



HOLOS

ISSN: 1518-1634

holos@ifrn.edu.br

Instituto Federal de Educação, Ciência e
Tecnologia do Rio Grande do Norte
Brasil

Fernandes Araújo Ribeiro, Cláudia Maria
SOBRE A UTILIZAÇÃO DE MEDIADORES SEMÂNTICOS PARA MONITORAMENTO
DE QUALIDADE DE SERVIÇO NA WEB
HOLOS, vol. 1, 2010, pp. 119-129
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Natal, Brasil

Disponível em: <http://www.redalyc.org/articulo.oa?id=481549219009>

- Como citar este artigo
- Número completo
- Mais artigos
- Home da revista no Redalyc

redalyc.org

Sistema de Informação Científica
Rede de Revistas Científicas da América Latina, Caribe, Espanha e Portugal
Projeto acadêmico sem fins lucrativos desenvolvido no âmbito da iniciativa Acesso Aberto

**SOBRE A UTILIZAÇÃO DE MEDIADORES SEMÂNTICOS PARA
MONITORAMENTO DE QUALIDADE DE SERVIÇO NA WEB**

Cláudia Maria Fernandes Araújo Ribeiro

IFRN – Campus Natal Central, claudia.ribeiro@ifrn.edu.br

UERN – Campus de Natal, claudiaribeiro@uern.br

RESUMO

Mediadores são entidades especializadas na resolução de conflitos. Um cenário típico em que mediadores desempenham um papel fundamental é durante a negociação de serviços na Web, uma vez que provedores e clientes freqüentemente têm objetivos opostos. Por exemplo, na perspectiva do provedor o objetivo é maximizar lucros através do uso racional de recursos, ao passo que clientes buscam maximizar a satisfação com o custo mínimo. Mediadores computacionais preservam similaridades com seus pares no mundo real e ambos necessitam de informações específicas para realizar seu trabalho. Este artigo tem como objetivo discutir o uso de mediadores semânticos por aplicações orientadas a serviço, durante a negociação de requisitos de qualidade na Web.

PALAVRAS-CHAVE: SOA, Web Semântica, Mediadores, QoS

**ON THE USE OF SEMANTIC MEDIATORS FOR QOS MONITORING IN
WEB APPLICATIONS**

ABSTRACT

Mediators are specialized entities used in conflict situations. The service negotiation scenario is a typical case where mediators play a key role, since provider and consumer frequently have opposite goals, e.g. to maximize profit with minimal resource (provider perspective) against to maximize satisfaction with minimal cost (consumer perspective). The computational mediators tend to preserve similarities from their pairs in the human life context and both of them need specific information to do their job. This paper aims discussing the use of semantic mediators by service-oriented applications, during quality of service negotiation.

KEYWORDS: SOA, Semantic Web, Mediators, QoS

SOBRE A UTILIZAÇÃO DE MEDIADORES SEMÂNTICOS PARA MONITORAMENTO DE QUALIDADE DE SERVIÇO NA WEB

INTRODUÇÃO

O aumento da oferta de serviços na Web aponta para a adoção crescente da arquitetura orientada a serviço (SOA) no desenvolvimento de aplicações distribuídas. A principal expectativa em torno da tecnologia *Web Services* (Serviços Web) está relacionada à profunda mudança na forma como a Web tem sido usada, de uma perspectiva humana para uma perspectiva de software.

O sucesso de aplicações compostas, nas quais as funcionalidades são suportadas por diversos serviços autônomos trabalhando conjuntamente, depende de uma gestão eficaz da qualidade de serviço (QoS). Requisitos de QoS, que são comumente classificados como requisitos não-funcionais (NFR) [1], podem servir de base a um contrato entre os serviços envolvidos no atendimento às necessidades dos usuários. Além disso, o processo de seleção de serviço também pode utilizar os requisitos de QoS como parâmetros para distinguir entre serviços similares.

A Web como conhecemos é essencialmente centrada no usuário e baseada na habilidade do ser humano em lidar e interpretar documentos textuais. A evolução da Web para a Web Semântica [2] representa um importante passo para a implantação de aplicações orientadas a serviço cujo significado possa ser automaticamente interpretado. O ambiente da Web Semântica é baseado em representação do conhecimento e regras de inferência, onde agentes de software podem realizar tarefas sofisticadas para os usuários. Embora a descrição do serviço Web seja focado em aspectos de automação, o enriquecimento por adição de semântica possibilita, entre outros benefícios, lidar com aspectos subjetivos como requisitos de QoS.

Qualidade de serviço pode ser tratada em diferentes níveis de abstração. Do ponto de vista do usuário, QoS pode ser usada para melhor expressar uma noção subjetiva de qualidade, por exemplo, desempenho. Por outro lado, os mecanismos de mais baixo nível que efetivamente realizam QoS usam parâmetros mais concretos, tais como, atraso e taxa de transferência. Neste contexto, serviços sensíveis a noção de QoS tornaram-se particularmente relevantes para a negociação de recursos e especialmente úteis em sistemas multimídia, onde o lado cliente da aplicação age em nome da pessoa humana, que são típicos usuários que não têm conhecimento de questões técnicas mas que são sensíveis à possível degradação do nível de QoS acordado.

O tratamento de QoS, contudo, não é uma tarefa trivial, pois implica em lidar com vários conceitos potencialmente conflitantes. Portanto, é fundamental estabelecer um consenso em torno de um vocabulário de QoS antes que possa ser efetivamente utilizado por *Web Services*. Uma solução importante para esse problema é o uso de ontologias [2] para representar conceitos comuns relacionados a QoS e *Web Services*, bem como o uso de mediadores [5] para efeitos de integração no caso de múltiplas ontologias. Apesar do uso de ontologias em QoS [3] [4], não ser novo, a sua utilização é geralmente focada em aspectos técnicos do tratamento de QoS, e exige uma investigação mais profunda sob a perspectiva do usuário. Por exemplo, o uso de ontologias para descrever *Web Services* é tratado pelas especificações OWL-S [5] e WSMO [6], mas QoS ainda é uma questão em aberto nestas iniciativas. Estes são precisamente os aspectos tratados neste trabalho, que

visa investigar e discutir aspectos abstratos da mediação no contexto do aprovisionamento de QoS e negociação de recursos na Web.

Este artigo está organizado da seguinte forma. A segunda seção introduz conceitos fundamentais utilizados no âmbito deste trabalho, relacionados principalmente a anotação semântica de serviços. A terceira seção descreve duas ontologias propostas: NFR e SLA, seguidas da descrição de mediadores WSMO concebidos para um cenário de negociação de QoS. A seção seguinte descreve uma arquitetura proposta para a mediação de QoS em um cenário de mediação. Finalmente, as conclusões e trabalhos futuros são apresentados na última seção.

ANOTAÇÃO SEMÂNTICA DE SERVIÇOS

O ambiente da Web Semântica permite aos usuários e agentes de software descobrir, invocar, compor e monitorar os serviços. A especificação OWL-S [5], por exemplo, consiste em um conjunto de ontologias que aborda a automatização destas atividades. Há três ontologias em OWL-S: (1) *Service Profile*, que apresenta o que o serviço faz; (2) *Service Model* que descreve como um serviço trabalha; e (3) *Service Grounding* que detalha como interagir com o serviço. Apesar de OWL-S não tratar explicitamente as questões de QoS, permite adicionar atributos de qualidade que podem ser usados por mecanismos de descoberta, através da ontologia *Service Profile*.

Web Service Modeling Ontology (WSMO) descreve vários aspectos de serviços semânticos [6]. Os quatro elementos de nível mais abstrato de WSMO são: (1) *Ontology* que fornece a terminologia utilizada por outros elementos WSMO; (2) *Service* que descreve *Web Services*, que são requisitados, providos e acordados entre requisidores e provedores; (3) *Gol* que descreve os problemas que devem ser resolvidos pelos serviços; e (4) *Mediator* que lida com problemas de interoperabilidade entre os diferentes elementos WSMO.

Aplicativos que têm rigorosas limitações de tempo, como os sistemas de multimídia, em que a resposta certa entregue tarde demais pode tornar-se a resposta errada, são definidos como aplicações sensíveis a QoS (*QoS-aware*) [8]. Neste tipo de aplicação, muitos serviços devem ser *QoS-aware*, incluindo os mediadores. O aprovisionamento de QoS é um aspecto que perpassa múltiplas camadas da aplicação ou perspectivas [9] e o processo de negociação de recursos representa um mapeamento entre os requisitos de QoS do usuário em parâmetros de QoS utilizados pelos mecanismos de QoS, resultando na efetiva alocação de recursos. Como diferentes interesses são envolvidos na negociação de recursos a utilização de mediador torna-se crucial para resolver os potenciais conflitos. A seção a seguir detalha duas ontologias no domínio de QoS [2], criadas no âmbito deste trabalho.

ONTOLOGIAS DE QOS

O uso de ontologia para representar conceitos de QoS não é uma abordagem nova. Uma ontologia de QoS para fluxos de trabalho (*workflow*) e *Web Services* é descrito em [13]. Existem diversas ontologias sobre QoS concebidas para fins específicos. A ontologia de QoS proposta neste trabalho pretende ser mais geral, mas um cenário específico de negociação de recursos nortearam a sua concepção.

Requisitos não funcionais (NFRs) pretendem descrever um sistema de software de forma qualitativa, abordando "como" o software funcionará em vez de "o que" ele vai fazer [1]

[14]. Apesar de existirem vários tipos de taxonomias para NFRs, parece que há um consenso sobre a classificação de QoS como requisito não funcional, mas o contrário não é necessariamente verdadeiro. A concepção da ontologia NFR proposta foi orientada em dois sentidos: a inclusão de requisitos que podem ser usados para restringir ou qualificar algum aspecto do software, e um conjunto de propriedades que caracterizam cada um desses requisitos. Ambos foram extraídos de [1].

A Figura 1 ilustra a ontologia NFR. No lado direito, um conjunto de propriedades é detalhado. Estas propriedades representam uma forma de classificar NFRs em categorias e para distinguir os requisitos não funcionais. Ambos, conceitos e propriedades foram definidos através de elementos OWL [11].

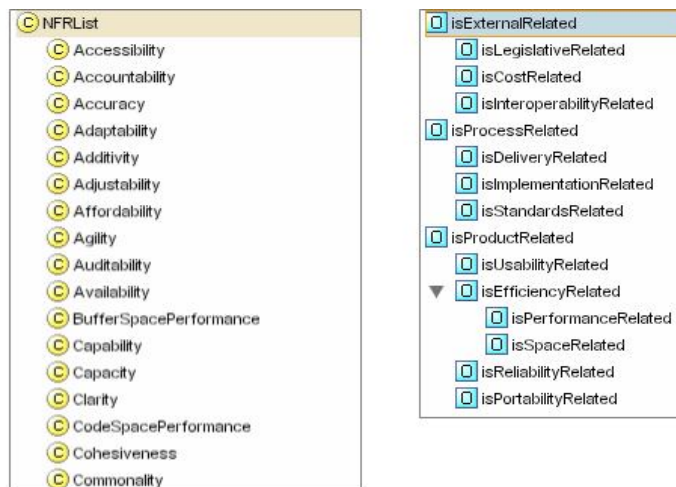


Figura 1. Visão parcial da ontologia NFR, suas classe e propriedades

Tradicionalmente, os contratos de serviços ou SLAs (*Service Level Agreements*) são definidos através de parâmetros técnicos que são essenciais para a configuração e monitoramento do serviço, mas frequentemente eles não têm nenhum significado para a maioria dos usuários. Um contrato de serviço que seja compreensível torna possível que o usuário desempenhe um papel mais ativo durante o processo de negociação de serviços [10]. Este tipo de contrato também agrega valor ao serviço e pode ser usado como diferencial competitivo entre os provedores.

A ontologia SLA inclui informações de QoS de acordo com um modelo de contrato de QoS [10]. A Figura 2 detalha os conceitos de nível superior relacionados ao contrato (SLA), que incluem as seguintes informações de QoS: requisito não funcionais (NFR); atributo de QoS (*QoSAttr*); característica de QoS (*QoSChar*) e parâmetro de QoS (*QoSParam*). SLA também inclui, além do conceito relacionado ao contrato de QoS (*QoSContract*), cinco conceitos: (1) *VirtualResource* que é uma noção abstrata de recursos físicos, (2) *QoSMapTables* representa informações sobre valores de QoS utilizados durante o mapeamento automático de QoS, da perspectiva do usuário para os mecanismos de baixo nível que realmente tratam QoS; (3) *Boolean* que representa valores verdadeiro/falso; (4) *Level* que compreende variáveis linguísticas através das quais o usuário define o nível de QoS desejado; e (5) *Status* que representa a situação do contrato.

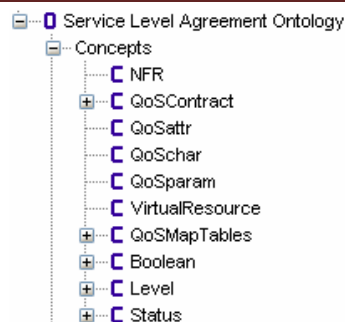


Figura 2. Conceitos de alto nível da ontologia SLA

O contrato de QoS serve como base para o processo de negociação, descrevendo os requisitos de QoS acordado por todos os participantes da negociação. A Figura 3 detalha os sub-conceitos do contrato de QoS. Existem três componentes: (1) *UserQoSspec* que inclui requisitos de QoS na perspectiva do usuário, (2) *ApplicationQoSspec* que representa QoS na perspectiva de aplicação, (3) *ServiceQoSspec* componente técnico representa QoS na perspectiva do serviço, semelhante ao SLS - *Service Level Specification* [12]. Este último composto por cinco elementos: (1) *QoSagreed* que representa o produto final de negociação de QoS; (2) *QoSmeasured* um conjunto de indicadores que torna possível avaliar o nível de QoS; (3) *QoSprovided* que é o nível de QoS efetivamente fornecido pela rede ; (4) *ServiceStatus* que indica a situação do contrato (ativo, em negociação ou violado); (5) *VirtualResourceAlloc* que representa o montante de recursos necessários para realizar o serviço.

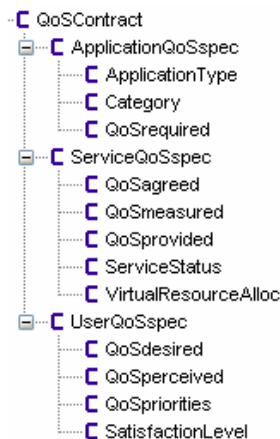


Figura 3. Contrato de QoS

Considerando-se as aplicações orientadas a serviço e a quantidade de informação envolvida durante a negociação de QoS, torna-se evidente a necessidade de agentes de software para mediar este processo. A próxima seção é dedicada a descrever os mediadores como indicado na especificação WSMO e investigar como elas podem ser usadas, no âmbito da negociação de QoS.

UM MODELO PARA CRIAÇÃO DE MEDIADORES

O conceito de mediação é genérico, mas especificamente em Ciência da Computação um mediador é um programa que visa resolver situações de conflito, como a negociação de

recursos para garantir QoS. A especificação WSMO distingue três níveis de mediação [7]. O primeiro deles, mediação no nível de dados (*data-level mediation*), se preocupa com a integração de fontes heterogêneas de dados. O segundo, a mediação em nível de protocolo, trata de tradução entre protocolos de transferência (por exemplo, SOAP, HTTP). A terceira e última, a mediação em nível de processo, refere-se a mediação entre os processos empresariais heterogêneos.

Quatro tipos de mediadores são definidos na ontologia WSMO: *ooMediators*, *ggMediators*, *wgMediators* e *wwMediators*. A primeira descreve mediadores para conflitos entre conceitos de diferentes ontologias (*ooMediators*), que pode ser considerado provavelmente, o tipo mais importante de mediador, uma vez que ontologia é um elemento-chave para representar o conhecimento na Web Semântica. WSMO também define dois tipos de mediadores *ooMediator*: sintáticos que transformam uma determinada ontologia fonte, de uma linguagem de representação para outra, e semânticos que fornecem suporte para resolver problemas de heterogeneidade entre componentes da ontologia fonte e da ontologia alvo. Mediadores do tipo Goal-Goal (*ggMediators*) declaram as relações entre objetivos e mediadores *wgMediators* estabelecem a relação entre um Web Service e um Goal, e resolvem eventuais desajustamentos entre eles. Finalmente, mediadores WebService-WebService (*wwMediator*) pretendem resolver problemas de heterogeneidade entre as descrições de Web Services. As próximas subseções são dedicadas a discutir sobre as situações de conflito que podem ocorrer no domínio do problema, quando a mediação é plenamente justificada.

MEDIAÇÃO DE CONCEITOS DE QOS

A quantidade de informações sobre QoS surgidas durante discussões prévias indicam inúmeras situações potenciais de conflito. Por exemplo, NFR é um conceito chave de SLA (ver Figura 2) que identifica os requisitos funcionais através dos quais o usuário define a QoS desejada (veja a Figura 3). No entanto, os requisitos de QoS são considerados um tipo específico de NFR. Assim, uma instância de NFR no SLA é uma classe de *NFRlist* na ontologia NFR (ver Figura 1). É interessante notar que apenas um subconjunto de *NFRlist* é relevante para a negociação de QoS, notadamente aqueles conceitos relacionados com a fase de execução do ciclo de vida do software. Assim, é necessário distingui-los. As propriedades definidas na ontologia NFR podem ser usadas para esta finalidade, por exemplo, a propriedade *isProductRelated*. Note-se que duas linguagens diferentes para ontologia (OWL e WSMO) foram utilizadas, respectivamente para NFR e SLA, fato que também necessita de mediação. A Figura 5 ilustra um mediador semântico *ooMediator*, conceitualmente concebida para integrar e alinhar NFR e ontologias SLA.

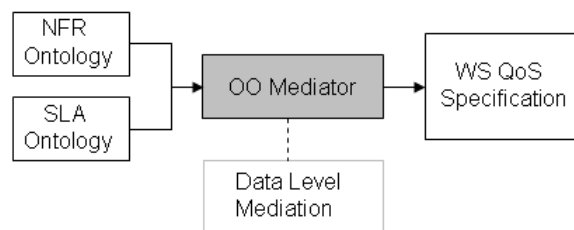


Figura 4. Representação gráfica de mediador WSMO *ooMediator* (adaptado de[7])

Este mediador é usado por *Web Services* para especificação de QoS. Normalmente, este serviço pode ser usado para apoiar a especificação da qualidade desejada pelo usuário, distinguindo NFRs que devem ser indicados no contrato de QoS para explicitar o nível de QoS acordada. Por exemplo, NFRs relacionados com o desenvolvimento do processo, por exemplo, "manutenibilidade", são considerados inadequados para expressar restrições em tempo de execução. Por outro lado, "custo" é um NFR relacionado com as fases de desenvolvimento e execução de software. Suponha que várias ontologias de mesmo domínio poderiam ser utilizadas, por exemplo, duas ontologias NFR. Neste caso, é possível usar um *ooMediator* sintático para traduzir os conceitos de um a outro, ou para gerar uma terceira ontologia que representa a fusão das outras existentes.

MEDIAÇÃO DE DESCRIÇÃO DE SERVIÇO

Vamos supor um ambiente heterogêneo onde aplicações orientadas a serviço necessitam negociar requisitos de qualidade de serviço, tais como videoconferência e VoIP, e que as funcionalidades de cada serviço foram descritas de acordo com diferentes ontologias, por exemplo, OWL-S e WSMO. Dar suporte a heterogeneidade entre diferentes descrições de *Web Services* é outra possível situação em que um mediador poderia facilitar a interoperabilidade. A Figura 6 ilustra a especificação de *Web Service* anteriormente mencionada e um *ooMediator* para o mapeamento de representação OWL-S em representação WSMO. Este mediador pode ser usado por *Web Services* para especificação de QoS.

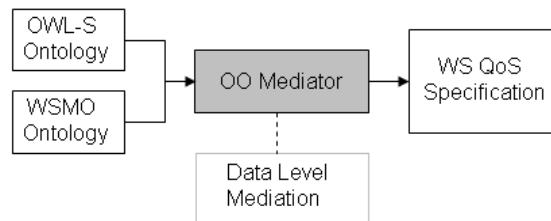


Figura 5. Representação gráfica de WSMO *ooMediator* para tradução de diferentes representações de Web Services (adaptado de [7])

Possivelmente outras situações que envolvem incompatibilidade entre descrições de serviços podem existir no complexo cenário de negociação de QoS. Diferentes tipos de mediadores WSMO podem ser utilizados, tais como *wgMediators*, que podem ser úteis para a descoberta de potenciais serviços capazes de satisfazer as necessidades dos usuários. A mediação neste caso é utilizada para identificar as capacidades dos serviços.

MEDIAÇÃO DO NÍVEL DE QOS

Negociação de QoS compreende múltiplos interesses que são muitas vezes conflitantes, podendo resultar em acordo ou não. Qualidade de serviço acordada representa uma negociação do tipo "ganha-ganha" quando o solicitante e o fornecedor conseguem estabelecer um consenso em torno de condições para a prestação do serviço. Esta é uma situação típica onde a mediação é necessária (ver Figura 7).

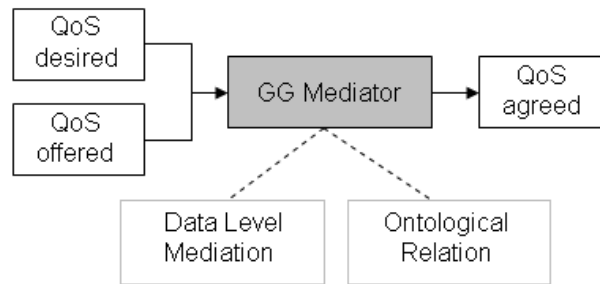


Figura 6. Representação gráfica de WSMO ggMediator para negociação do nível de QoS (adaptado de [7])

WSMO *ggMediators* são convenientes para resolver este tipo de situação, onde há uma clara intenção de maximizar os benefícios (na perspectiva do solicitante) e para minimizar a utilização de recursos no lado oposto (na perspectiva do fornecedor). Este tipo de mediador pode ser usado para selecionar o melhor "serviço", a partir da perspectiva do usuário, executando técnicas de mediação baseadas em análise custo-benefício. Em outras palavras, significa analisar se a melhoria do nível de QoS realmente justifica o aumento no preço do serviço.

O provedor de serviço oferece um nível de QoS (*QoSoffered*) que atende (ou não) o nível de QoS desejado pelo usuário (*QoSdesired*), resultando (ou não) em um acordo (*QoSagreed*). A QoS oferecida, como proposta neste trabalho, é utilizada apenas como ponto de partida para a negociação de QoS, em um mecanismo para distinguir serviços com funcionalidades similares.

Após a criação do SLA, a execução do serviço é iniciada. Durante esta fase, diversos mecanismos de monitoramento, tais como serviços de Middleware (*QoSmeasured*), serviço de rede (*QoSprovided*) e da percepção humana (*QoS percebida*) podem ser usados. Esta é outra situação onde a mediação é necessária. Possivelmente, ocorre degradação de QoS, mudando o nível de QoS acordado e conseqüentemente o nível de satisfação do usuário. A Figura 8 representa esta situação.

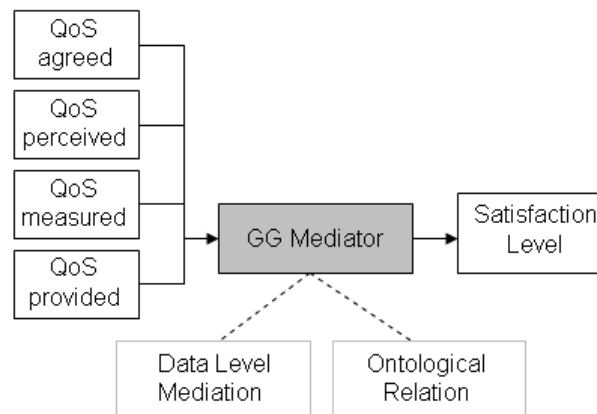


Figura 7. Representação gráfica de WSMO ggMediator para monitoramento de QoS (adaptado de [7])

CENÁRIO DE USO PARA MEDIADORES DE QoS

Esta seção lida com um cenário de uso para negociação QoS que visa enfatizar o papel dos mediadores no contexto dos serviços semânticos. Para isto, foi concebida uma extensão da arquitetura SOA (*Service-Oriented Architecture*), conforme Figura 9. Foram mantidos os componentes básicos de SOA na arquitetura estendida, quais sejam: o solicitante de serviço (ou cliente), o prestador de serviço e o repositório (ou o registro de serviço).

Dois novos componentes foram incluídos: Ontologia e SWS (Servidor Web Semântico). O primeiro representa as ontologias definidas no escopo deste trabalho, ou seja, SLA e NFR, ambas descritas nas seções anteriores. A ontologia WS representa ontologias de descrição de *Web Services* existentes, tais como o OWL-S e WSMO.

Serviços Web semânticos, incluindo aqueles sensíveis a QoS, foram resumidos pelo componente SWS. Na arquitetura proposta, os mediadores são representados por três (possivelmente compostos e complexos) mediadores: (1) *Negotiation Mediator* para negociação do nível de QoS; (2) *Service Mediator* que lida com mediação de descrição de Web Services; e (3) *Ontology Mediator* para reconciliação de ontologias de QoS, incluindo técnicas de fusão e alinhamento.

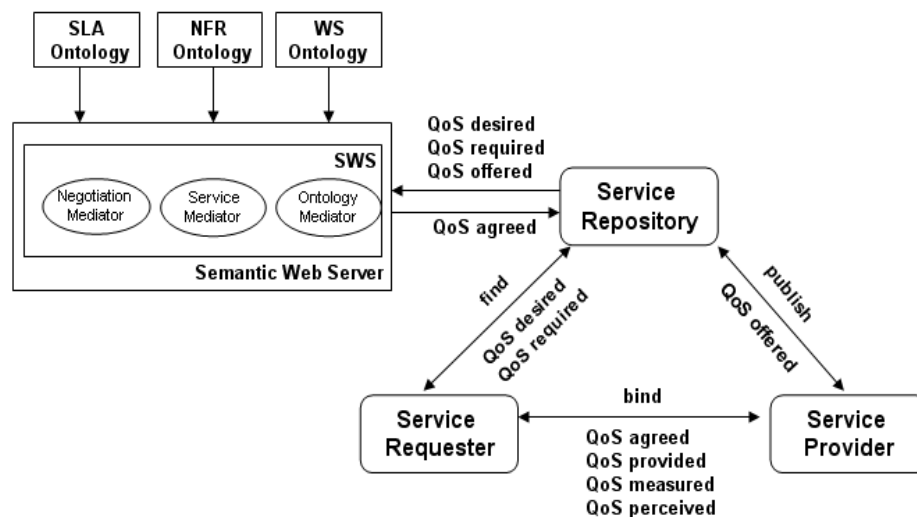


Figura 8. Uma extensão da arquitetura SOA para negociação semântica de QoS

As atividades básicas, descritas na arquitetura SOA, foram modificadas na proposta de modo a introduzir explicitamente os requisitos de QoS. O anúncio ou publicação (*publish*) de um serviço pelo fornecedor (*Service Provider*), em um registro de serviço (*Service Repository*), foi enriquecido pela inclusão do nível de QoS oferecido (*QoSoffered*). O cliente (*Service Requester*) tenta encontrar (*find*) no repositório (*Service Repository*) um serviço de acordo com a especificação de QoS do usuário (*QoSdesired*) e os requisitos de QoS da aplicação (*QoSrequired*).

As atividades de descoberta e seleção são feitas pelos serviços semânticos, com capacidades de mediação e informações de QoS (*QoSdesired*, *QoSrequired* e *QoSoffered*), como mostrado na figura 9. Se uma negociação bem sucedida ocorreu, um acordo sobre o nível de QoS a ser fornecido (*QoSagreed*) é estabelecido, o serviço correspondente

(possivelmente composto) pode ser invocado automaticamente (*bind*) e os recursos podem ser alocados (*VirtualResources*).

Durante a fase de execução (*ServiceStatus Active*), o nível de QoS acordado pode ser monitorado através de mecanismos de rede (*QoSprovided*), serviços de Middleware (*QoSmeasured*) ou sentidos físicos humanos (*QoSperceived*). Se a degradação do nível de QoS é detectado (*ServiceStatus violated*), o serviço pode ser suspenso e uma fase de renegociação é iniciada (*ServiceStatus negotiating*).

Vamos considerar um exemplo para ilustrar as funcionalidades dos componentes da arquitetura proposta. Um cliente interessado em aplicações de telefonia IP especifica seu nível de QoS desejado. Neste exemplo, o usuário necessita de "Alta Qualidade de Voz" e um "custo médio". Estes dois NFRs são conflitantes, uma vez que o usuário está interessado em um aumento da "qualidade de voz" e uma diminuição no custo. Neste caso específico, vamos supor que o usuário considera "Custo" mais restritivo, pois tem prioridade sobre a "Qualidade de voz". Esta hipótese é considerada pelo mediador de negociação durante o processo de descoberta para identificar o serviço (ou serviços) que melhor atende as necessidades dos usuários. Este mediador usa a ontologia SLA para orientar o cumprimento do contrato de QoS e da ontologia NFR para ajudar a categorização de NFR utilizado. Alinhar conceitos semanticamente equivalentes, tais como "*Quality of Speech*" e "Qualidade de Voz", em diferentes ontologias, também pode ser objeto de mediação.

É importante notar que o processo de negociação também considera requisitos de QoS exigidos pela aplicação. Com efeito, aplicações de telefonia IP requer diferentes requisitos de QoS que uma videoconferência, por exemplo, "Qualidade de Áudio" e "Qualidade de Vídeo". Além disso, é fundamental observar que um determinado serviço, prevendo diferentes níveis de QoS pode ser considerado serviços diferentes. Esta é precisamente a razão pela qual é fundamental considerar explicitamente os requisitos de QoS durante as operações *find* e *bind*.

CONCLUSÕES E TRABALHOS FUTUROS

Este artigo apresentou uma arquitetura de QoS para a negociação de recursos, cujo objetivo principal é servir de base a discussão sobre aspectos conceituais envolvidos na mediação de QoS. A arquitetura proposta baseia-se no uso de ontologias para a criação de um consenso em torno de um vocabulário de QoS.

Apesar da natureza conceitual do trabalho, o posicionamento explícito dos mediadores no contexto da negociação de QoS consiste em contribuição efetiva para o tratamento eficaz das questões de QoS em ambiente da Web Semântica. Outros benefícios da proposta incluem: a criação de um cenário onde ontologias como WSMO e OWL-S podem coexistir, as discussões sobre a integração da ontologia com a ajuda de mediação e da possível oferta de serviços personalizados que considera diretamente as necessidades dos usuários.

Atualmente, está sendo implementado um motor de mediação baseado na arquitetura proposta neste trabalho, através do framework Jena (<http://jena.sourceforge.net>) e do engenho de inferência Pellet [2]. Em termos de trabalho futuro, pretende-se investigar a comunicação entre os serviços de aplicações compostas e do papel do contrato de QoS no

contexto da composição. Mais especificamente, pretende-se estudar como as informações de QoS podem ser usadas para manter a funcionalidade e o nível de QoS acordado em aplicações dinâmicas e adaptáveis.

REFERÊNCIAS BIBLIOGRÁFICAS

1. CHUNG, L., NIXON, B. A., YU, E., MYLOPOULOS, J.: NonFunctional Requirements in Software Engineering. Ed. Kluwer Academic Publishers (2000).
2. GÓMEZ-PEREZ, A. FERNÁNDEZ-LÓPEZ, M., CORCHO, O.: Ontological Engineering: with example from the areas of Knowledge Management, e-commerce and the Semantic Web. Springer-Verlag (2004)
3. MAXIMILIEN, E. M., SINGH, M. P.: A Framework and Ontology for Dynamic Web Services Selection, IEEE Internet Computing, Volume 8, Issue 5, (2004) 84 - 93
4. ZHOU, C., CHIA, L. T., LEE, B.S.: DAML-QoS Ontology for Web Services, IEEE International Conference on Web Services (2004)
5. PAOLUCCI, M. et al.: Bringing Semantics to Web Services: The OWL-S Approach, Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, California, USA, July 6-9 (2004)
6. ROMAN, D., LAUSEN, H., KELLER, U.(eds.): WSMO Web Service Modeling Ontology. Working Draft 20 September 2004 reference, latest version available at <http://www.wsmo.org/2004/d2/v1.0/>
7. MOCAN, A., CIMPIAN, E., STOLLBERG, M., SCHARFFE, F., SCICLUNA, J. (eds.): WSMO Mediators. Working Draft 16 September 2005, latest version available at <http://www.wsmo.org/TR/d29/>
8. SCHMIDT, D. C. et al: QoS-enable Middleware. Middleware for Communications. John Wiley & sons LTD (2004) 131-162
9. RIBEIRO, C. M. F. A., ROSA, N. S., CUNHA, P.R.F.: Towards a Model for Personalized Communication Services. Proceedings of 18th International Conference on Advanced Information Networking and Applications, IEEE Computer Society Press (2004) 99-102
10. RIBEIRO, C. M. F. A., ROSA, N. S., CUNHA, P.R.F.: Meaningful SLA: Towards a QoS Contract based on user perception. Proceedings of 8th International IEEE Enterprise Distributed Object Computing Conference - Workshop on Contract Architectures and Languages - CoALa (2004)
11. SMITH, M. K., WELTY, C. MCGUINNESS, D. L. (eds.): OWL Web Ontology Language Guide. W3C Recommendation, 10 February 2004. Latest version available at <http://www.w3.org/TR/owl-guide/>
12. MEMENIOS, G., PAVLOU, G., GRIFFIN, D., GEORGIADIS, L.: Service Level Specification Semantics and Parameters. Internet Draft, tequila-sls-02 (2002)
13. CARDOSO, J., SHETH, A. P., MILLER, J. A., ARNOLD, J., KOCHUT., K. J.: Modeling Quality of Service for Workflows and Web Service Processes, Web Semantics Journal: Science, Services and Agents on the World Wide Web Journal, Vol 1(3). Elsevier (2004) 281-308
14. IAN SOMMERVILLE: Software Engineering, 6 edition. Addison Wesley (2000).