para treinamento não supervisionado do algoritmo SOM e supervisionado dos algoritmos ADAPTREE e ANFIS.

- Capacidade computacional dos dispositivos que executam os algoritmos de seleção citados anteriormente.
- Acuracidade do motor de inferência difusa, após o treinamento do algoritmo ANFIS. Quanto maior a acuracidade, menor o desempenho esperado.

Tabela 5.4.1 Comparação dos resultados obtidos com os esperados

Requisitos não funcionais	Critérios de avaliação	Resultados esperados	Resultados obtidos
Capacidade de integração	Critério subjetivo, tratando-se de um protótipo	Alta	Não se aplica
Capacidade de interoperação	Critério subjetivo, tratando-se de um protótipo	Alta	Não se aplica
Acuracidade	Número de casos classificados com sucesso em relação ao total de casos avaliados	Alta	Alta
Desempenho	Tempo de resposta desde a requisição até a proposta de um ou mais serviços	Baixo	Baixo
Escalabilidade	Relação entre a capacidade computacional do sistema e o tamanho do sistema para atender uma maior demanda futura	Alta	Alta
Segurança	Não será avaliado na PoC	Alta	Não se aplica

A escalabilidade obtida através da PoC apresenta um valor alto, uma vez que não é necessário aumentar a capacidade computacional do sistema em relação ao seu tamanho, isto é, o número de linhas de código, devido a capacidade de adaptação dos algoritmos de seleção utilizados.

5.5 Comentários finais sobre a implementação do protótipo

Através da implementação do protótipo utilizado neste estudo de caso, conclui-se que os atributos de qualidade dos serviços utilizados, permitiram classificar os mesmos sob o ponto de vista não funcional, complementando a classificação funcional, segundo critérios de

similaridade funcional. A tabela 5.5.1 permite visualizar a comparação entre a classificação funcional e não funcional, segundo os domínios de aplicação dos 578 serviços avaliados.

Tabela 5.5.1 Classificação de serviços segundo critérios funcionais e não funcionais

Classificação Funcional			Classificação não funcional			
Domínio Funcional	**Quantidade de Serviços**	**Quantidade de Classes**	**Classe #1**	**Classe #2**	**Classe #3**	**Classe #4**
Educação	135	6	122	6	4	3
Cuidados com Saúde	52	1	49	1	2	1
Alimentação	25	1	23	1	1	0
Viagem	106	6	95	6	3	2
Comunicação	29	2	25	2	1	1
Economia	206	11	185	11	6	4
Armamentos	25	1	23	1	1	0
Total Absoluto	**578**	**28**	**521**	**28**	**18**	**11**
Total Percentual	**100,0%**	**100,0%**	**90%**	**5%**	**3%**	**2%**

A interpretação das quatro classes, sob o ponto vista não funcional, é descrita a seguir, com base nas regras ilustras na figura 5.3.4.

- ***Classe #1 – Serviços regulares***

  Representa serviços regulares, isto é, serviços cuja disponibilidade é indefinida, isto é alta ou baixa, porém possuem baixa confiabilidade, apesar do baixo preço e baixo atraso na execução destes serviços, conforme regras #3 e #4. Constituem a grande maioria, ou seja, 90% dos 578 serviços avaliados, estando presente em todos os domínios de aplicação.

- ***Classe #2 – Serviços de alto preço***

  Representa serviços de alto preço, conforme regra #1, na qual os demais requisitos de qualidade são indefinidos. Constituem 5% dos serviços avaliados.

- ***Classe #3 – Serviços não preferenciais***

  Representa serviços de baixa qualidade, isto é, alto atraso na execução, baixa disponibilidade, e confiabilidade indefinida, embora de baixo preço, conforme regra #3. Constituem 3% dos 578 serviços avaliados.

- ***Classe #4 – Serviços preferenciais***

  Representam serviços de alta qualidade, isto é, serviços de alta disponibilidade, alta confiabilidade e baixo preço, embora o atraso na execução seja indefinido, conforme regra # 4. Constituem apenas 2% dos 578 serviços avaliados.

Baseado no significa de cada classe, o ator Consumidor de Serviços optará preferencialmente pela seleção dos serviços classificados como Classe #4, embora representem apenas 2% do total de serviços disponíveis, são os serviços que melhor satisfazem três requisitos de QoS que constituem o USLA. Eventualmente, optar-se-á pela seleção de serviços da Classe #1, caso o requisito de confiabilidade possa ser negligenciado pelo usuário final, em detrimento do nível de serviço acordado.

A seleção de serviços da Classe #3 representa os serviços que não atendem três requisitos de QoS, definidos no USLA, e portanto não fazem parte da escolha preferencial do Consumidor de Serviços.

A seleção de serviços da Classe #2 representa uma incógnita para o Consumidor de Serviços, uma vez que, apesar do alto preço, não permite definir os demais requisitos de QoS. Possivelmente a escolha da Classe #2 dependerá da seleção de serviços da Classe #4 e da Classe #1, nesta ordem.

6. CONSIDERAÇÕES FINAIS

Este capítulo tem por objetivo apresentar as conclusões e considerações finais da tese, destacando a relevância do tema proposto, sua contribuição e ineditismo, bem com a proposta de continuidade e futuras pesquisas.

O capítulo é composto de três seções, descritas a seguir:

- Seção 6.1 apresenta a conclusão e contribuições desta tese, através da revisão dos objetivos e como os mesmos foram endereçados pelo processo de seleção proposto;
- Seção 6.2 comprova o ineditismo e contribuição ao conhecimento através dos artigos publicados com base nas soluções propostas nesta tese;
- Seção 6.3 apresenta a proposta de continuidade do trabalho e futuras pesquisas.

6.1 Conclusões e contribuições

A Internet de Serviços (IoS) é um termo abrangente utilizado para descrever diversas iniciativas que definirão como serviços serão providos e consumidos através da Internet do Futuro (IoF), a qual permitirá o compartilhamento de informações em escala global, acessíveis através de serviços contextualizados, proativos e personalizados para os usuários finais.

Neste contexto, a emergência de computação orientada a serviços, em escala da Internet, impõe novos desafios tecnológicos e conseqüentemente requer novas arquiteturas de *software*, como Arquitetura de Referência da Internet do Futuro (NEXOF-RA-1,2010), resultado dos trabalhos de pesquisa de inúmeras comunidades científicas, patrocinadas pela Comunidade Européia (NESSI,2009).

Na solução de seleção adaptativa de serviços móveis, proposta nesta tese, os principais requisitos desta nova arquitetura de *software* foram atendidos, propiciando:

- **Acessibilidade e interoperabilidade**, através da proposta de encapsulamento dos algoritmos de seleção em serviços, segundo padrões da arquitetura orientada a serviços (SOA), possibilitando a utilização em diferentes interfaces de usuários, a qual operam em plataformas de computação heterogêneas, porém convergentes sob o ponto de vista de redes e dispositivos de acesso;

- **Adaptatividade e proatividade**, através do processo de seleção dinâmico e adaptativo proposto, usuários finais recebem a oferta de serviços adaptada continuamente a sua demanda, considerando as restrições impostas pela ausência de uma infra-estrutura de rede de acesso fixa, através do uso de redes móveis Ad-Hoc e sensibilidade ao contexto;
- **Aprendizado contínuo,** através da capacidade de aprendizado supervisionado e não supervisionado dos algoritmos de seleção, possibilitando a inferência de regras estritas ou difusas, tratando as incertezas na definição dos parâmetros de qualidade de serviço que gerenciam o acordo de nível de serviço entre provedores e consumidores.

Através dos resultados apresentados, conclui-se que a combinação de métodos de inteligência artificial de aprendizado não supervisionado e supervisionado, presentes nos algoritmos SOM e ANFIS respectivamente, e aprendizado adaptativo por máquina, presente no algoritmo ADAPTREE, suportam o processo de decisão desde o provisionamento do serviço até o seu consumo.

As incertezas nas definições dos valores dos atributos de requisitos não funcionais de QoS fim-a-fim, normalmente presentes em requisições de serviços em redes Ad-Hoc foram tratadas pelo algoritmo ANFIS, permitindo que a solução proposta adapte a oferta de serviços de forma interativa à requisição, buscando maior aderência ao USLA definido de forma difusa pelo usuário final.

Dessa forma, conclui-se que a combinação de métodos de inteligência artificial de aprendizado não supervisionado e supervisionado, melhorou o processo de descoberta de serviços, atendendo aos objetivos propostos nesta tese:

- Reduziu-se o universo de escolha de serviços, agrupando a oferta de 578 serviços, classificados em 28 classes sob o ponto de vista funcional, em apenas 4 classes sob o ponto de vista não funcional, organizadas e definidas através de um conjunto de 5 regras difusas;
- Estas regras permitiram o tratamento de incertezas na definição de requisitos não funcionais QoS, adaptando a oferta de serviços às requisições de consumo e conseqüentemente auxiliando provedores no cumprimento do nível de serviço acordo com consumidores;

- A descoberta das relações hierárquicas entre as quatro classes de serviços, obtidas de forma incremental e adaptativa, permitiram o aprendizado supervisionado de padrões aprendidos de forma não supervisionada, viabilizando a seleção interativa de serviços.
- A seleção interativa complementou a etapa de procura de serviços, permitindo a melhora do processo de descoberta de serviços, através da combinação entre classificações baseadas em requisitos funcionais e não funcionais.

6.2 Ineditismo e contribuição ao conhecimento

O ineditismo da solução proposta consistiu na combinação de diferentes métodos de redes neurais artificiais, árvores de decisão e lógica difusa, para a seleção de serviços de forma adaptativa e interativa com base em requisitos não funcionais de qualidade dos mesmos. Esta combinação supriu as deficiências de cada método, cujo resultado final superou a soma dos resultados individuais.

Este trabalho contribuiu para o projeto de pesquisa da Internet de Serviços, e ao conhecimento, através de publicações em congressos nacionais e internacionais, citados a seguir:

- O primeiro artigo, denominado de "*Adaptive Fuzzy Neural Tree Network" (AFNTN)* (DUTRA; MARTUCCI, 2008), foi publicada como "*full paper*" na revista IEEE América Latina, sendo apresentado no *Workshop* de Tecnologia Adaptativa (WTA) da Escola Politécnica da USP em 2008. Este artigo apresenta a combinação dos algoritmos utilizados nesta tese para a prospecção de dados.
- Adicionalmente a este artigo, foi proposta a criação de uma linguagem adaptativa para seleção de serviços utilizando a AFNTN, aceita como "*short paper*" sob o título "*Adaptive Dynamic Discovery Language*" (DUTRA; et al., 2009) e apresentado no Workshop de Tecnologia Adaptativa da Escola Politécnica da USP em 2009.
- Finalmente, a solução de serviços adaptativa e interativa de serviços móveis, foi publicada como "*full paper*" na revista Springer Verlag, sob o título "*Dynamic Adaptive Middleware Services for Service Selection in Mobile Ad-Hoc Networks*" (DUTRA; MARTUCCI, 2010), apresentado no congresso *Mobilware* 2010 em Chicago (EUA).

Esta tese contribuiu para os trabalhos de pesquisa relacionados ao tema de mobilidade e interatividade, desenvolvidos no Laboratório de Sistemas Abertos (LSA) da Poli-USP:

- KARAM (2006) propôs um modelo para um *framework* de mobilidade e interatividade aplicáveis a sistemas abertos; e um modelo de negócio voltado aos serviços num cenário com soluções tecnológicas interoperáveis. A solução proposta nesta tese melhora a seleção interativa e adaptativa de serviços, fornecendo o suporte tecnológico ao *framework* proposto.
- SERRA (2007) propôs um primeiro passo para o equacionamento da falta de métodos padronizados para a identificação de requisitos de QoS sob a percepção do usuário final e sua especificação no USLA. Esta tese complementa este trabalho, propondo uma solução para atendimento do USLA em redes móveis Ad-Hoc.
- SAKURAI (2010) propôs uma arquitetura para provisionamento de serviços, pressupondo a existência de um único provedor de serviços. Esta tese complementa este trabalho, permitindo que o provedor de serviços selecione, de forma dinâmica, serviços disponíveis em um ambiente convergente e heterogêneo.

Finalmente esta tese representa a continuidade da dissertação de mestrado (DUTRA;CABRAL,2001), na qual foi descrita a combinação dos algoritmos de rede neural SOM e árvore de decisão C4.5.

6.3 Proposta de continuidade

Conforme citado nos objetivos desta tese, os temas que não foram tratados e constituem a proposta de continuidade deste trabalho, como descrito a seguir:

A privacidade dos usuários é um tema de crescente relevância, pois o aprendizado dos padrões de consumo de serviço pode ser utilizado para a identificação dos usuários dos mesmos, gerando possíveis conflitos de invasão de privacidade. Uma possível solução para este problema implicará em um aprendizado seletivo de padrões, segundo regras de privacidade estabelecidas pelo usuário do serviço, aumentando a complexidade dos

algoritmos utilizados e possivelmente introduzirá um conflito entre a precisão e usabilidade da solução proposta.

Para avaliar o conflito entre precisão e usabilidade, deverá ser incrementado o número de atributos de QoS, considerando por exemplo os atributos de qualidade dos serviços de rede, verificando o impacto no desempenho da solução de seleção proposta.

Adicionalmente pretende-se avaliar as questões relacionadas à composição, coreografia e orquestração de serviços móveis, em ambientes heterogêneos e convergentes, nos quais a seleção de serviços poderá auxiliar na escolha da melhor composição de um conjunto complementar de serviços, visando facilitar a coreografia e a orquestração dos mesmos, de forma dinâmica e adaptativa.

Novos estudos de caso serão avaliados, objetivando testar a solução em diferentes aplicações de *Computação em Nuvem* (*cloud computing*), tais como:

- IaaS (*Infrastructure as a Service)* Infra-estrutura como Serviço: quando se utiliza uma porcentagem de um servidor, geralmente com configuração que se adeque à sua necessidade;
- PaaS (*Plataform as a Service*) Plataforma como Serviço: utilizando-se apenas uma plataforma como um banco de dados, um *web service*, por exemplo;
- DaaS (*Development as a Service*) Desenvolvimento como Serviço: as ferramentas de desenvolvimento baseadas na Internet e serviços baseados em *mashup*;
- SaaS (*Software as a Service*) *Software* como Serviço: uso de um *software* em regime de utilização na Internet, por exemplo : *Google Docs* ® , *Microsoft Sharepoint Online* ®;
- CaaS (*Communication as a Service*) Comunicação como Serviço: uso de uma solução de comunicação unificada hospedada em centro de dados de provedor es.

Adicionalmente as aplicações de *Computação em Nuvem*, será avaliado a aplicação da proposta de seleção de serviços em redes móveis Ad-Hoc, possibilitando a descoberta de Serviços Baseados em Localização (*Location Based Services*), na qual serviços serão oferecidos aos usuários com base na posição ou localização onde o mesmo se encontra.

REFERÊNCIAS BIBLIOGRÁFICAS

(AGASSI,et al.,2010) AGASSI,et al.; **Internet of Services. NESSI Open Framework – Reference Architecture**. IST- FP7-216446.2010.

(ALRIFAI;et. al,2009) ALRIFAI, M.;et al;**A Scalable Approach for QoS-based Web Service Selection**;L3S Research Center Leibniz University of Hannover, Germany; 2009

(BOOTH;et al,2007) BOOTH, D.; et al.; **Web Services Description Language (WSDL) Version 2.0 Part 0: Primer**.W3C Recommendation. Última versão disponível em http://www.w3.org/TR/wsdl20-primer. Último acesso em Janeiro, 2010.

(BOOTH;et al.,2004) BOOTH, D.; et al.;**Web Services Architecture**; W3C Working Group Note 11. 2004. Disponível em http://www.w3.org/TR/ws-arch/ . Ultimo acesso: Julho, 2010.

(BOTAZZI;et al.,2007) BOTTAZI, D., MONTANARI, R., TONINELLI, A. "**Context-Aware *Middleware* for Anytime, Anywhere Social Networks**." IEEE Intelligent Systems, 22(5), 23-32.2007.

(BOTTAZZI;et al.,2008) BOTTAZI, D., MONTANARI; GIOVANNI, R.," **A self-organizing group management *middleware* for mobile ad-hoc networks**". Computer Communications 31 3040–3048. 2008.

(BRODLEY;UTGOFF,1992) BRODLEY, C. E.; UTGOFF, P. E. **Multivariate versus univariate decision trees**. Technical report, Department of Computer Sciences University of Massachussetts. 1992.

(CANFORA,2005) CANFORA G, et. At.; A**n approach for QoS-aware service composition based on genetic algorithms.** Proc of the 2005 conference on Genetic and Evolutionary Computation (GECCO), Washington DC, USA. pp. 1069–1075

(CHENG;et al.,1988) CHENG, J.; FAYYAD, U. M.; IRANI, K. B.; QIAN, Z.; **Improved decision trees: A generalized version of ID3. Proceedings of the Fifth International Conference on Machine Learning** (pp. 100-106). Ann Arbor, MI: Morgan Kaufman, 1988.

(CLEMENT;et al.,2007) CLEMENT, L.; et al; **UDDI Version 3.0.2**, Tech. Rep., OASIS, 2004, http://uddi.org/pubs/uddi-v3.0.2-20041019.htm. Last access in January, 2010.

(CONNER, 2010) CONNER, M. **Sensors empower the "Internet of Things"**. pp. 32–38. ISSN 0012-7515.

(CORBA,2007) CORBA.OMG. **Common Object Request Broker Architecture.** Disponível em: http://www.corba.org. Último acesso: Agosto, 2007.

(CUGOLA;NITTO,2008) CUGOLA, G., NITTO, E. D.," **On adopting Content-Based Routing in service-oriented architectures**". Information and *Software* Technology 50,22–35.2008.

(DAMLS,2010) DAMLS. **DAML-S Home Page.** http://www.daml.org/services/owl-s/ Último acesso: Agosto, 2010.

(DAVIES;BOULDIN,1979) DAVIES, D. L.; BOULDIN, D.W. **A Cluster Separation Measure**. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. PAMI-1, no. 2, pp. 224-277. 1979.

(DCOM,2007) **MICROSOFT DCOM**. Disponível em: http:/www.microsoft.com/com/tech/ dcom.asp Último acesso: Agosto, 2007.

(DUTRA;CABRAL,2001) DUTRA, R. ; CABRAL, E. **Aplicação de métodos de inteligência artificial em inteligência de negócios**. Dissertação de Mestrado, Poli USP, 2001.

(DUTRA;et. al,2009) DUTRA, R. G. **et al. Adaptive Dynamic Discovery Language**. [S.l.]. 2009. Short Paper. III Workshop Tecnologia Adaptativa, Poli-USP.

(DUTRA;MARTUCCI,2008) DUTRA, R. G.; MARTUCCI, M. **Adaptive Fuzzy Neural Tree Network**. [S.l.]. 2008. II. Workshop de Tecnologia Adaptativa. Revista IEEE R9 América Latina.

(DUTRA;MARTUCCI,2010) DUTRA, R. G.; MARTUCCI, M. **Dynamic Adaptive *Middleware* Services for Service Selection in Mobile Ad-Hoc Networks**. [S.l.], p. 14. 2010. Apresentado no Congresso Mobilware 2010. Chicago. A ser publicado na Springer Verlag.

(EFII,2010) EFII. **The EUROPEAN FUTURE INTERNET INITIATIVE White paper on the Future Internet PPP Definition**. [S.l.], p. 14. 2010.

(FUZZY, 2011) MATLAB. **Fuzzy Toolbox**. Disponível em http://www.mathworks.com/ help/toolbox/fuzzy/fp715dup12.html. Último acesso: Janeiro, 2011.

(GAMAL;et al.,2004) GAMAL,A., et al.; **Throughput delay trade-off in wireless networks**. Proc of the IEEE Infocom1:464–475.2004

(GEIHS,2004) GEIHS, K. ***Middleware* Challenges Ahead**. Computer, Junho 2001. p. 24-30.

(GU;et al.,2006) GU, X., et al.; **Distributed multimedia service composition with statistical QoS assurances**. IEEE Trans Multimedia 8(1):141–151

(GUDGIN;et al.,2007) GUDGIN, M., et al. **SOAP. Simple Object Access Protocol (SOAP) 1.2.** Disponível em: http://www.w3.org/TR/soap12-part1/. Último acesso: Agosto, 2010.

(GUPTA;et al.,2009) GUPTA A.;KALRA, A; BOSTON, D., BORCEA, C., "**MobiSoC: A *Middleware* for Mobile Social Computing Applications**". Mobilware Congress, Berlin. 2009.

(IEEE1471,2007) IEEE1471; ISO/IEC. **Systems and *Software* Engineering - Recommended Practice for Architectural Description of *Software*-Intensive Systems.** [S.l.], p. ISO/IEC 42010 IEEE Std 1471-2000 First edition 2007-07-15, c1–24. 2007.

(JANG, 1993) JANG, S-J. R;**Adaptive Network-base Fuzzy Inference System**. IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no. 3, May/June 1993.

(JONES;et al.,2004) JONES, Q.; et al.;**People-to-People-to-Geographical-Places: The P3 Framework for Location-Based Community Systems.** Computer Supported Cooperative Work 13: 249–282, 2004.

(KARAM,2006) KARAM JR., D. **Modelo de negócio para mobilidade e interatividade em ambientes convergentes heterogêneos**. [S.l.]. 2006. 80 p. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo, 2006.

(KICZALES,1997) KICZALES, G.; et al.;**Aspect-oriented programming**. In Proceedings of the European Conference on Object-Oriented Programming (ECOOP), Springer-Verlag LNCS 1241, 1997.

(KOHONEN,1995) KOHONEN, T. **Self-Organizing Maps**. Springer-Verlag. 1995.

(LENDERS;et.al.,2005) LENDERS,V.;et.al.,**Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach**. World of Wireless Mobile and Multimedia Networks (WoWMoM 05), IEEE Press, vol.1, pp. 120–130.2005.

(MACKENZIE;et al.,2006) MACKENZIE, C.M. et al, **Reference Model for Service Oriented Architecture 1.0**. OASIS Standard. Outubro, 2006. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf. Último acesso

em Fevereiro de 2010.

(MAES,1987) MAES, P.; **Concepts and experiments in computational reflection**.In Proceedings of the ACM Conference on Object-Oriented Languages (OOPSLA), 1987.

(MAMDANI; ASSILIAN,1975) MAMDAMI, E.H. ; ASSILIAN, S.; "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.

(MATLAB,2010) MATLAB. **Wiress Network Simulation.** Free *software* available in http://wireless-matlab.sourceforge.net/. Last access in January, 2010.

(MIAN;et al.,2009) MIAN, A.; BALDONI, R.; BERALDI, R. **A Survey of Service Discovery Protocols in Multihop Mobile Ad Hoc Networks**. University of Rome La Sapienza. [S.l.], p. 9. 2009.

(NESSI,2009) NESSI. NETWORKED EUROPEAN *SOFTWARE* & SERVICES INITIATIVE **Research Priorities for FP7: A NESSI Roadmap series document**. [S.l.]. 2009. NESSI. VOL 2.

(O'REILLY,2009) O'REILLY, T. **http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.**. Último acesso Fevereiro. [S.l.]. 2009.

(PAPADIMITRIOU; et al.,2004) PAPADIMITRIOU, D.; et al.; **Elementos da teoria da computação.** Bookmann, 2a. edição, 2004.

(PAPADIMITRIOU; et al.,2009) PAPADIMITRIOU, D.; et al.; **Future Internet. The cross-ETP vision document**. Version 1.0. Nessi Project resources. 2009.

(PISTORI;et al.,2006) PISTORI, H.; NETO, J. J.; PEREIRA, M. C. **Adaptive Non-Deterministic Decision Trees: General Formulation and Case Study**. INFOCOMP Journal of Computer Science, Lavras, MG, 2006 (accepted).

(PISTORI;NETO,2002) PISTORI, H.; NETO, J. J. ADAPTREE **- Proposta de um Algoritmo para Indução de Árvores de Decisão Baseado em Técnicas Adaptativas.** Anais Conferência Latino Americana de Informática - CLEI 2002. Montevideo, Uruguai, Novembro, 2002

(QUINLAN,1990) QUINLAN, J. R.; **Comparing Connectionist and Symbolic Learning Methods. Basser Department of Computer Science**; University of Sydney; Sydney NSW 2006;Australia. 1990.

(QUINLAN,1992) QUINLAN, J. R. **C4.5 Programs for Machine Learning**. Morgan Kaufmann. 1992.

(REID;et al.,2005) REID,B.;et al., **A Cross Layer Approach to Service Discovery and Selection in Manets.** Proc. Int'l Conf. Mobile Ad-Hoc and Sensor Systems (MASS 05), IEEE Press, 2005.

(RMI,2007) RMI. **Java Remote Method Invocation**. Disponível em: http://java.sun.com/products/jdk/rmi/. Último acesso: Agosto, 2007.

(RM-ODP,1998) ISO/IEC 10746-1, 2, 3, 4 | ITU-T Recommendation X.901, X.902, X.903, X.904. "Open Distributed Processing - Reference Model". OMG, 1995.-98. Disponível em http://isotc.iso.ch/livelink/livelink/fetch/2000/2489/ Ittf_Home/PubliclyAvailableStandards.htm. Último acesso em Dezembro,2009.

(SADJADI, 2009) SADJAD, S. M. **A Survey of Adaptive *Middleware***. Disponível em http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.9321&rep=rep1&type=pdf. Último acesso Dezembro, 2009.

(SAKURAI,2010) SAKURAI, C. A. **Proposta de uma arquitetura para o fornecimento de serviços móveis em redes convergentes heterogêneas com foco no usuário**. USP. São Paulo, p. 118. 2010.

(SAP SDN,2010) SAP Developer Network. E**nterprise Services Repository.** Disponível em sdn.sap.com/soa. Último acesso: Janeiro.2010.

(SCHMIDT,2001) SCHMIDT,D.; et al.; **Pattern-Oriented *Software* Architecture**, vol. 2. John Wiley, 2001.

(SCHULTE;NATIS,1996) SHULTE, W.R., NATIS, Y. V. **SSA *Research Note* SPA-401-068, , "'Service Oriented' Architectures, Part 1" e SSA *Research Note* SPA-401-069, "Service Oriented' Architectures, Part 2".** 1996.

(SERRA,2007) SERRA, A. P. G.; **Método para Identificação de Parâmetros de Qualidade de Serviços Aplicados a Serviços Móveis e Interativos**. [S.l.]. 2007. 136p. Tese (Doutorado em Engenharia) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2007.

(SUGENO,1977) Sugeno, M., **Fuzzy measures and fuzzy integrals: a survey**.pp. 89-102, North-Holland, NY, 1977.

(SUGENO,1985) SUGENO, M., **Industrial applications of fuzzy control**, Elsevier Science Pub. Co., 1985.

(SZYPERSKI,1999) SZYPERSKI, C.; **Component *Software*: Beyond Object-Oriented Programming.** Addison-Wesley, 1999.

(TSELENTIS;et. al,2009) TSELENTIS, G.;et al. **Towards the Internet of Future. An European Research Perspective**. IOS press. 2009.

(TYLAN;et al.,2004) TYLAN, J. ;et al., **A Network Layer Based Architecture for Service Discovery in Mobile Ad Hoc Networks**. Proc. IEEE Canadian Conf. Electrical and Computer Eng. (CCECE 04), IEEE Press, vol. 3, pp.1379–1384.2004.

(VENKATA,2002) VENKATASUBRAMANIAN, N., **Safe Composability of *Middlewares* Systems**. Communications ACM 45, 6, p 49-52. 2002.

(VESANTO;et al.,1999) VESANTO, J.; ALHONIEMI, E.; HIMBERG, J.; PARHANKANGAS, J. **Som Toolbox 2.0 BETA online documentation.** Internet address http:// www.cis.hut.fi/projects/ somtoolbox. 1999.

(HERNAN,2001) HERNAN, I. .**People will never get common agreement on terms; won't this lead to the failure of the Semantic Web? Semantic Web FAQ.** Disponível em http://www.w3.org/2001/sw/ SW-FAQ#What1. Item 4.17. 2001. Último acesso Maio, 2010.

(W3C,2010) **World Wide Web Consortium (W3C)**. Disponível em http://www.w3.org/. Último acesso Maio, 2010.

(WEKA,2010) WEKA. **Toolbox for Datamining**.Software Livre disponível em http://www.cs.waikato.ac.nz/ml/weka. Último acesso: 2010.

(YANG;et al.,2009) YANG, K. et al.**QoS-Aware Service Selection Algorithms for Pervasive Service Composition in Mobile Wireless Environments**. Mobile Netw Appl, Springer, 2009.

(YU;et al.,2007) YU, T; et al.; **Service selection algorithms for Web services with end-to-end QoS constraints**. Proc of IEEE Int Conf on e-Commerce Technology, July, pp 129–136.2007.

(ZADEH,1965) ZADEH, L.A., "Fuzzy sets," Information and Control, Vol. 8, pp. 338-353, 1965.

(ZAHARIADIS; et al.,2011) ZAHARIADIS T. et al., **Towards a Future Internet Architecture.** LNCS 6656, pp. 7–18, 2011

(ZENG;et al.,2004) ZENG, K. et al. QoS-aware middleware for Web services composition.IEEE Trans Softw Eng 30(5):311–327.2004.

(ZHU;et al.,2003) ZHU,F.;et al., **A Secure, Private and Location-Aware Service Discovery Protocol Supporting Mobile Services**. Pervasive Computing and Comm. (Per-Com 03), ACM , pp. 235-242.2003

ANEXO – Serviços utilizados na prova de conceito

O conjunto de 578 serviços da base de serviços denominada OWL-S versão 2 (DAMLS, 2011) é composto de 28 consultas, cada uma está associada a um conjunto contendo 10 a 15 serviços. Cada serviço está associado aos atributos de QoS normalizados, descritos no capítulo 5, combinando-se a base OWL-S versão 2 com atributos não funcionais extraídos de (SAP SDN,2010) e apresentados a seguir, observando-se as seguintes abreviações:

- Disponibilidade (Qav)
- Confiabilidade (Qre)
- Atraso na Execução (Qde)
- Preço (Qpr)

Para avaliação da característica Preço, considerou-se o serviço com o custo de consulta mais elevado receberá a classificação “Alto” e todos os demais serviços dentro de uma mesma consulta receberão a classificação “Baixo”.

Consulta 1: economy/ car_price_service.owls

Esta consulta permite ao consumidor consultar o preço de automóveis. O consumidor não deseja alugar um automóvel.

Nome do Serviço	Qav	Qre	Qde	Preço
car_price_service.owls	0.98	1.0	1.0	Baixo
car_priceauto_service.owls	0.64	0.51	0.61	Baixo
car_pricecolor_service.owls	0.57	0.55	0.59	Baixo
car_pricequality_service.owls	0.92	0.89	0.93	Baixo
car_pricereport_service.owls	0.67	0.57	0.67	Baixo
car_taxedpriceprice_service.owls	0.95	0.85	0.94	Baixo
_RedFerrariprice_service.owls	0.5	0.5	0.5	Baixo
_Toyotaprice_service.owls	0.5	0.5	0.5	Baixo
car_recommendedprice_service.owls	0.94	0.89	0.94	Baixo
auto_price_service.owls	0.98	0.99	1.0	Baixo
auto_yearprice_service.owls	0.82	0.72	0.81	Baixo
car_recommendedpriceindollar_service.owls	0.92	0.84	0.9	Baixo
car_recommendedpriceineuro_service.owls	0.92	0.89	0.93	Baixo
4wheeledcar_price_service.owls	0.97	0.94	0.97	Baixo
cheapcar_price_service.owls	0.97	0.94	0.97	Baixo
expensivecar_price_service.owls	0.98	0.99	1.0	Baixo
fastcar_recommendedprice_service.owls	0.95	0.93	0.96	Baixo
vehicle_price_service.owls	0.64	0.58	0.68	Baixo
amount-of-moneycar_price_service.owls	0.91	0.87	0.93	Baixo
amount-of-moneycar_pricecompany_service.owls	0.56	0.38	0.53	Baixo
car_report_service.owls	0.48	0.5	0.47	Alto

Consulta 2: economy/ book_price_service.owls

Esta consulta informa o preço de livros, excluindo jornais e revistas.

Nome do Serviço	Qav	Qre	Qde	Preço
book_price_service.owls	0.99	1.0	1.0	Baixo
BookPrice.owls	0.99	1.0	1.0	Baixo
book_Cheapestprice_service.owls	0.99	1.0	1.0	Baixo
book_authorprice_Novelservice.owls	0.93	0.89	0.93	Baixo
book_authorprice_service.owls	0.93	0.89	0.93	Baixo
book_reviewprice_service.owls	0.96	0.94	0.96	Baixo
book_taxedpriceprice_service.owls	0.96	0.86	0.96	Baixo
book_pricereviewbook_service.owls	0.72	0.61	0.7	Baixo
book_pricesizebook-type_service.owls	0.93	0.89	0.93	Baixo
book_recommendedprice_service.owls	0.96	0.94	0.96	Baixo
monograph_price_service.owls	0.86	0.8	0.88	Baixo
book_recommendedpriceindollar_service.owls	0.93	0.89	0.93	Baixo
Tizonbook_recommendedpriceindollar_service.owls	0.93	0.89	0.93	Baixo
monograph_recommendedpriceineuro_service.owls	0.8	0.68	0.81	Baixo
bookpersonOptional_price_service.owls	0.99	1.0	1.0	Baixo
author_bookprice_service.owls	0.41	0.17	0.34	Baixo
title_pricebook_service.owls	0.41	0.17	0.35	Alto

Consulta 3: economy/ dvdplayermp3player_price_service.owls

Esta consulta informa o preço de tocadores MP3 e DVDs.

Nome do Serviço	Qav	Qre	Qde	Preço
dvdplayermp3player_price_service.owls	0.94	0.98	0.99	Baixo
dvdplayermp3player_price_Rservice.owls	0.92	0.94	0.97	Baixo
dvdplayermp3player_price_MDservice.owls	0.97	1.0	1.0	Baixo
dvdplayermp3player_pricemessage_service.owls	0.69	0.58	0.73	Baixo
mp3playerdvdplayer_priceshipping_service.owls	0.8	0.72	0.81	Baixo
mp3playerdvdplayer_recommendedprice_service.owls	0.91	0.91	0.95	Baixo
dvdplayermp3player_Recprice_service.owls	0.93	0.94	0.96	Baixo
mp3playerdvdplayer_Recpriceshipping_service.owls	0.76	0.67	0.79	Baixo
dvdplayermp3player_Recpricemessage_service.owls	0.7	0.61	0.73	Baixo
dvdplayermp3player_RecpriceEuro_service.owls	0.88	0.86	0.92	Baixo
mp3playerdvdplayer_Recpriceshipping_USservice.owls	0.75	0.66	0.77	Baixo
mp3playerportabledvdplayer_price_service.owls	0.96	0.98	0.99	Baixo
mp3playerportabledvdplayer_recommendedpricequality_service.owls	0.88	0.83	0.89	Baixo
portabledvdplayermp3player_recommendedpricetaxedprice_service.owls	0.9	0.79	0.92	Alto

Consulta 4: economy/ 1personbicyclecar_price_service.owls

Esta consulta informa o preço de automóveis e bicicletas.

Nome do Serviço	Qav	Qre	Qde	Preço
1personbicyclecar_price_service.owls	0.98	1.0	1.0	Baixo
1personbicyclecar_price_Kohlservice.owls	0.97	0.96	0.98	Baixo
1personbicyclecar_price_TheBestservice.owls	0.97	1.0	1.0	Baixo
auto1personbicycle_price_service.owls	0.97	0.96	0.97	Baixo
bicyclecar_priceyear_service.owls	0.81	0.72	0.81	Baixo
carcycle_price_service.owls	0.95	0.95	0.97	Baixo
carbicycle_recommendedprice_service.owls	0.93	0.89	0.94	Baixo
cyclecar_pricetaxedprice_service.owls	0.93	0.84	0.95	Baixo
cycle1personbicycle_price_service.owls	0.71	0.63	0.75	Baixo
cyclecar_recommendedpriceineuro_service.owls	0.9	0.87	0.92	Baixo
1personbicycle4wheeledcar_price_service.owls	0.97	0.96	0.97	Baixo
4wheeledcarbicycle_price_service.owls	0.96	0.99	0.99	Alto

Consulta 5: economy/bookpersoncreditcardaccount_price_service.owls

Esta consulta informa o preço de um livro, verificando a conta de crédito pessoal do consumidor.

Nome do Serviço	Qav	Qre	Qde	Preço
bookpersoncreditcardaccount_price_service.owls	0.97	1.0	1.0	Baixo
bookperson_price_service.owls	0.89	0.93	0.96	Baixo
book_price_service.owls	0.89	0.93	0.96	Baixo
BookPrice.owls	0.89	0.93	0.96	Baixo
book_Cheapestprice_service.owls	0.89	0.93	0.96	Baixo
bookpersonOptional_price_service.owls	0.89	0.93	0.96	Baixo
book_pricereviewbook_service.owls	0.62	0.54	0.66	Baixo
book_pricesizebook-type_service.owls	0.83	0.81	0.89	Baixo
book_reviewprice_service.owls	0.86	0.86	0.92	Baixo
book_taxedpriceprice_service.owls	0.86	0.78	0.92	Baixo
bookpersoncreditcardaccount_taxedfreeprice_service.owls	0.94	0.94	0.96	Baixo
bookpersoncreditcardaccount_recommendedprice_service.owls	0.94	0.94	0.96	Baixo
monographpersoncreditcardaccount_recommendedprice_service.owls	0.87	0.78	0.87	Baixo
book_recommendedprice_RegisteredUserservice.owls	0.86	0.86	0.92	Baixo
book_recommendedprice_service.owls	0.86	0.86	0.92	Baixo
monograph_price_service.owls	0.79	0.75	0.85	Baixo
printedmaterialpersoncreditcardaccount_price_service.owls	0.82	0.68	0.79	Baixo
printedmaterial_price_service.owls	0.61	0.57	0.66	Baixo
book_recommendedpriceindollar_service.owls	0.83	0.81	0.89	Baixo
Tizonbook_recommendedpriceindollar_service.owls	0.83	0.81	0.89	Baixo
monograph_recommendedpriceineuro_service.owls	0.73	0.64	0.78	Alto

Consulta 6: economy/ maxprice_cola_service.owls

Esta consulta informa o preço de um refrigerante.

Nome do Serviço	Qav	Qre	Qde	Preço
maxprice_cola_service.owls	0.99	1.0	1.0	Baixo
maxprice_cola_Bestservice.owls	0.99	1.0	1.0	Baixo
maxprice_beercola_service.owls	0.92	0.86	0.94	Baixo
maxprice_colabreadorbiscuit_Bothservice.owls	0.77	0.65	0.75	Baixo
maxprice_colabeer_service.owls	0.89	0.8	0.9	Baixo
price_cola_Gudduservice.owls	0.93	0.94	0.96	Baixo
price_cola_Halloservice.owls	0.93	0.94	0.96	Baixo
price_cola_Hallo2service.owls	0.93	0.94	0.96	Baixo
maxprice_whiskeycolabeer_service.owls	0.89	0.75	0.89	Baixo
maxprice_drinks_service.owls	0.86	0.83	0.9	Baixo
taxfreeprice_cola_service.owls	0.91	0.89	0.93	Baixo
qualitymaxprice_cola_service.owls	0.94	0.9	0.94	Alto

Consulta 7: economy/ bookpersoncreditcardaccount__service.owls

Esta consulta permite verificar o conteúdo de um carrinho de compras de livros e a conta de crédito do consumidor.

Nome do Serviço	Qav	Qre	Qde	Preço
bookpersoncreditcardaccount__service.owls	0.95	1.0	1.0	Baixo
bookpersoncreditcardaccount__BShopservice.owls	0.95	1.0	1.0	Baixo
bookperson__service.owls	0.78	0.86	0.92	Baixo
bookpersoncreditaccount__service.owls	0.9	0.97	0.98	Baixo
bookpersoncreditaccount__Beaservice.owls	0.9	0.97	0.98	Baixo
personmonographcreditcardaccount__service.owls	0.8	0.68	0.81	Baixo
monographperson__service.owls	0.59	0.54	0.71	Baixo
printedmaterialpersoncreditcardaccount__service.owls	0.63	0.36	0.58	Baixo
personbookliabilityaccount__service.owls	0.86	0.93	0.96	Baixo
printedmaterialperson__service.owls	0.31	0.22	0.43	Baixo
publicationperson__service.owls	0.5	0.44	0.63	Baixo
bookusercreditcardaccount__service.owls	0.93	0.97	0.98	Alto

Consulta 8: economy/ shoppingmall_cameraprice_service.owls

Esta consulta permite verificar o preço de cameras fotográficas digitais em um centro de compras.

Nome do Serviço	Qav	Qre	Qde	Preço
shoppingmall_cameraprice_service.owls	0.91	1.0	1.0	Baixo
shoppingmall_calendar-datepricecamera_service.owls	0.7	0.64	0.76	Baixo
_pricecamera_Wallmartservice.owls	0.45	0.5	0.5	Baixo
shoppingmall_pricedigitalanalog_service.owls	0.69	0.61	0.77	Baixo
shoppingmall_analogpricecalendar-date_service.owls	0.62	0.59	0.7	Baixo
_digitalstandardpriceprice_MediaMarktservice.owls	0.27	0.22	0.35	Baixo
SRcamera_service.owls	0.86	0.91	0.95	Baixo
shoppingmall_maxpricedigital-video_service.owls	0.69	0.66	0.76	Baixo
shoppingmall_digital-slrpricecalendar-date_service.owls	0.59	0.6	0.7	Baixo
shoppingmall_purchaseableitemprice_service.owls	0.82	0.94	0.97	Baixo
PhillipDigCamera_price_service.owls	0.72	0.87	0.92	Baixo
shoppingmall_pricecellphonewithcamera_service.owls	0.69	0.66	0.76	Baixo
KodakDigCamera_price_service.owls	0.29	0.39	0.43	Baixo
_price_CannonCameraservice.owls	0.29	0.39	0.43	Baixo
_price_CannonCameraservice.owls	0.29	0.39	0.43	Alto

Consulta 9: economy/recommendedprice_coffeewhiskey_service.owls

Esta consulta informa o preço de bebidas como café e whisky.

Nome do Serviço	Qav	Qre	Qde	Preço
recommendedprice_coffeewhiskey_service.owls	0.99	1.0	1.0	Baixo
recommendedprice_coffeewhiskey_Bestservice.owls	0.99	1.0	1.0	Baixo
recommendedprice_contentbearingobjectwhiskeycoffee_service.owls	0.87	0.78	0.84	Baixo
recommendedprice_coffeewhiskeysymbolicstring_service.owls	0.86	0.76	0.83	Baixo
recommendedprice_coffeewhiskeysymbolicstringquality_service.owls	0.83	0.73	0.8	Baixo
price_whiskeycoffee_service.owls	0.94	0.94	0.96	Baixo
price_coffeewhiskey_service.owls	0.94	0.94	0.96	Baixo
price_whiskeycoffee_Zikoservice.owls	0.94	0.94	0.96	Baixo
price_coffeewhiskey_Thebestservice.owls	0.94	0.94	0.96	Baixo
price_coffeewhiskeytimemeasure_service.owls	0.83	0.74	0.82	Baixo
price_coffeewhiskeyquality_service.owls	0.88	0.83	0.89	Baixo
price_coffeewhiskeyqualitytimeposition_service.owls	0.79	0.68	0.78	Baixo
hotelrecommendedprice_coffeewhiskey_service.owls	0.94	0.9	0.94	Baixo
maxprice_coffeewhiskey_service.owls	0.91	0.89	0.93	Baixo
recommendedpriceindollar_whiskeycoffee_service.owls	0.97	0.94	0.97	Baixo
recommendedpriceineuro_coffeewhiskey_service.owls	0.97	0.94	0.97	Baixo
taxfreeprice_whiskeycoffee_service.owls	0.91	0.89	0.93	Alto

Consulta 10: economy/ userscience-fiction-novel_price_service.owls

Esta consulta fornece o preço de um livro de ficção científica.

Nome do Serviço	Qav	Qre	Qde	Preço
userscience-fiction-novel_price_service.owls	0.98	1.0	1.0	Baixo
userscience-fiction-novel_Relprice_service.owls	0.98	1.0	1.0	Baixo
DeoSFN_price_service.owls	0.98	1.0	1.0	Baixo
userscience-fiction-novel_price_Bestservice.owls	0.98	1.0	1.0	Baixo
sciencefictionbookuser_price_service.owls	0.93	0.86	0.96	Baixo
SFNRecommendedPrice_service.owls	0.95	0.94	0.96	Baixo
novelperson_price_service.owls	0.92	0.86	0.97	Baixo
novel_price_service.owls	0.93	0.86	0.97	Baixo
ScienceFNovelReview_service.owls	0.92	0.89	0.93	Baixo
userbook_price_service.owls	0.92	0.85	0.96	Baixo
science-fiction-noveluser_recommendedpriceindollar_service.owls	0.92	0.89	0.93	Baixo
SFNovelReview_service.owls	0.9	0.85	0.9	Alto

Consulta 11: economy/ preparedfood_price_service.owls

Esta consulta informa o preço de comida pronta.

Nome do Serviço	Qav	Qre	Qde	Preço
preparedfood_price_service.owls	0.95	1.0	1.0	Baixo
preparedfood_taxedpriceindollarprice_service.owls	0.9	0.83	0.95	Baixo
preparedfood_priceday_service.owls	0.79	0.73	0.81	Baixo
MerkelD_service.owls	0.83	0.79	0.86	Baixo
MarkoPS_service.owls	0.77	0.64	0.8	Baixo
food_price_service.owls	0.84	0.92	0.96	Baixo
preparedfood_taxfreeprice_service.owls	0.91	0.94	0.96	Baixo
food_maxpricequantity_Aldiservice.owls	0.74	0.73	0.84	Baixo
food_recommendedprice_service.owls	0.81	0.85	0.92	Baixo
preparedfood_GSprice_service.owls	0.65	0.59	0.69	Baixo
Ben_service.owls	0.89	0.89	0.93	Baixo
MAK_service.owls	0.89	0.89	0.93	Baixo
ZAD_service.owls	0.89	0.89	0.93	Baixo
breadorbiscuit_recPricetaxedpriceineuro_service.owls	0.85	0.73	0.88	Baixo
agent_price_MianMarktservice.owls	0.65	0.71	0.8	Alto

Consulta 12: food/ grocerystore_food_service.owls

Esta consulta informa o preço de estabelecimentos comerciais horti-fruti-granjeiros.

Nome do Serviço	Qav	Qre	Qde	Preço
grocerystore_food_service.owls	0.93	1.0	1.0	Baixo
grocerystore_foodquantity_service.owls	0.83	0.81	0.89	Baixo
_food_HEBgroceryCompservice.owls	0.48	0.5	0.5	Baixo
grocerystore_preparedfood_service.owls	0.89	0.92	0.96	Baixo
grocerystore_preparedfoodprice_service.owls	0.75	0.69	0.81	Baixo
mercantileorganization_food_service.owls	0.88	0.97	0.98	Baixo
retailstore_apple_service.owls	0.76	0.65	0.84	Baixo
Available_preparedfoodquantity_service.owls	0.81	0.78	0.87	Baixo
retailstore_preparedfood_service.owls	0.87	0.9	0.95	Baixo
_cerealgrain_PlantSeedservice.owls	0.33	0.17	0.35	Baixo
_honey_Providerservice.owls	0.38	0.26	0.39	Baixo
grocerystore_flourdoughbutter_service.owls	0.85	0.67	0.91	Baixo
grocerystore_teaprice_service.owls	0.82	0.7	0.85	Baixo
drugstore_tea_service.owls	0.86	0.77	0.92	Baixo
store_preparedfood_Merchantservice.owls	0.57	0.52	0.65	Baixo
store_preparedfood_service.owls	0.57	0.52	0.65	Alto

Consulta 13: communication/ title_comedyfilm_service.owls

Esta consulta informa o título de filmes de comédia.

Nome do Serviço	Qav	Qre	Qde	Preço
title_comedyfilm_service.owls	1.0	1.0	1.0	Baixo
title_comedyfilm_Megaservice.owls	1.0	1.0	1.0	Baixo
title_comedyfilm_BFservice.owls	1.0	1.0	1.0	Baixo
_comedyfilmfantacyfilm_service.owls	0.45	0.36	0.46	Baixo
_comedyfilmactionfilm_service.owls	0.45	0.36	0.46	Baixo
title_lowcomedyfilm_service.owls	0.94	0.9	0.95	Baixo
title_highcomedyfilmreport_service.owls	0.64	0.54	0.63	Baixo
title_filmActionComedy_service.owls	0.9	0.88	0.93	Baixo
title_filmP2P_service.owls	0.9	0.88	0.93	Baixo
title_filmpricequality_service.owls	0.7	0.62	0.71	Baixo
title_film_service.owls	0.9	0.88	0.93	Baixo
_filmHighlyRated_service.owls	0.4	0.38	0.43	Baixo
title_media_service.owls	0.71	0.63	0.73	Baixo
title_actionfilm_service.owls	0.84	0.8	0.88	Alto

Consulta 14: communication/ title_videomedia_service.owls

Esta consulta informa os arquivos de media disponíveis para um dado título de filme, independente da classificação do mesmo.

Nome do Serviço	Qav	Qre	Qde	Preço
title_videomedia_service.owls	0.96	1.0	1.0	Baixo
title_videomediaMM_service.owls	0.96	1.0	1.0	Baixo
title_videomediarecommendedprice_service.owls	0.79	0.69	0.81	Baixo
_videomediaSaturn_service.owls	0.46	0.5	0.5	Baixo
_filmvideomediaDiscoveryChannel_service.owls	0.41	0.35	0.42	Baixo
_videomediaBBC_service.owls	0.46	0.5	0.5	Baixo
_videomediaSmithLee_service.owls	0.46	0.5	0.5	Baixo
linguisticexpression_videomedia_service.owls	0.87	0.83	0.9	Baixo
title_obtainablevideomedia_service.owls	0.92	0.92	0.96	Baixo
title_vhsdvd_service.owls	0.9	0.77	0.92	Baixo
title_vhs_service.owls	0.89	0.86	0.92	Baixo
title_media_service.owls	0.77	0.6	0.7	Alto

Consulta 15: education/ researcher-in-academia_address_service.owls

Esta consulta informa o enderço de um pesquisador.

Nome do Serviço	Qav	Qre	Qde	Preço
researcher-in-academia_address_service.owls	0.97	1.0	1.0	Baixo
researcher-in-academia_address_TREEservice.owls	0.97	1.0	1.0	Baixo
researcher-in-academia_address_ZOOservice.owls	0.97	1.0	1.0	Baixo
researcher_address_service.owls	0.77	0.61	0.85	Baixo
researcher-in-academia_publication-referencepostal-address_.owls	0.72	0.58	0.74	Baixo
researcher_address_HOM2service.owls	0.77	0.61	0.85	Baixo
researcher_postal-address_service.owls	0.72	0.51	0.8	Baixo
employee_postal-address_service.owls	0.75	0.59	0.84	Baixo
researcher_abstract-information_service.owls	0.66	0.48	0.78	Baixo
researcher-in-academia_abstract-information_service.owls	0.86	0.88	0.93	Baixo
professor-in-academia_address_service.owls	0.96	0.99	0.99	Baixo
research-fellow-in-academia_publication-reference_service.owls	0.64	0.56	0.67	Baixo
visiting-researcher_address_service.owls	0.8	0.69	0.89	Alto

Consulta 16: education/ university_lecturer-in-academia_service.owls

A consulta informa o periódico publicado por um determinado centro de pesquisa.

Nome do Serviço	Qav	Qre	Qde	Preço
university_lecturer-in-academia_service.owls	0.98	1.0	1.0	Baixo
university_lecturer-in-academia_Recommendservice.owls	0.98	1.0	1.0	Baixo
university_lecturer-in-academiaCurrentSemmester_service.owls	0.98	1.0	1.0	Baixo
_lecturer-in-academiaMunchenUniversity_service.owls	0.5	0.5	0.5	Baixo
_lecturer-in-academiaSaarlandUniversity_service.owls	0.5	0.5	0.5	Baixo
_lecturer-in-academiaZambiaUniversity_service.owls	0.5	0.5	0.5	Baixo
higher-educational-organization_lecturer-in-academia_MostUsedservice.owls	0.97	0.99	0.99	Baixo
higher-educational-organization_lecturer-in-academia_service.owls	0.97	0.99	0.99	Baixo
university_senior-lecturer-in-academia_service.owls	0.98	0.99	1.0	Baixo
educational-organization_lecturer-in-academia_service.owls	0.96	0.98	0.99	Baixo
learning-centred-organization_lecturer-in-academia_service.owls	0.95	0.98	0.98	Baixo
organization_lecturer-in-academia_service.owls	0.95	0.97	0.98	Baixo
university_professor-in-academia_service.owls	0.98	0.99	1.0	Baixo
higher-educational-organization_professor-in-academia_Pioneerservice.owls	0.96	0.98	0.99	Baixo
higher-educational-organization_professor-in-academia_service.owls	0.96	0.98	0.99	Baixo
higher-educational-organization_senior-research-fellow-in-academia_service.owls	0.96	0.98	0.99	Baixo
higher-educational-organization_senior-research-fellow-in-academia_Firstservice.owls	0.96	0.98	0.99	Baixo
university_academic-support-staff_service.owls	0.97	0.99	0.99	Baixo
university_researcher_service.owls	0.77	0.6	0.83	Baixo
university_research-fellow-in-academia_service.owls	0.97	0.98	0.99	Alto

Consulta 17: education/ governmentdegree_scholarship_service.owls

Esta consulta informa o grau de escolaridade oferecido por um determinado governo.

Nome do Serviço	Qav	Qre	Qde	Preço
governmentdegree_scholarship_service.owls	0.94	1.0	1.0	Baixo
governmentdegree_scholarship_TheBestservice.owls	0.94	1.0	1.0	Baixo
governmentdegree_scholarshipquantity_service.owls	0.88	0.88	0.93	Baixo
government_scholarship_service.owls	0.86	0.93	0.95	Baixo
government_scholarshiporganization_service.owls	0.69	0.64	0.8	Baixo
degreegovernmentorganization_scholarship_service.owls	0.92	0.98	0.99	Baixo
governmentaward_scholarship_service.owls	0.91	0.98	0.99	Baixo
award_scholarship_GermanGovservice.owls	0.7	0.56	0.65	Baixo
award_scholarshipduration_SwissGovservice.owls	0.54	0.27	0.48	Baixo
governmentorganization_scholarship_service.owls	0.84	0.91	0.94	Baixo
degreegovernment_funding_service.owls	0.88	0.94	0.97	Baixo
government_funding_ForPhDservice.owls	0.8	0.87	0.92	Baixo
academic-degreegovernment_scholarship_service.owls	0.93	0.98	0.99	Baixo
degreegovernment_unilateralgiving_service.owls	0.81	0.85	0.91	Baixo
governmentdegree_welfare_service.owls	0.86	0.9	0.94	Baixo
degreenationalgovernment_scholarship_service.owls	0.93	0.98	0.99	Baixo
nationalgovernment_scholarship_service.owls	0.85	0.91	0.94	Baixo
nationalgovernment_scholarshipquantity_service.owls	0.78	0.79	0.87	Baixo
government_welfare_service.owls	0.77	0.83	0.9	Baixo
nationalgovernment_scholarshipquantityduration_service.owls	0.65	0.59	0.75	Baixo
nationalgovernment_physical-quantityscholarshiplandarea_service.owls	0.72	0.68	0.81	Alto

Consulta 18: education/ publication-number_publication_service.owls

Esta consulta informa uma publicação dado o número ISBN.

Nome do Serviço	Qav	Qre	Qde	Preço
publication-number_publication_service.owls	0.99	1.0	1.0	Baixo
publication-number_currencypublication_service.owls	0.99	0.99	0.99	Baixo
publication-number_publicationauthor_service.owls	0.99	0.99	0.99	Baixo
_publication_PPservice.owls	0.01	0.0	0.01	Baixo
AcademicBookNumberOrISBNSearch.owls	0.95	0.89	0.95	Baixo
publication-number_book_Portalservice.owls	0.95	0.89	0.95	Baixo
publication-number_book_service.owls	0.95	0.89	0.95	Baixo
publication-number_edited-book_service.owls	0.94	0.88	0.95	Baixo
publication-number_bookauthorpublisher_service.owls	0.94	0.88	0.95	Baixo
_book_Oracleservice.owls	0.45	0.39	0.45	Baixo
_journal_Tutorialservice.owls	0.47	0.3	0.49	Baixo
AcademicBookNumberSearch.owls	0.8	0.64	0.8	Baixo
academic-item-number_book_service.owls	0.8	0.64	0.8	Baixo
isbn_book_service.owls	0.8	0.64	0.8	Baixo
isbn_publication_service.owls	0.85	0.75	0.84	Alto

Consulta 19: education/ novel_author_service.owls

Esta consulta informa o autor dado um tipo de publicação, exceto ficção científica.

Nome do Serviço	Qav	Qre	Qde	Preço
novel_author_service.owls	0.98	1.0	1.0	Baixo
novel_author_BookOntoservice.owls	0.98	1.0	1.0	Baixo
novel_author_MyOntoservice.owls	0.96	0.92	0.98	Baixo
novel_authorgenre_service.owls	0.89	0.83	0.9	Baixo
novel_userreviewauthor_service.owls	0.84	0.75	0.84	Baixo
novel_authorcommitting_service.owls	0.71	0.6	0.7	Baixo
novel_authortime_service.owls	0.89	0.83	0.9	Baixo
book_author_service.owls	0.94	0.94	0.97	Baixo
book_authorbook-type_service.owls	0.85	0.78	0.87	Baixo
book_authorprice_Novelservice.owls	0.68	0.56	0.7	Baixo
book_authortext_service.owls	0.6	0.49	0.61	Baixo
publication_author_service.owls	0.77	0.7	0.81	Baixo
novel_person_Writerservice.owls	0.79	0.75	0.84	Alto

Consulta 20: education/ country_skilledoccupation_service.owls

Esta consulta informa as oportunidades de trabalho disponíveis em um determinado país.

Nome do Serviço	Qav	Qre	Qde	Preço
country_skilledoccupation_jobsservice.owls	0.99	1.0	1.0	Baixo
country_skilledoccupation_service.owls	0.99	1.0	1.0	Baixo
country_skilledoccupationparttimeposition_service.owls	0.96	0.86	0.97	Baixo
country_skilledoccupationtimeduration_service.owls	0.83	0.73	0.83	Baixo
country_profession_service.owls	0.95	0.92	0.96	Baixo
geopolitical-entity_skilledoccupation_service.owls	0.96	0.97	0.98	Baixo
geopolitical-entity_skilledoccupationcompany_service.owls	0.81	0.72	0.82	Baixo
geographical-region_organizationskilledoccupation_service.owls	0.79	0.67	0.8	Baixo
_professiongeographical-region_USservice.owls	0.27	0.15	0.28	Baixo
_bankeraddress_CityBankservice.owls	0.35	0.25	0.34	Baixo
country_sportsposition_service.owls	0.95	0.92	0.96	Baixo
_medicaldoctor_UNOservice.owls	0.42	0.36	0.42	Baixo
citycountry_skilledoccupation_service.owls	0.82	0.71	0.82	Baixo
companycountry_skilledoccupation_service.owls	0.83	0.69	0.84	Baixo
public-companycountry_skilledoccupation_service.owls	0.83	0.68	0.84	Alto

Consulta 21: medical/ hospital_investigating_service.owls

Esta consulta informa os exames disponíveis em um determinado hospital.

Nome do Serviço	Qav	Qre	Qde	Preço
hospital_investigating_service.owls	0.92	1.0	1.0	Baixo
hospital_investigatingaddress_service.owls	0.86	0.89	0.93	Baixo
hospital_postal-addressinvestigating_service.owls	0.85	0.87	0.92	Baixo
_investigating_Saarservice.owls	0.47	0.5	0.5	Baixo
hospital_diagnosticprocess_MedDiagservice.owls	0.9	0.97	0.98	Baixo
hospital_diagnosticprocesscost_service.owls	0.86	0.89	0.93	Baixo
HDP2_service.owls	0.83	0.84	0.89	Baixo
HDP_service.owls	0.78	0.76	0.86	Baixo
CAD_medical_service.owls	0.72	0.74	0.83	Baixo
_diagnosticprocessorganization_service.owls	0.34	0.27	0.38	Baixo
organization_diagnosticprocess_service.owls	0.76	0.84	0.9	Baixo
organization_diagnosticprocesscost_service.owls	0.72	0.76	0.85	Baixo
hospital_IIPsummary_service.owls	0.76	0.75	0.85	Baixo
hospital_predicting_service.owls	0.87	0.94	0.96	Baixo
questionhospital_diagnosticprocess_service.owls	0.87	0.91	0.95	Alto

Consulta 22: travel/ surfing_destination_service.owls

Esta consulta informa lugares para a prática de surfing.

Nome do Serviço	Qav	Qre	Qde	Preço
surfing_destination_service.owls	0.96	0.84	0.96	Baixo
surfing_destination_AUSservice.owls	0.96	0.72	0.96	Baixo
surfing_destination_SOHservice.owls	0.91	0.9	0.92	Baixo
surfing_destination_Alwaysservice.owls	0.96	0.7	0.96	Baixo
surfing_beach_service.owls	0.78	0.59	0.77	Baixo
sports_ruralarea_service.owls	0.78	0.62	0.74	Baixo
surfing_ruralarea_service.owls	0.75	0.54	0.72	Baixo
sports_beach_service.owls	0.7	0.54	0.69	Baixo
sports_destination_service.owls	0.83	0.81	0.84	Baixo
sports_nationalpark_service.owls	0.55	0.34	0.51	Baixo
sports_town_service.owls	0.72	0.33	0.71	Baixo
surfing_farmland_service.owls	0.77	0.38	0.73	Baixo
surfing_nationalpark_service.owls	0.5	0.37	0.45	Alto

Consulta 23: travel/ surfinghiking_destination_service.owls

Esta consulta informa lugares para a prática de surfing e caminhada ecológica.

Nome do Serviço	Qav	Qre	Qde	Preço
surfinghiking_destination_service.owls	0.96	0.76	0.96	Baixo
surfinghiking_destination_DFGservice.owls	0.9	0.93	0.91	Baixo
surfinghiking_destination_PDSservice.owls	0.99	0.88	0.98	Baixo
surfinghiking_destination_PFservice.owls	0.92	0.95	0.94	Baixo
surfinghiking_ruralarea_service.owls	0.79	0.6	0.79	Baixo
sports_destination_service.owls	0.84	0.77	0.86	Baixo
sports_ruralarea_service.owls	0.78	0.51	0.75	Baixo
hikingsurfing_city_service.owls	0.51	0.37	0.49	Baixo
surfinghiking_city_SFservice.owls	0.62	0.42	0.59	Baixo
surfinghiking_city_service.owls	0.71	0.4	0.65	Baixo
surfinghiking_nationalpark_service.owls	0.52	0.34	0.5	Baixo
surfinghiking_town_service.owls	0.71	0.6	0.68	Baixo
sports_nationalpark_service.owls	0.55	0.45	0.51	Alto

Consulta24:travel/ geographical-Regiongeographical-region_map_service.owls

Esta consulta informa as rotas de viagem entre cidades da Europa.

Nome do Serviço	Qav	Qre	Qde	Preço
geographical-regiongeographical-region_map_service.owls	0.97	1.0	1.0	Baixo
geographical-regiongeographical-region_map_Gorgservice.owls	0.97	1.0	1.0	Baixo
geographical-region_mapFromFrankfurt_service.owls	0.97	1.0	1.0	Baixo
geographical-region_mapToBerlin_service.owls	0.97	1.0	1.0	Baixo
_mapFrankfurtBerlin_service.owls	0.47	0.5	0.5	Baixo
_mapGerman_service.owls	0.47	0.5	0.5	Baixo
locationlocation_map_service.owls	0.93	0.86	0.96	Baixo
locationlocation_map_SRIservice.owls	0.93	0.86	0.96	Baixo
geographical-regiongeographical-region_icon_service.owls	0.91	0.94	0.97	Baixo
locationlocation_icon_service.owls	0.87	0.79	0.93	Baixo
citycity_map_service.owls	0.9	0.87	0.92	Baixo
citycity_arrowfigure_service.owls	0.81	0.76	0.85	Baixo
locationlocation_arrowfigure_service.owls	0.84	0.75	0.9	Baixo
country_map_service.owls	0.92	0.89	0.93	Alto

Consulta 25: travel/ surfingorganization_destination_service.owls

Esta consulta informa localidades para organizações para pratica de surfing.

Nome do Serviço	Qav	Qre	Qde	Preço
surfingorganization_destination_service.owls	0.94	0.99	0.94	Baixo
surfingorganization_destination_Bestservice.owls	0.99	1.0	0.99	Baixo
surfingorganization_destination_SODservice.owls	0.99	0.99	0.99	Baixo
surfingorganization_destination_SAARservice.owls	0.97	0.96	0.98	Baixo
sportsorganization_destination_service.owls	0.96	0.95	0.97	Baixo
legal-agentsurfing_destination_service.owls	0.77	0.63	0.77	Baixo
sportslegal-agent_destination_service.owls	0.68	0.6	0.68	Baixo
generic-agentsports_destination_service.owls	0.69	0.6	0.7	Baixo
surfinggeneric-agent_destination_service.owls	0.77	0.6	0.78	Baixo
surfingorganization_city_service.owls	0.6	0.49	0.56	Baixo
surfinggeneric-agent_city_service.owls	0.31	0.14	0.29	Baixo
learning-centred-organizationsurfing_destination_service.owls	0.95	0.9	0.95	Baixo
surfingorganizationperson_destination_service.owls	0.95	0.99	0.96	Alto

Consulta 26: travel/ citycountry_hotel_service.owls

Esta consulta informa o hotel em uma determinada cidade de um país.

Nome do Serviço	Qav	Qre	Qde	Preço
citycountry_hotel_service.owls	0.95	0.61	0.95	Baixo
countrycity_hotel_service.owls	0.82	0.65	0.82	Baixo
citycountry_destinationhotel_service.owls	0.67	0.43	0.65	Baixo
countrycity_sportshotel_service.owls	0.55	0.3	0.54	Baixo
_hotel_Worldwideservice.owls	0.36	0.22	0.34	Baixo
city_hotel_Germanservice.owls	0.77	0.54	0.78	Baixo
city_hotel_Saarlandservice.owls	0.47	0.33	0.45	Baixo
city_hotel_service.owls	0.97	0.92	0.99	Baixo
country_hotel_service.owls	0.78	0.56	0.8	Baixo
countrycity_luxuryhotel_service.owls	0.59	0.37	0.58	Baixo
countrycity_luxuryhotel_Gelservice.owls	0.63	0.45	0.63	Baixo
__luxuryhotel_Heidelburgservice.owls	0.34	0.19	0.31	Baixo
geographical-region_hotel_service.owls	0.72	0.44	0.75	Baixo
geopolitical-entity_hotel_service.owls	0.93	0.89	0.97	Baixo
citycountry_accommodation_service.owls	0.71	0.56	0.71	Baixo
durationcountrycity_hotel_service.owls	0.64	0.42	0.61	Baixo
countrycapital-city_hotel_service.owls	0.77	0.6	0.78	Baixo
countryvillage_hotel_service.owls	0.96	0.97	0.98	Baixo
towncountry_hotel_service.owls	0.82	0.69	0.82	Baixo
village_hotel_service.owls	0.71	0.49	0.72	Alto

Consulta 27: travel/ geopolitical-entity_weatherprocess_service.owls

Esta consulta informa a previsão do tempo para determinada região ou país da Europa.

Nome do Serviço	Qav	Qre	Qde	Preço
geopolitical-entity_weatherprocess_service.owls	0.96	1.0	1.0	Baixo
_weatherprocess_Germanservice.owls	0.47	0.5	0.5	Baixo
geographical-region_weatherprocess_service.owls	0.92	0.92	0.95	Baixo
geopolitical-entity_weathersystem_service.owls	0.92	0.92	0.96	Baixo
geographical-region_weatherprocess_GRWservice.owls	0.92	0.92	0.95	Baixo
geopolitical-entity_weatherseason_service.owls	0.92	0.92	0.96	Baixo
geographical-region_weatherseason_service.owls	0.88	0.84	0.91	Baixo
geographical-region_weathersystem_service.owls	0.88	0.84	0.91	Baixo
geopolitical-entity_weatherseasonproposition_service.owls	0.86	0.81	0.89	Baixo
geopolitical-entity_weatherseasontimeposition_service.owls	0.8	0.71	0.81	Baixo
_warmfront_Italyservice.owls	0.4	0.36	0.42	Baixo
_weatherfront_Britishservice.owls	0.43	0.42	0.46	Baixo
_weatherfront_Germanyservice.owls	0.22	0.08	0.15	Baixo
_icing_Germanservice.owls	0.43	0.42	0.46	Baixo
geographical-region_lightning_service.owls	0.86	0.75	0.9	Baixo
country_weatherprocess_service.owls	0.95	0.97	0.98	Alto

Consulta 28: weapon/ governmentmissile_funding_service.owls

Esta consulta fornece informações sobre fundos governamentais de pesquisa a mísseis balísticos.

Nome do Serviço	Qav	Qre	Qde	Preço
governmentmissile_funding_service.owls	0.96	1.0	1.0	Baixo
governmentmissile_funding_Reliableservice.owls	0.96	1.0	1.0	Baixo
missile_funding_Asianservice.owls	0.9	0.85	0.92	Baixo
missile_funding_NKoreaservice.owls	0.9	0.85	0.92	Baixo
missile_funding_Indiaservice.owls	0.9	0.85	0.92	Baixo
missile_funding_Pakservice.owls	0.9	0.85	0.92	Baixo
government_funding_BallMissileservice.owls	0.81	0.81	0.9	Baixo
government_funding_Missileservice.owls	0.81	0.81	0.9	Baixo
governmentmissileweapon_funding_service.owls	0.96	0.99	1.0	Baixo
governmentmissileweapon_fundingfunding_service.owls	0.96	0.99	1.0	Baixo
governmentmissile_financing_service.owls	0.87	0.77	0.92	Baixo
projectilegovernment_funding_service.owls	0.91	0.95	0.98	Baixo
missilegovernmentorganization_funding_service.owls	0.94	0.99	0.99	Baixo
missile_financing_Chinaservice.owls	0.82	0.62	0.83	Baixo
missile_financing_Russianservice.owls	0.82	0.62	0.83	Baixo
missile_financing_USservice.owls	0.82	0.62	0.83	Baixo
governmentweapon_funding_service.owls	0.9	0.94	0.97	Baixo
missilegovernment_giving_service.owls	0.91	0.94	0.97	Baixo
ballisticmissilegovernment_fundingrange_service.owls	0.92	0.93	0.96	Baixo
governmentorganizationmissile_unilateralgiving_service.owls	0.87	0.88	0.93	Baixo
nationalgovernmentweapon_funding_service.owls	0.89	0.93	0.96	Alto