

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Composição de serviços para ampliar o
suporte à interoperabilidade semântica em
uma infraestrutura de middleware**

Frâncila Weidt Neiva

JUIZ DE FORA
AGOSTO, 2013

Composição de serviços para ampliar o suporte à interoperabilidade semântica em uma infraestrutura de middleware

FRÂNCILA WEIDT NEIVA

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: José Maria Nazar David

JUIZ DE FORA

AGOSTO, 2013

COMPOSIÇÃO DE SERVIÇOS PARA AMPLIAR O SUPORTE À
INTEROPERABILIDADE SEMÂNTICA EM UMA
INFRAESTRUTURA DE MIDDLEWARE

Frâncila Weidt Neiva

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS
EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTE-
GRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

José Maria Nazar David
Doutor em Engenharia de Sistemas e Computação

Regina Maria Maciel Braga
Doutora em Engenharia de Sistemas e Computação

Fernanda Cláudia Alves Campos
Doutora em Engenharia de Sistemas e Computação

JUIZ DE FORA
28 DE AGOSTO, 2013

Resumo

Contexto: Infraestruturas de middleware têm sido largamente utilizadas para apoiar o desenvolvimento e a execução de sistemas colaborativos. Com esses sistemas diversos desafios surgiram, entre eles os relacionados à interoperabilidade semântica ganharam notável destaque. **Objetivo:** Este trabalho tem como objetivo ampliar o suporte à interoperabilidade semântica em uma infraestrutura chamada *Web-based Groupware Service-Oriented Architecture* (WGWSOA) que utiliza serviços de middleware para apoiar o desenvolvimento e a execução de sistemas colaborativos. **Resultados:** Serviços foram construídos nessa infraestrutura para ampliar o suporte à interoperabilidade semântica em tempo de desenvolvimento e execução de sistemas colaborativos. Avaliações foram realizadas utilizando os serviços existentes na WGWSOA. **Conclusões:** Através de um estudo experimental foi possível gerar evidências que apontam para a validade da solução proposta em apoiar a construção e a execução de sistemas colaborativos.

Palavras-chave: Interoperabilidade semântica, Sistemas colaborativos, Serviços de middleware, Groupware, Ontologia, Tesauros, Arquitetura Orientada à Serviços, SOA.

Abstract

Background: Middleware infrastructure have been widely adopted to support the development and execution of collaborative systems. With these systems many challenges arise between them, those related with the interoperability subject gained remarkable importance. **Aim:** This work aims to increase the semantic interoperability support in an infrastructure named Web-based Groupware Service-Oriented Architecture (WGWSOA) that uses middleware services to support the development and execution of collaborative systems. **Results:** Services were built on this infrastructure to extend the interoperability semantic support in development and execution time of collaborative systems. **Conclusions:** Evidences pointing to the validity of the proposed solution to support the development and execution of collaborative systems were produced through an experimental study.

Keywords: Semantic Interoperability, Collaborative Systems, Middleware Services, Groupware, Ontology, Thesaurus, Service-Oriented Architecture, SOA.

Agradecimentos

A Deus por ser um pai amoroso que caridosamente me permitiu conviver com pessoas que tanto admiro e passar por desafios necessários a minha evolução.

Aos meus pais por me ensinarem o valor do conhecimento e por me fornecerem o sustentáculo de um lar, onde sempre encontrei um pouso seguro nos momentos difíceis.

Ao professor José Maria pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Ao professor Marcos Kalinowski, pelas sugestões, dicas e apoio que enriqueceram esse trabalho.

Às pessoas do NEnC (Núcleo de Engenharia do Conhecimento), em especial ao Márcio Arakaki, pela disposição em ajudar.

À Rita Suzana, Diego Arize e Thiago de Almeida pela atenção despendida ao responder dúvidas sobre a WGWSOA e propor contribuições durante o desenvolvimento deste trabalho.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

Aos amigos e colegas, pelo apoio e pelo incentivo constantes.

A todos, muito obrigada!

*“Na vida, não vale tanto o que temos,
nem tanto importa o que somos. Vale o
que realizamos com aquilo que possuímos
e, acima de tudo, importa o que fazemos
de nós.”*

Chico Xavier

Sumário

Lista de Figuras	7
Lista de Abreviações	8
1 Introdução	9
2 Fundamentação Teórica	13
2.1 Sistemas Colaborativos e <i>Middleware</i>	13
2.2 Interoperabilidade	15
2.3 WGWSOA: Uma Arquitetura Orientada a Serviços de <i>Middleware</i>	16
2.4 Interoperabilidade Semântica na WGWSOA	18
2.4.1 Ontologia	19
2.4.2 Tesouros	20
2.4.3 Limitações dos Serviços de Apoio à Interoperabilidade Semântica	21
2.5 Considerações Finais do Capítulo	22
3 Solução Proposta	23
3.1 Especificação	23
3.2 Projeto	26
3.3 Implementação	27
3.4 Testes Funcionais	32
3.5 Considerações Finais do Capítulo	33
4 Avaliação da Solução Proposta	35
4.1 Definição do Estudo Experimental	35
4.2 Planejamento do Estudo Experimental	36
4.3 Operação: Primeiro Estudo de Caso	40
4.3.1 Objetivo	40
4.3.2 Caracterização do Participante	40
4.3.3 Cenário	41
4.3.4 Fontes de Evidências	42
4.3.5 Lições Aprendidas	42
4.4 Operação: Segundo Estudo de Caso	43
4.4.1 Objetivo	43
4.4.2 Caracterização do Participante	43
4.4.3 Cenário	43
4.4.4 Fontes de Evidências	44
4.4.5 Lições Aprendidas	44
4.5 Resultado e Análise dos Dados Coletados	44
4.6 Considerações Finais do Capítulo	47
5 Conclusões	48
Referências Bibliográficas	50

A	APÊNDICE - Questionário de Caracterização do Participante	52
B	APÊNDICE - Questionário de Avaliação do Estudo de Caso	54
C	APÊNDICE - Termo de Consentimento Livre e Esclarecido	58
D	APÊNDICE - Protótipos da Funcionalidade SemanticMediationSupport-View	61

Lista de Figuras

2.1	Arquitetura da WGWSOA	17
2.2	Estrutura do Tesauros na WGWSOA - Retirado de (Arize et al., 2013) . .	21
3.1	Diagrama de Casos de Uso da Solução Proposta.	25
3.2	Diagrama de Componentes da Solução Proposta.	27
3.3	Seleção dos Arquivos com a Descrição dos Serviços Candidatos à Interoperabilidade Semântica.	28
3.4	Relatório Geral da Interoperabilidade Semântica entre os Serviços Candidatos.	29
3.5	Detalhamento da interoperabilidade semântica entre termos.	30
3.6	Material de Auxílio sobre como Desenvolver na WGWSOA.	31
3.7	Log de Monitoramento da Interoperabilidade Semântica.	31
3.8	Quadro de Casos de Teste do Componente SemanticMediationSupportView	33
3.9	Quadro de Casos de Teste do Componente SemanticExecutionSupport . . .	33
4.1	Etapas da Fase de Planejamento de um Estudo Experimental	39
4.2	Resultados Obtidos Através dos Dados Coletados no Estudo Experimental.	46
4.3	Comparativo dos Resultados Obtidos por Estudo de Caso.	47
D.1	Verificação da Interoperabilidade Semântica	61
D.2	Dados da interoperabilidade semântica	62
D.3	Dados da Interoperabilidade Semântica	62

Lista de Abreviações

API	<i>Application Programming Interface</i>
Aw2SOA	<i>Awareness to Service-Oriented Architecture</i>
CORBA	<i>Common Object Request Broker Architecture</i>
GQM	<i>Goal Question Metric</i>
LPS	<i>Linha de Produto de Software</i>
HLA OMT	<i>HLA Object Model Template</i>
OWL	<i>Web Ontology Language</i>
RDF	<i>Resource Description Framework</i>
RMI	<i>Remote Method Invocation</i>
SOA	<i>Service-Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
SVN	<i>Subversion</i>
ThesIS	<i>Thesaurus-based Interoperability Service</i>
UML	<i>Unified Modeling Language</i>
XML	<i>eXtensible Markup Language</i>
WGWSOA	<i>Web-based Groupware Service-Oriented Architecture</i>

1 Introdução

O software desempenha um importante papel em diversos setores da sociedade. Essa disseminação do uso do software para diferentes fins e sua dependência para executar tarefas cotidianas geraram a necessidade de se buscar soluções para aumentar a produtividade, diminuir os custos e aumentar a qualidade nos processos de desenvolvimento. Buscando suprir essas necessidades, as organizações que desenvolvem software passaram a distribuir suas atividades geograficamente. Todavia, com os grupos de trabalho localizados em sítios distintos a colaboração se tornou mais complexa. Diante desse novo cenário, se faz necessário desenvolver sistemas que viabilizem essa forma de colaboração.

Segundo Fuks et al. (2005) a colaboração pode ser entendida como resultado da combinação de três elementos: (i) comunicação - diz respeito à troca de mensagens advindas de interações entre os usuários; (ii) cooperação - gerenciamento de pessoas, suas atividades e seus recursos e; (iii) coordenação - trabalho realizado em um espaço de trabalho compartilhado.

Oferecer um suporte adequado a esses elementos em sistemas colaborativos a fim de promover uma colaboração eficiente não é uma tarefa trivial. Contribui para isso o fato de que nesse contexto, as interações e os processos nos quais elas ocorrem impõem muitas vezes políticas e regras flexíveis. Por exemplo, imagine o cenário em que uma lista de discussão está sendo usada para apoiar a tomada de decisões. Neste cenário, a partir do momento em que decisões são tomadas, papéis são modificados, novas atividades são definidas e novas políticas organizacionais são criadas, demandando muitas vezes, modificações imediatas. Outro fato que aumenta a complexidade desses sistemas é a dinamicidade dos componentes utilizados para representar o contexto da interação. Ou seja, tomando como exemplo o mesmo cenário anterior, durante a tomada de decisões, modificações no contexto ocorrem ao longo da interação, gerando aos participantes novas necessidades e modificando seus objetos de interesse. Logo, devem ser criados mecanismos para dar suporte a essas atividades, além de evoluir com a mesma agilidade de dinamicidade dos contextos nos quais elas estão inseridas (David et al., 2009).

A arquitetura orientada a serviço (SOA) foi um fator importante para facilitar a construção de sistemas colaborativos. A arquitetura foi proposta como solução para viabilizar a construção de aplicações distribuídas, propondo uma maneira lógica para projetar um sistema de software em que serviços são fornecidos às aplicações de usuários finais ou a outros serviços distribuídos (Papazoglou et al., 2007). Para Erl (2008) a SOA estabelece um modelo arquitetônico cujo objetivo é aprimorar a eficiência, agilidade e a produtividade no desenvolvimento de sistemas, posicionando os serviços como principal meio para atingir tais objetivos. Diferentes tecnologias podem se basear em SOA ou seja, uma implementação dessa arquitetura pode consistir em uma combinação de produtos, tecnologias e várias outras partes.

Os serviços em SOA obedecem a alguns princípios que vão ao encontro de requisitos de qualidade demandados por sistemas colaborativos. São eles: baixo acoplamento, capacidade de reuso, capacidade de composição, abstração de serviços e ausência de estado. O atendimento a esses requisitos propicia um ambiente favorável para atingir a flexibilidade necessária para que os serviços construídos segundo o conceito SOA sejam capazes de se adequarem rapidamente às modificações no contexto das interações (Erl, 2008).

O caráter distribuído dos sistemas colaborativos, faz com que a interoperabilidade se torne um requisito fundamental para a promoção da colaboração. Neste contexto, interoperabilidade pode ser definida como a capacidade de aplicações construídas sobre diferentes plataformas conseguirem se comunicar e trocar dados (Bernstein, 1996). Entretanto, tecnologias que aplicam os conceitos de SOA geralmente se baseiam em padrões XML, estes padrões permitem apenas que aplicações distribuídas interoperem em um nível sintático isto é, trata apenas do reconhecimento de padrões de comunicação sem agregar valor semântico ao seu conteúdo (Almeida, 2013). Porém, para uma colaboração satisfatória isso não é suficiente, sendo assim, é necessário procurar meios de oferecer um suporte mais amplo à questão da interoperabilidade.

Outra solução frequentemente adotada para o desenvolvimento de aplicações distribuídas de sistemas colaborativos são os *middlewares* (David et al., 2009). Eles são capazes de resolver problemas relacionados à dificuldades de conexão e interoperabilidade

em aplicações heterogêneas (Schantz et al., 2001). Segundo esses autores o *middleware* é um sistema de software que se situa entre as aplicações e os sistemas operacionais subjacentes, pilhas de protocolo de rede e hardware. Além disso, oferece e disponibiliza um conjunto de serviços através de uma interface bem definida e de maneira transparente para as aplicações (Maciel et al., 2007). Todavia, apesar dessas aplicações construídas em plataformas de *middleware* serem capazes de manipular objetos de semântica equivalente, elas não conseguem acessar dados e objetos de outra aplicação, mesmo quando construídas sob uma mesma plataforma (David et al., 2009).

Com o objetivo de solucionar desafios relacionados à interoperabilidade em sistemas colaborativos foi definida uma arquitetura orientada a serviços de *middleware* chamada *Web-based Groupware Service-Oriented Architecture* (WGWSOA) que une os benefícios do *middleware* e da SOA. A infraestrutura de *middleware* WGWSOA (Maciel et al., 2007) provê serviços para apoiar o desenvolvimento e a execução de sistemas colaborativos que são escaláveis e podem ser compostos.

O suporte a interoperabilidade sintática na WGWSOA está limitado às diferentes tecnologias utilizadas pelas plataformas. A interoperabilidade semântica foi atingida através da construção de dois serviços: um baseado no uso de ontologias (Almeida et al., 2011) e outro que utiliza o conceito de tesauros (Arize et al., 2013). Todavia esses serviços não estão disponibilizados na infraestrutura de maneira que apoiem facilmente o desenvolvedor e o usuário que utilizam a WGWSOA. Por exemplo, interfaces não são oferecidas para apoiar a seleção de serviços que irão interoperar em tempo de desenvolvimento, nem tampouco a infraestrutura apoia o monitoramento da interoperabilidade em tempo de execução.

Desta maneira, o objetivo deste trabalho é ampliar o suporte à interoperabilidade semântica em tempo de desenvolvimento e execução de sistemas colaborativos através da construção de serviços que sejam capazes de compor com os serviços já existentes na WGWSOA que apoiam esse tipo de interoperabilidade. Através desta composição de serviços, busca-se facilitar a realização de atividades que demandem interoperabilidade semântica, por exemplo, abstraindo complexidades de programação atualmente necessárias para o uso desses serviços.

A monografia está organizada da seguinte forma: no segundo capítulo será apresentada a fundamentação teórica visando abordar o estado da arte das pesquisas relacionadas ao desenvolvimento de sistemas colaborativos. Neste capítulo são ressaltadas as dificuldades inerentes a este tipo de sistema, em especial no que se refere à interoperabilidade. Finalmente, são discutidas propostas para a utilização do *middleware* WGWSOA para apoiar o desenvolvimento de sistemas colaborativos. No terceiro capítulo será apresentada a solução proposta para apoiar a interoperabilidade no contexto desta pesquisa. As etapas de especificação, projeto, implementação e testes funcionais do projeto são abordadas e discutidas. No quarto capítulo é feita uma avaliação da proposta de solução. São abordadas as hipóteses, as questões levantadas durante o processo de pesquisa e o planejamento do estudo experimental usado para avaliação, bem como sua realização e a análise dos resultados obtidos. No quinto capítulo são apresentadas as conclusões do trabalho. Esse capítulo compreende um breve resumo da solução proposta, as contribuições obtidas, as dificuldades enfrentadas durante a pesquisa e os trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo será apresentada a fundamentação teórica relacionada às pesquisas em sistemas colaborativos tendo como ênfase: os benefícios e limitações do uso de *middleware* para auxiliar o desenvolvimento de sistemas colaborativos, a importância da interoperabilidade nesses sistemas e as soluções em interoperabilidade semântica oferecidas pela WGWGOA.

2.1 Sistemas Colaborativos e *Middleware*

Para lidar com a crescente demanda por produtividade, eficiência e qualidade, atividades complexas que até então eram realizadas de maneira isolada e individual passaram a ser feitas de forma colaborativa. Essa colaboração, frequentemente ocorre através de grupos geograficamente distribuídos e isto requer interações remotas complexas. As aplicações colaborativas distribuídas são comumente construídas sobre diferentes plataformas e tecnologias e isto faz com que ambientes colaborativos não sejam naturalmente interoperáveis, por isso, promover interoperabilidade nesses casos geralmente demanda esforço e custos elevados. Por exemplo, não são apenas questões relacionadas à incompatibilidade das plataformas de hardware e software que devem ser solucionadas, é também necessário lidar com uma série de soluções específicas relacionadas a protocolos, APIs (*Application Programming Interface*), componentes, entre outros (Maciel et al., 2009).

O uso do *middleware* foi motivado por essa necessidade de oferecer um suporte mais avançado e eficaz, que vai além da simples conectividade, na construção de sistemas colaborativos eficazes. O principal objetivo do *middleware* é coordenar como diferentes partes de uma aplicação distribuída estão conectadas e como essas partes interoperam. Além disso, o uso do *middleware* simplifica a integração de componentes desenvolvidos em plataformas heterogêneas (Schantz et al., 2001).

O *middleware* oferece serviços que possuem um propósito geral e se situam entre as plataformas e as aplicações. Os serviços de *middleware* permitem que APIs se tornem

independentes da plataforma, logo essas aplicações são capazes de serem executadas em diversas plataformas. Eles também oferecem serviços de alto nível que mascaram grande parte da complexidade de redes e sistemas distribuídos. Os serviços de *middleware* surgem da necessidade de expor funcionalidades de necessidade comum às aplicações de diversos domínios em componentes independentes, para serem compartilhados através de diferentes plataformas e ambientes (Berstein, 1993).

Entretanto, algumas complexidades de sistemas colaborativos não são resolvidas através do uso de *middleware*. Os serviços de *middleware* são muitas vezes parte de um ambiente organizacional que se modifica. Uma vez que às organizações crescem e se integram à outras organizações, suas políticas e processos de negócio evoluem, se unificam e tornam-se mais abrangentes. Assim, os serviços também passam a ser parte de um cenário mais abrangente e dificuldades relacionadas à falta de padrões entre as organizações podem se tornar evidentes. Por exemplo, muitas vezes serviços de *middleware* utilizam APIs e protocolos proprietários, o que dificulta a construção de implementações interoperáveis. Em outros casos, essas APIs e protocolos não estão disponíveis para uso em plataformas importantes, limitando assim a capacidade de se conectar sistemas heterogêneos (Fuks and Pimentel, 2012).

Outro desafio surge em relação ao número de serviços de *middleware* disponibilizados, pois um número muito alto pode comprometer a utilização dos mesmos, uma vez que o gerenciamento desses serviços pode se tornar muito complexo. Logo, é importante que o desenvolvedor selecione um número pequeno de serviços mas, esses serviços devem cobrir de maneira adequada as funcionalidades desejadas. Além disso, apesar dos serviços de *middleware* aumentarem a abstração da programação em aplicações distribuídas, esses serviços ainda delegam ao desenvolvedor às principais decisões de projeto. Por exemplo, o desenvolvedor continua tendo que decidir qual funcionalidade colocará no lado do cliente e qual colocará no lado do servidor (Bernstein, 1996).

O uso da arquitetura orientada a serviços sobre a camada de serviços de *middleware* surge como uma possibilidade para diminuir a complexidade do desenvolvimento em sistemas colaborativos. A introdução dessa arquitetura evita que situações como: aplicativos desenvolvidos sem levar em consideração atributos de interoperabilidade, resultem

em um labirinto de arquiteturas de integração complicadas quando se desejar compartilhar dados entre esses aplicativos e que; aplicativos desenvolvidos baseados em tecnologias ou plataformas diferentes e que muitas vezes possuem requisitos arquitetônicos únicos gerem um ambiente incompatível. Além disso, o uso de SOA fornece uma interoperabilidade inerente, uma vez que os serviços são projetados para serem intrinsecamente interoperáveis. Porém, essa interoperabilidade não se dá em todos os níveis que são necessários para promover uma colaboração satisfatória em um ambiente de sistemas colaborativos (Erl, 2008).

2.2 Interoperabilidade

A interoperabilidade pode ser entendida como a propriedade de dois ou mais sistemas interagirem e trocarem dados, baseando-se para isso em métodos definidos, com a finalidade de obterem os resultados esperados. Na área da computação, o conceito de interoperabilidade se difere do conceito de integração. Enquanto integração trata da capacidade de unir em um portal diferentes ambientes, a interoperabilidade diz respeito a capacidade de operação e cooperação entre sistemas diferentes (Mucheroni and Silva, 2011).

Para Tolk et al. (2007) a interoperabilidade pode ser dividida em sete níveis distintos: (i) sem interoperabilidade - neste nível estão incluídos os sistemas *stand-alone*; (ii) interoperabilidade técnica - neste nível existe um protocolo de comunicação para permitir a troca de dados entre sistemas; (iii) interoperabilidade sintática - neste nível é introduzida uma estrutura comum para a troca de informação; (iv) interoperabilidade semântica - alcançada a partir do uso de um modelo de referência padrão para a troca de informação, neste nível o significado dos dados é compartilhado de forma não ambígua; (v) interoperabilidade pragmática - neste nível o uso dos dados é compreendido pelos sistemas participantes e o contexto em que esses dados são trocados está definido de maneira não ambígua; (vi) interoperabilidade dinâmica - neste nível o sistema é capaz de compreender as mudanças de estado que ocorrem nas premissas e restrições ao longo do tempo e tirar vantagem dessa situação; (vii) interoperabilidade conceitual - este é o nível mais alto e ocorre quando o modelo conceitual está alinhado. Isto é, quando as premissas e restrições geradas a partir da abstração da semântica do mundo real estão alinhadas.

Para o contexto deste trabalho, os níveis sintático, semântico e pragmático serão destacados. Para atingir a interoperabilidade sintática, algumas tecnologias surgiram com o objetivo de propor uma estrutura padronizada para a troca de dados, entre elas as mais comuns são: *eXtensible Markup Language* (XML), *HLA Object Model Template* (HLA OMT), *Interface Description Language*, *Common Object Request Broker Architecture* (CORBA) e *Simple Object Access Protocol* (SOAP) (Wang et al., 2009).

Contudo, dados interoperáveis são muitas vezes insuficientes para uma interoperabilidade satisfatória em aplicações de software, principalmente em se tratando de contextos colaborativos. Portanto, é necessário transformar os dados em informação e com isso, atender a interoperabilidade em um nível semântico. A interoperabilidade semântica pode ser entendida como a habilidade de sistemas trocarem informações, cujo significado é capaz de ser inferido, interpretado e classificado, sem que para isso haja envolvimento humano (Pollock and Hodgson, 2004). Para atingir este nível de interoperabilidade, metadados sobre o conjunto de dados relacionado são fornecidos. Para cumprir esses objetivos, algumas tecnologias se destacaram como: *Web Ontology Language* (OWL), *Unified Modeling Language* (UML) e *Resource Description Framework* (RDF) (Li et al., 2004).

A interoperabilidade pragmática vai além do nível semântico, pois busca assegurar que os envolvidos na troca de mensagens, além de possuírem um mesmo entendimento sobre o significado dos dados, possuam uma mesma expectativa em relação aos efeitos causados por esta troca. Isto é, as mensagens enviadas sempre possuem uma intenção, ou seja, desejam causar algum efeito em quem as recebe. Porém, nem sempre o efeito desejado se traduz no efeito causado, desta forma, a interoperabilidade pragmática surge para resolver este tipo de conflito (Pokraev et al., 2005).

2.3 WGWSOA: Uma Arquitetura Orientada a Serviços de *Middleware*

Com a finalidade de solucionar problemas relacionados à interoperabilidade em sistemas colaborativos, construídos sobre uma mesma plataforma ou sobre plataformas distintas, foi desenvolvida uma arquitetura orientada a serviços de *middleware* chamada WGWSOA

(*Web-based Groupware Service-Oriented Architecture*) (Maciel et al., 2007).

A WGWSOA foi projetada para apoiar o desenvolvimento e a execução de sistemas colaborativos e busca atender a requisitos comumente necessários nesse domínio através de uma arquitetura orientada a serviços. Esses requisitos são: flexibilidade, portabilidade, escalabilidade, capacidade de reutilização e composição dos serviços (Almeida et al., 2011).

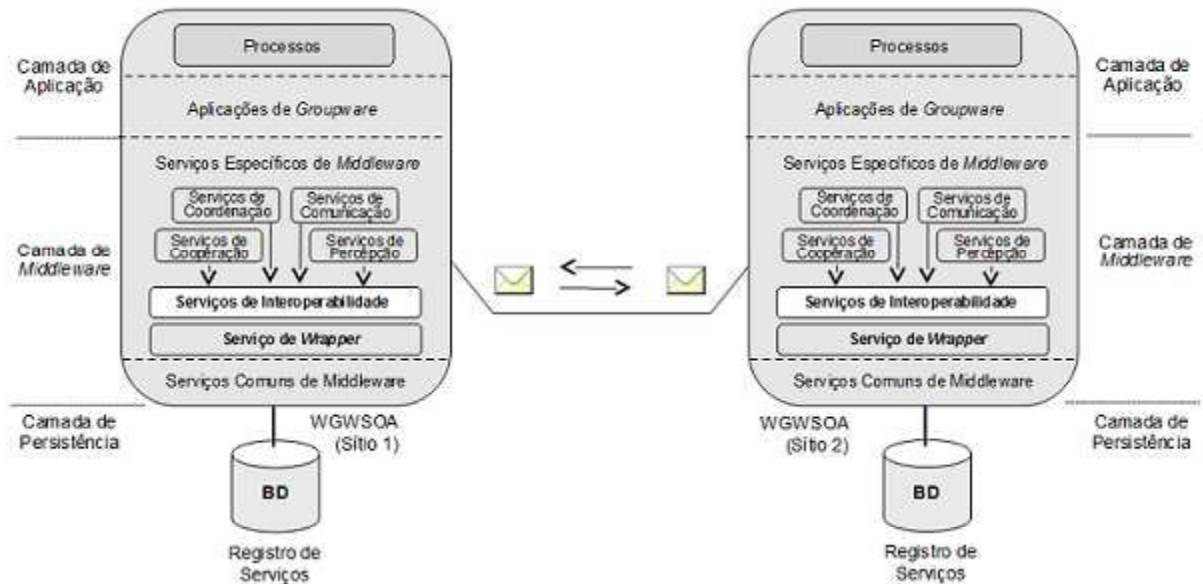


Figura 2.1: Arquitetura da WGWSOA

A Figura 2.1 apresenta uma visão geral da arquitetura de funcionamento da WGWSOA. A arquitetura pode ser dividida em três camadas: aplicação, *middleware* e persistência. Na primeira camada encontram-se as aplicações de *groupware* que utilizam serviços da WGWSOA. A segunda camada é composta por serviços comuns e específicos. Os serviços comuns implementam as funcionalidades necessárias à diversas aplicações distribuídas, já os serviços específicos implementam as funcionalidades que serão usadas por um domínio determinado, neste caso, o domínio de sistemas colaborativos. Finalmente, na camada de persistência encontram-se descrições dos serviços armazenados em repositórios, essas descrições auxiliam na localização dos serviços (Almeida et al., 2011).

Os serviços específicos de *middleware* oferecidos pela infraestrutura podem ser categorizados como sendo de coordenação, comunicação, cooperação, percepção, Wrapper e interoperabilidade semântica: (i) os serviços de coordenação são úteis para apoiar ativi-

dades de coordenação em grupos que utilizam sistemas colaborativos heterogêneos. Esses serviços foram projetados para permitir a construção de aplicações que apoiem atividades de discussão e a tomada de decisões. (ii) Os serviços de comunicação podem ser subdivididos em síncronos e assíncronos e tem como principal objetivo apoiar diferentes modos de colaboração através da composição com outros serviços. (iii) Os serviços de cooperação foram projetados para apoiar atividades de avaliação dos serviços de comunicação, coordenação e percepção. (iv) O serviço de percepção na infraestrutura WGWSOA é conhecido como Aw2SOA (*Awareness to Service-Oriented Architecture*) e tem como principal objetivo apoiar a percepção sobre ações ocorridas nas aplicações e nos serviços da WGWSOA, extraindo informações como: o que, onde, quem e quando as ações ocorreram. (v) O serviço de Wrapper na infraestrutura é responsável pela comunicação entre instâncias distintas da WGWSOA. (vi) E finalmente, os serviços de interoperabilidade que incluem suporte aos níveis de interoperabilidade sintática, semântica e pragmática (David et al., 2009).

2.4 Interoperabilidade Semântica na WGWSOA

No contexto da WGWSOA, será utilizado o conceito de interoperabilidade semântica definido por (Pollock and Hodgson, 2004). Para que os sistemas colaborativos estejam aptos a promover a interoperabilidade semântica, devem ser capazes de (i) relacionar conceitos que possuam o mesmo significado e (ii) interpretar e classificar esse significado (Almeida et al., 2011).

Em sistemas colaborativos o significado dos conceitos está ligado ao contexto no qual está inserido. É importante que técnicas de representação do contexto o façam de forma adequada e sejam capazes de se adaptar para responder as evoluções do mesmo. Desta forma, aplicações colaborativas que visam interoperar em um nível semântico devem ser construídas com a capacidade de apresentar, armazenar e reconhecer a semântica referente ao serviço que irá interagir na comunicação (Almeida, 2013).

Através dos serviços oferecidos pela infraestrutura, a interoperabilidade semântica tem sido tratada através de duas abordagens: ontologias (Almeida et al., 2011) e tesauros (Arize et al., 2013).

2.4.1 Ontologia

Segundo Berners-Lee (2001), é possível definir ontologia como uma taxonomia associada a um conjunto de regras de inferência. As taxonomias classificam os objetos (entidades) e determinam os relacionamentos entre eles. Por exemplo, o objeto endereço pode ser classificado como uma localização que por sua vez, está relacionada a um código postal. Desta forma, podemos usar a ontologia para representar de forma esquematizada um domínio do conhecimento.

Na infraestrutura WGWSOA, a ontologia pode ser usada para descrever os serviços disponíveis. Através dessas descrições um serviço mediador associado a essas ontologias chamado *OntologyMediator*, busca apoiar a interoperabilidade semântica entre aplicações colaborativas situadas em diferentes instâncias da infraestrutura.

As ontologias usadas por esse serviço são construídas em um arquivo OWL (*Web Ontology Language*). Esse arquivo é gerado através de um editor de ontologias chamado *Protégé*¹. Outros editores e linguagens de descrição de ontologias não são considerados pelo serviço.

O uso do serviço *OntologyMediator* busca atender a cenários em que dados desejam interoperar e para isso necessitam de uma avaliação sobre seus significados. O serviço tem como objetivo tratar os conflitos semânticos existentes durante a interoperabilidade semântica entre dados de aplicações de instâncias distintas da WGWSOA e para isso, representa a semântica do contexto no qual os dados estão inseridos através de ontologias (Almeida, 2013).

O *OntologyMediator* faz uso da composição com outros serviços para atingir seus objetivos, entre seus componentes se incluem: (i) um serviço mediador - principal responsável pelo alinhamento dos dados; (ii) um serviço de consulta à base de termos (*WordNet*) - provê sinônimos e hiperônimos necessários ao alinhamento de dados; e (iii) um serviço que analisa e manipula as ontologias dos respectivos dados que irão interoperar (Almeida et al., 2011).

Através do uso do *OntologyMediator*, o desenvolvedor de sistemas colaborativos têm facilitada sua tarefa de manipulação de ontologias dos serviços que desejam intero-

¹<http://protege.stanford.edu/>

perar e sua tarefa de fornecer conceitos necessários ao alinhamento dos dados e execução da operação de alinhamento (Almeida, 2013).

2.4.2 Tesouros

O Tesouros pode ser definido em termos da sua função e sua estrutura. Considerando sua função, o tesouros pode ser entendido como uma ferramenta para controle terminológico usada para traduzir a linguagem natural presente por exemplo, em documentos, em uma linguagem mais sistêmica. Em relação a sua estrutura, o tesouros é um vocabulário dinâmico e controlado de termos com valor semântico agregado (ISO 2788, 1986).

Visando prover uma nova forma de suporte à interoperabilidade semântica entre aplicações localizadas em instâncias distintas da WGWSOA, foi desenvolvido um serviço baseado no uso do tesouros, chamado ThesIS (*Thesaurus-based Interoperability Service*).

O serviço ThesIS representa uma nova forma de apoiar à interoperabilidade semântica, além do serviço OntologyMediator. O serviço tem como atrativo uma maior simplicidade e flexibilidade se comparado ao OntologyMediator. Ou seja, é capaz de oferecer uma estrutura mais simples de representação da semântica dos serviços (Arize et al., 2013).

Para conseguir prover a interoperabilidade em tempo de desenvolvimento e execução de sistemas colaborativos, o ThesIS implementa três requisitos: (i) buscar conceitos que são candidatos à interoperar; (ii) realizar a correspondência semântica de conceitos sinônimos e; (iii) traduzir conceitos, caso a correspondência de conceitos tenha um retorno positivo (Arize et al., 2013).

A Figura 2.2 mostra como é feita a representação conceitual de um serviço e seus métodos no ThesIS. Essa representação se dá através de um arquivo XML (eXtensible Markup Language). A estrutura do XML deve obedecer à seguinte hierarquia: Uma tag mais externa <thesaurus>, que irá englobar a descrição do serviço. A tag <service> limita os dados do serviço em questão, que é identificado pelo seu nome através da tag <name>, por exemplo, Chat. A tag <classification> classifica o serviço através de verbos usando a tag <verb> e substantivos usando a tag <nouns>. Além disso, sinônimos relacionados ao nome do serviço podem ser descritos em <synonyms>. Cada método do serviço,

```

<thesaurus>
  <service><name>
    Chat    <classification><verb>communicate</verb>
            <nouns>conversation,chitchat.</nouns>
          </classification>
          <synonyms>confabulate,          chatter.</synonyms>

          </name>
          <methodList><method>
            <name>
              getMessages    <classification><verb>get</verb>
                             <nouns>exchange,communication.</nouns>
              </classification>
              <synonyms>obtainMessages,requestMessages.</synonyms>
            </name>
            <field>
              chatRoomId<synonyms>roomId,roomName.</synonyms>
            </field>
            </method>...</methodList>
          </service>
        </thesaurus>

```

Figura 2.2: Estrutura do Tesauros na WGWSOA - Retirado de (Arize et al., 2013)

representado por `<method>` é colocado internamente a tag `<methodList>`. Os métodos podem ser identificados e descritos através das mesmas tags utilizadas pelo serviço.

De maneira geral, tesauros quando comparado com ontologias, apresenta normalmente um número menor de elementos e de relacionamentos, resultando em uma estrutura mais simples e flexível. Desta forma, o serviço ThesIS pode tornar o desenvolvimento de aplicações semanticamente interoperáveis mais ágil, além de prover a flexibilidade necessária devido a dinamicidade dos elementos em ambientes colaborativos (Arize et al., 2013).

2.4.3 Limitações dos Serviços de Apoio à Interoperabilidade Semântica

Os serviços criados para apoiar a interoperabilidade semântica estão dispostos na WGWSOA de forma totalmente independente, isto é, não existe um serviço que ofereça um uso combinado destes serviços. Além disso, o uso do ThesIS e do OntologyMediator exigem esforços de programação para serem usados para apoiar as atividades que demandam interoperabilidade semântica tanto em tempo de execução como em tempo de desenvolvimento. As informações sobre como os resultados foram obtidos através desses serviços não são organizadas e apresentadas ao desenvolvedor e usuário que os utiliza. Portanto,

é necessário ampliar o suporte a interoperabilidade semântica em tempo de execução e desenvolvimento na WGWSOA, de modo que o usuário e desenvolvedor da WGWSOA tenham um meio de utilizar o ThesIS e OntologyMediator com um maior nível de abstração, diminuindo as complexidades referentes ao esforço de programação para usá-los e combiná-los e também oferecer acesso de forma organizada às informações que justifiquem como cada serviço chegou aos resultados obtidos.

2.5 Considerações Finais do Capítulo

Neste capítulo foram apresentados alguns conceitos que justificam a complexidade de sistemas colaborativos, bem como o papel do *middleware* para mitigar problemas relacionados à interoperabilidade em sistemas colaborativos. Além disso, foram exploradas as vantagens pertinentes à associação de *middleware* com uma arquitetura baseada em SOA. Foram também mostrados os aspectos gerais da arquitetura orientada a serviços de *middleware* WGWSOA e suas soluções oferecidas no tocante ao suporte da interoperabilidade semântica, como os serviços: ThesIS e OntologyMediator. As limitações existentes na maneira em que a interoperabilidade semântica está sendo tratada, não oferecendo meios de combinar e compor esses serviços de uma maneira que abstraia complexidades de programação ao desenvolvedor e usuário da WGWSOA, serviram de motivação para este trabalho.

O capítulo seguinte apresenta a solução proposta para ampliar o suporte à interoperabilidade semântica na WGWSOA através do detalhamento de todo processo de desenvolvimento, que inclui as seguintes fases: especificação, projeto, implementação e testes funcionais.

3 Solução Proposta

Neste capítulo serão apresentados as etapas de especificação, projeto, implementação e testes da solução proposta. A solução visa ampliar o suporte a interoperabilidade semântica na WGWSOA.

3.1 Especificação

Com o objetivo de tornar mais simples e ágil para o desenvolvedor e o usuário a tarefa de construir e gerenciar serviços semanticamente interoperáveis na infraestrutura WGWSOA, foram elicitados os seguintes requisitos funcionais:

- RF1: A WGWSOA deve oferecer ao desenvolvedor um mecanismo para selecionar os arquivos contendo a descrição semântica dos serviços candidatos a interoperabilidade: O desenvolvedor deve ter a sua disposição um meio para selecionar tais arquivos de maneira automatizada.
- RF2: A WGWSOA deve oferecer ao usuário um meio para monitorar a interoperabilidade semântica durante o uso de serviços da infraestrutura: Esse monitoramento oferece subsídios para que o usuário possa tomar decisões sobre melhorias na descrição semântica dos serviços.
- RF3: A WGWSOA deve oferecer um meio do desenvolvedor e do usuário utilizarem os serviços *OntologyMediator* e *ThesIS* já existentes na infraestrutura para apoiar a interoperabilidade semântica: A composição dos serviços *OntologyMediator* e *ThesIS* não está disponível. Ou seja, para utilizar esses serviços e combinar seu uso, é necessário esforço de programação, o que dificulta seu uso por tornar a atividade mais demorada e complexa.
- RF4: A WGWSOA deve oferecer ao desenvolvedor e ao usuário opções de visualização detalhadas sobre os resultados obtidos através do uso *OntologyMediator* e do *ThesIS*: É importante que o desenvolvedor e o usuário saibam mais que uma resposta

afirmativa ou não sobre a interoperabilidade semântica entre serviços distintos, eles devem saber como esse resultado foi atingido.

- RF5: A WGWSOA deve oferecer ao desenvolvedor mecanismos para registrar sua concordância ou não com os resultados obtidos através do uso do *OntologyMediator* e do *ThesIS*: Registrar o *feedback* do desenvolvedor sobre os resultados retornados fornece dados importantes para análise dos serviços de suporte à interoperabilidade semântica oferecidos pela infraestrutura.
- RF6: A WGWSOA deve oferecer ao desenvolvedor acesso a informações sobre como desenvolver na infraestrutura: É importante que os desenvolvedores da WGWSOA tenham acesso à tutorias de criação de serviços, criação de aplicações, configuração de ambiente, entre outros. Pois, além de criar padrões e desta forma encapsular as melhores práticas de desenvolvimento, isso facilita a inserção de novos desenvolvedores ao documentar e disponibilizar os processos de desenvolvimento.

Além dos requisitos funcionais, a solução proposta deve atender aos seguintes requisitos não funcionais:

- RN1 - Escalabilidade: É importante que a solução seja capaz de se adaptar à novos serviços relativos a interoperabilidade semântica que possam vir a ser implementados na infraestrutura. Isto é, atualmente a WGWSOA conta com dois serviços de interoperabilidade semântica, um baseado em ontologias e outro baseado em tesau-ros. No entanto, é provável que novos serviços baseados em outras técnicas surjam e por isso a solução deve ser escalável para conseguir se adaptar.
- RN2 - Reusabilidade: É importante que a solução possa ser reusada por outros serviços que queiram utilizar sua lógica. O reuso é um atributo de qualidade que deve ser implementado por todos os serviços disponíveis na WGWSOA que permite, entre outras coisas, a criação ágil de soluções mais completas e complexas através da composição com outros serviços.
- RN3 - Capacidade de composição: A solução deve ser capaz de compor com os serviços de apoio interoperabilidade semântica existentes na WGWSOA, com a fi-

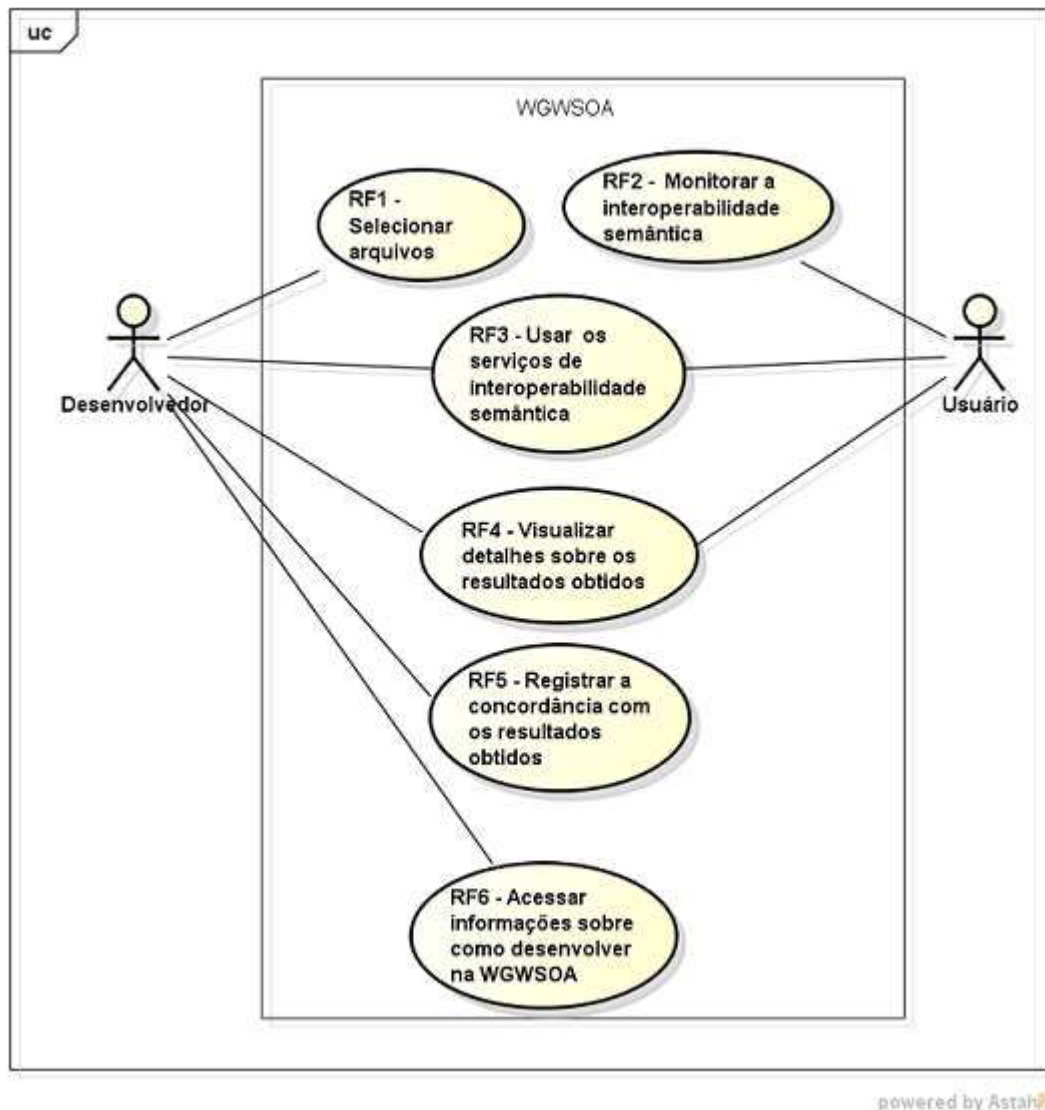


Figura 3.1: Diagrama de Casos de Uso da Solução Proposta.

nalidade de abstrair a complexidade do uso desses serviços, diminuindo ao máximo o esforço de programação para utilizá-los.

Existem outros requisitos não funcionais importantes para uma solução inserida em um contexto de sistemas colaborativos, como por exemplo: baixo acoplamento e ausência de estado. Porém, esses requisitos não fazem parte do escopo desta solução, sendo assim, não serão considerados ou tampouco avaliados.

3.2 Projeto

Baseado nos requisitos funcionais e não funcionais apontados na seção anterior, foi elaborado um projeto que compreende a divisão e associação entre os componentes que formam a solução proposta. Como pode ser visto pela Figura 3.2 os componentes que compõem a solução são: (i) Wrapper - promovendo a interoperabilidade sintática da solução. (ii) OntologyMediator - serviço de interoperabilidade semântica baseado no uso de ontologias. (iii) ThesIS - serviço de interoperabilidade semântica baseado no uso do tesauros. (iv) WordNetHandler - funcionalidade de consulta à WordNet. (v) SemanticMediationSupport - serviço que obtém resultados da interoperabilidade semântica através da composição com os serviços OntologyMediator e ThesIS. (vi) SemanticExecutionSupport - serviço de apoio a interoperabilidade semântica em tempo de execução. (vii) SemanticMediationSupportView - oferece interfaces de usuário para apoiar a interoperabilidade semântica em tempo de desenvolvimento. Os serviços de apoio a interoperabilidade semântica SemanticMediationSupport, SemanticExecutionSupport e a interface de usuário SemanticMediationSupportView, diferente dos outros componentes não existiam na WGWSOA e foram implementados neste trabalho.

Os requisitos não funcionais influenciaram nas decisões de projeto. Para atingir a escalabilidade, o componente SemanticMediationSupport foi criado com alto grau de independência funcional. O SemanticMediationSupport tem como objetivo ser capaz de apoiar novos serviços de interoperabilidade semântica que venham a ser desenvolvidos, através da criação de novas composições. A decisão de criação de dois serviços, SemanticMediationSupport e SemanticExecutionSupport, garantiram uma maior modularização da solução potencializando sua capacidade de reúso. A capacidade de composição buscou ser atendida através da funcionalidade SemanticMediationSupportView e do serviço SemanticExecutionSupport, pois foram pensados para usarem o ThesIS e o OntologyMediator de forma a abstrair a complexidade de programação presente nesses serviços.

Conforme representado na Figura 3.2, a associação dos componentes da solução proposta ocorre da seguinte forma: para apoiar a interoperabilidade semântica em tempo de execução, o serviço SemanticExecutionSupport é chamado via Wrapper. Em tempo de desenvolvimento, a interface de usuário SemanticMediationSupportView é executada

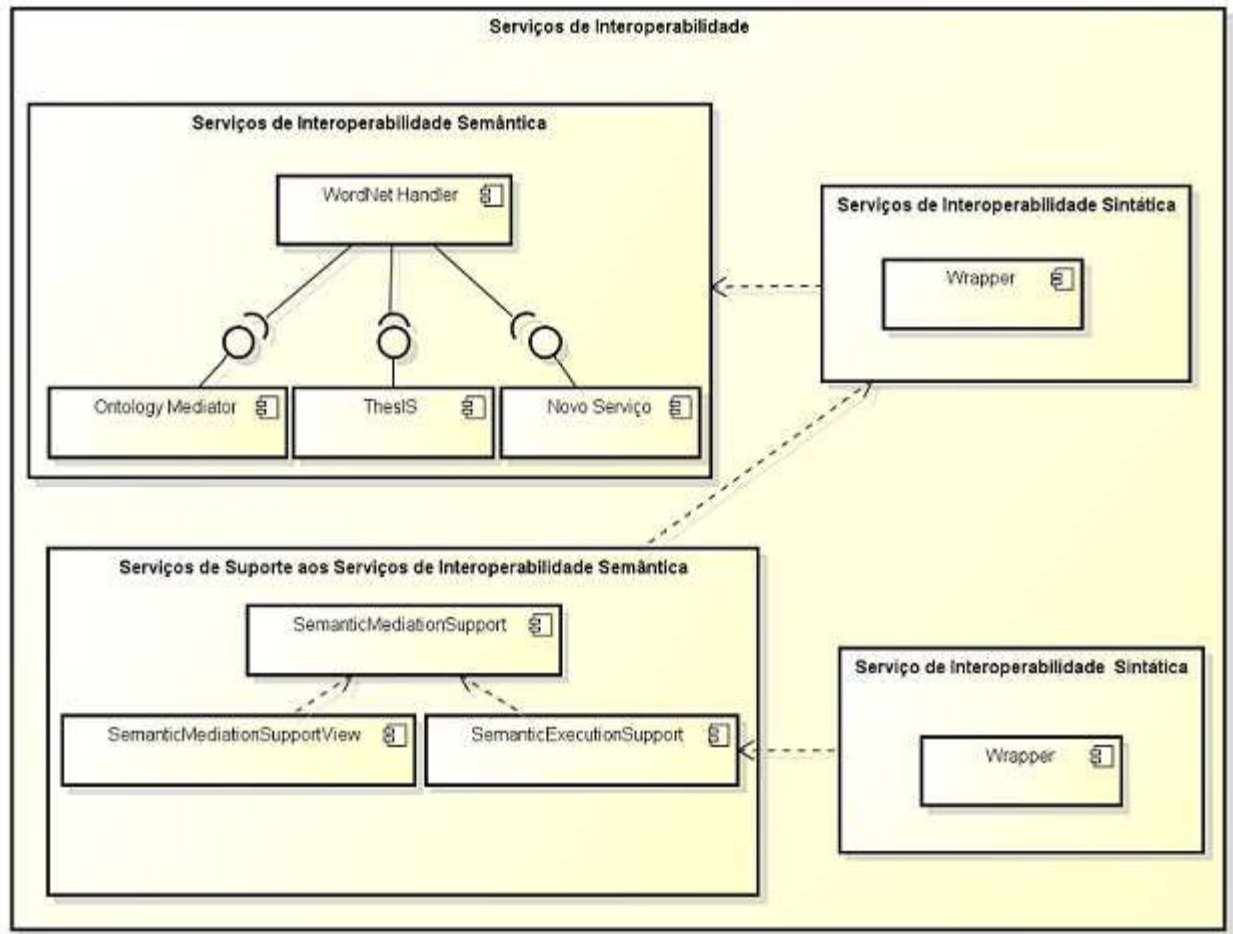


Figura 3.2: Diagrama de Componentes da Solução Proposta.

diretamente. Neste ponto, tanto a interface de usuário `SemanticMediationSupportView` quanto o serviço `SemanticExecutionSupport` compõem com o serviço `SemanticMediationSupport`. O `SemanticMediationSupport` é um serviço composto por dois outros serviços, `ThesIS` e `OntologyMediator`. Além da composição, esse serviço prepara os resultados obtidos para serem retornados corretamente a quem o chamou. Os serviços `ThesIS` e `OntologyMediator` têm como principal função nesta solução retornarem a resposta afirmativa ou não sobre a possibilidade de dois serviços distintos serem interoperáveis. Para isso, ambos utilizam uma consulta à WordNet através da funcionalidade `WordNetHandler`.

3.3 Implementação

A fase de implementação da solução proposta contemplou o apoio à interoperabilidade semântica em dois momentos: desenvolvimento e execução. Para apoiar à interoperabi-

lidade em tempo de desenvolvimento foi construída a funcionalidade SemanticMediationSupportView. Essa funcionalidade conta com uma série de interfaces de usuário que buscam atender aos requisitos funcionais necessários.

Para atender aos requisitos funcionais de (i) oferecer um mecanismo para selecionar os arquivos contendo a descrição semântica dos serviços candidatos a interoperabilidade (RF1) e (ii) de utilizar os serviços OntologyMediator e ThesIS (RF3), foi implementada a interface de usuário exibida na Figura 3.3. Nessa interface, o desenvolvedor escolhe o arquivo contendo a descrição dos serviços candidatos à interoperabilidade. Para usar o serviço ThesIS é necessário ter o tesauros que descreve o serviço gravado em um arquivo no formato XML, já para usar o serviço OntologyMediator a ontologia que descreve o serviço deve estar gravada no formato OWL. Não é obrigatório o preenchimento dos arquivos referentes aos dois serviços.

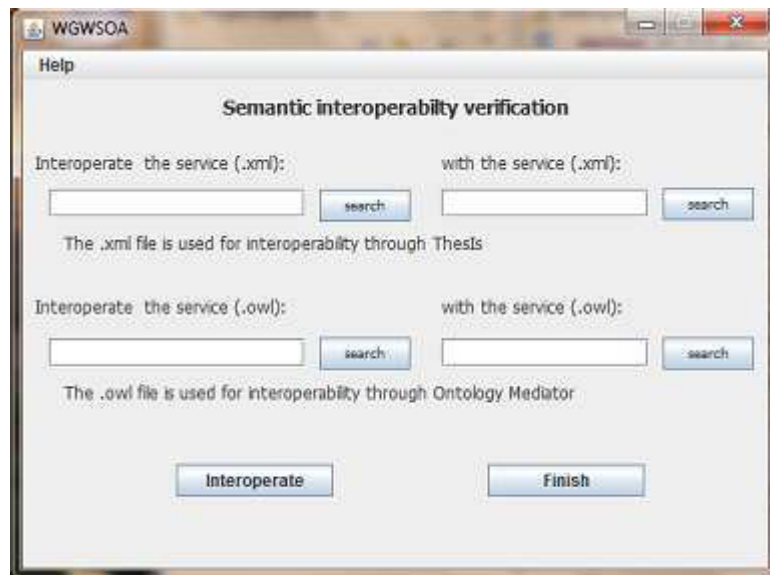


Figura 3.3: Seleção dos Arquivos com a Descrição dos Serviços Candidatos à Interoperabilidade Semântica.

Ao clicar no botão *interoperate*, são exibidos os resultados gerais de cada técnica (ThesIS e OntologyMediator), além de um gráfico comparativo entre elas que pode ser visto na Figura 3.4. Nessa interface de usuário, também podem ser vistos dois *checkboxes* cuja marcação irá determinar qual técnica (podendo ser ambas) será usada na análise dos resultados de interoperabilidade presentes na interface de usuário representada pela Figura 3.5.

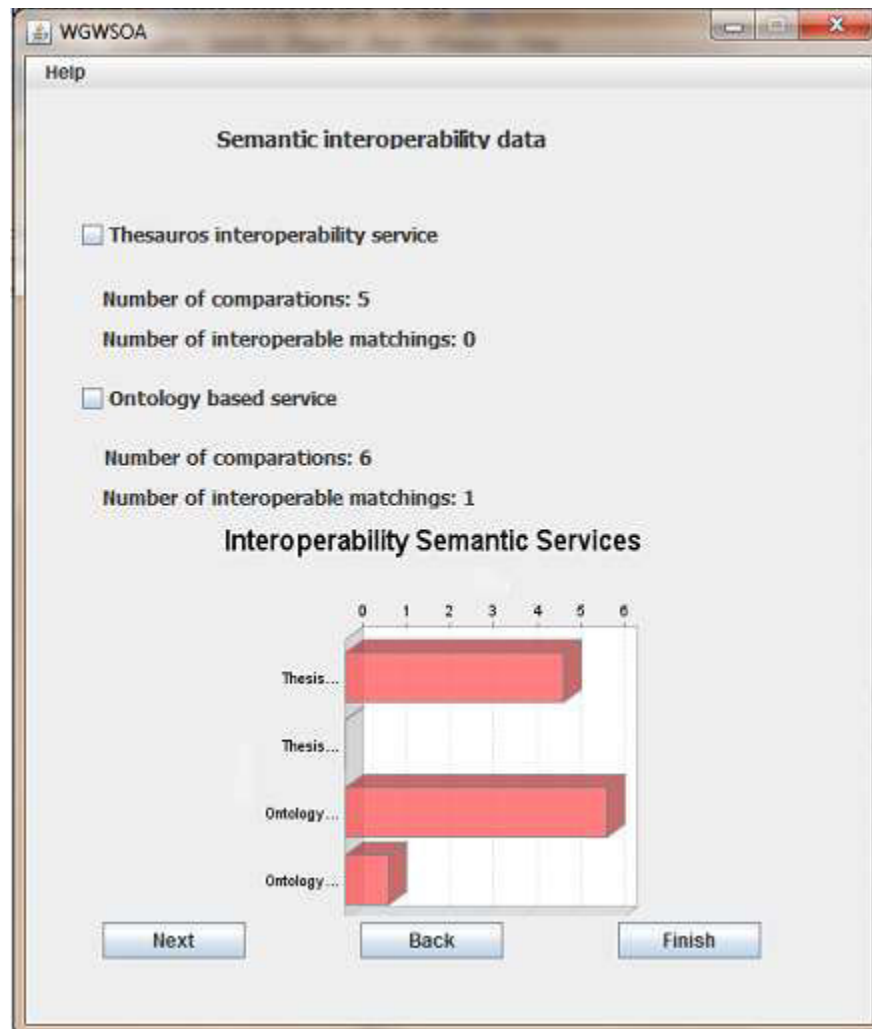


Figura 3.4: Relatório Geral da Interoperabilidade Semântica entre os Serviços Candidatos.

Desta forma, a interface de usuário representada pela Figura 3.4 atende ao requisito funcional de oferecer uma visualização detalhada sobre os resultados obtidos através do uso do OntologyMediator e do ThesIS (RF4).

Ao clicar no botão *next* é exibida uma interface de usuário, representada pela Figura 3.5, onde é possível obter uma análise mais refinada dos resultados. Nessa interface, o desenvolvedor poderá marcar quais resultados considerados interoperáveis e não interoperáveis pela técnica ele concorda ou discorda. Com a implementação dessa interface, conseguiu-se atingir o requisito funcional de oferecer ao desenvolvedor mecanismos para registrar sua concordância ou não com os resultados obtidos através do uso OntologyMediator e do ThesIS (RF5).

Ao clicar no botão *next* na interface de usuário representada pela Figura 3.5, a última interface de usuário é exibida. Essa interface, como mostrada na Figura 3.6

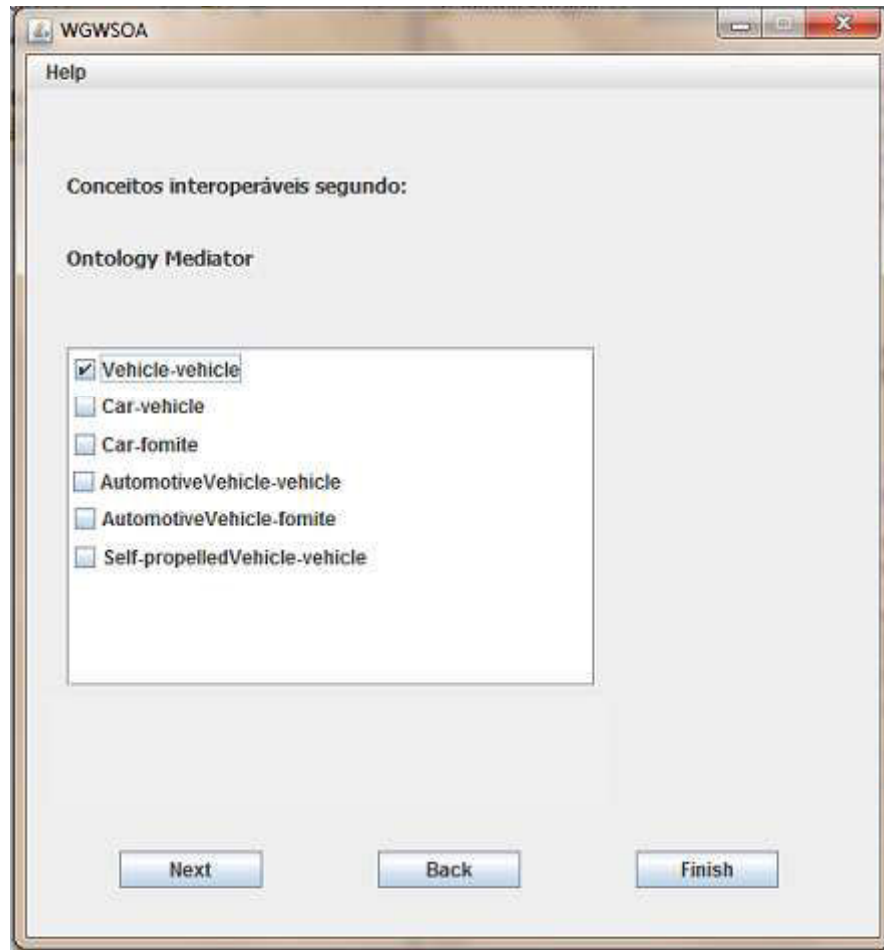


Figura 3.5: Detalhamento da interoperabilidade semântica entre termos.

atende ao requisito funcional de oferecer ao desenvolvedor acesso a informações sobre como desenvolver na infraestrutura (RF6). Na Figura 3.6 é possível observar a presença de *links* para diferentes tutoriais que irão ajudar o desenvolvedor da WGWSOA.

Com relação ao apoio à interoperabilidade semântica em tempo de execução e os requisitos funcionais que ele demanda, foi construído o serviço SemanticExecutionSupport. O serviço oferece um *log* que registra os eventos relevantes relacionados à interoperabilidade semântica durante a comunicação de serviços. Desta forma, ele atende ao requisito funcional de oferecer um meio de monitorar e visualizar detalhes sobre a interoperabilidade semântica (RF2)(RF4). Os dados deste *log* são fornecidos pelos serviços OntologyMediator e ThesIS e assim, atende ao requisito funcional de usar tais serviços para apoiar à interoperabilidade semântica (RF3).

Além das informações sobre implementação apresentadas nesta seção, é impor-

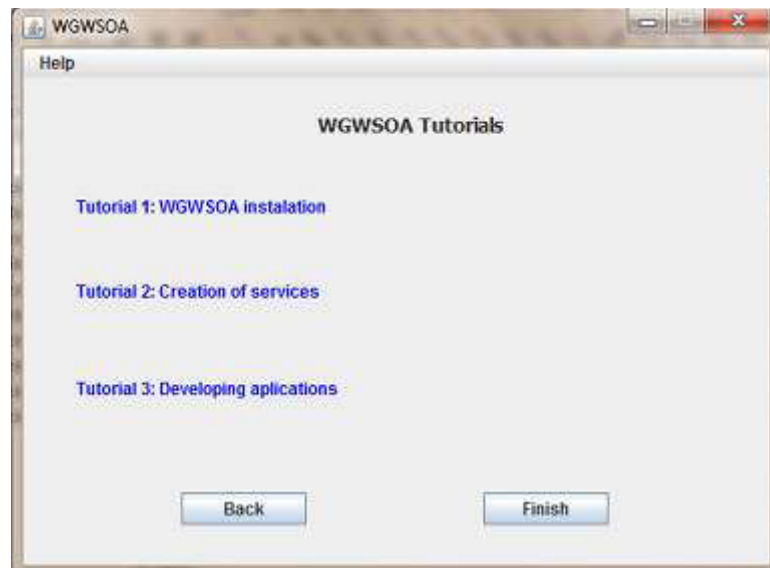


Figura 3.6: Material de Auxílio sobre como Desenvolver na WGWSOA.

```
***** Início do Log - Ontology Mediator *****
Interoperabilidade semântica do arquivo:
C:/Documents and Settings/user/Desktop/Ontologias/carontology1.owl
Com o arquivo:
C:/Documents and Settings/user/Desktop/Ontologias/carontology2.owl
Comparações:
Vehicle-vehicle
Car-vehicle
Car-fomite
AutomotiveVehicle-vehicle
AutomotiveVehicle-fomite
Self-propelledvehicle-vehicle
Casamentos:
Vehicle-vehicle
***** Fim do Log - Ontology Mediator *****
```

Figura 3.7: Log de Monitoramento da Interoperabilidade Semântica.

tante destacar outros aspectos relacionados às tecnologias e ferramentas utilizadas: Java² foi a linguagem de implementação usada para o desenvolvimento dos serviços e funcionalidades. A IDE Eclipse³ foi a ferramenta usada para constituir o ambiente de desenvolvimento. O *plugin* RMI⁴ foi usado na IDE Eclipse para compilação e geração dos stubs dos serviços. O *plugin* Window Builder⁵ foi usado na IDE Eclipse para auxiliar no desenvolvimento das interfaces de usuário. O servidor web Apache TomCat 7.0⁶ foi

²<http://www.java.com>

³<http://www.eclipse.org>

⁴<http://www.genady.net/rmi/v20/downloads.html>

⁵<http://www.eclipse.org/windowbuilder/>

⁶<http://tomcat.apache.org>

usado para viabilizar o serviço de Wrapper necessário às comunicações entre instâncias remotas da WGWSOA. O servidor Xamp⁷ foi usado para a manutenção da base de dados da WGWSOA.

3.4 Testes Funcionais

Os testes funcionais serviram para verificar o atendimento da solução proposta aos requisitos funcionais. A solução proposta pode ser dividida em duas soluções menores, uma para o apoio a interoperabilidade semântica em tempo de desenvolvimento e outra em tempo de execução, desta forma os testes também foram divididos em duas partes para facilitar sua execução e entendimento.

Os testes referentes ao apoio a interoperabilidade semântica em tempo de desenvolvimento estão relacionados ao componente `SemanticMediationSupportView`. Como indica o quadro representado na Figura 3.8, os requisitos funcionais cobertos pelo `SemanticMediationSupportView` foram testados através das funcionalidades implementadas que os endereçam. Para elaboração e execução dos testes foram destacados os resultados esperados e os resultados obtidos. Todos os resultados obtidos para o conjunto de casos de teste definido foram ao encontro do resultado que era esperado e desta forma, há evidências que a solução proposta atendeu aos requisitos funcionais elicitados.

Os testes referentes ao apoio a interoperabilidade semântica da solução proposta em tempo de execução estão relacionados ao serviço `SemanticExecutionSupport`. Para verificar o atendimento dos requisitos funcionais cobertos por este serviço foi desenvolvida uma aplicação, cujo principal objetivo era chamá-lo via Wrapper afim de criar um cenário propício para execução dos testes. No quadro representado pela Figura 3.9 é possível observar que a execução dessa aplicação conseguiu atender aos dois requisitos funcionais que devem ser cobertos pelo serviço `SemanticExecutionSupport`. Os testes revelaram que os resultados obtidos ficaram dentro do esperado, logo, há evidências que os requisitos funcionais foram satisfeitos.

⁷<http://sourceforge.net/projects/xampp/>

Funcionalidade Testada	Resultados Esperados	Resultados Obtidos	Requisito Funcional Relacionado
Selecionar arquivo através do botão "search". Vide Figura 3.3.	Abrir uma janela navegável que permita a procura e seleção de arquivos.	Conforme o esperado.	A WGWSOA deve oferecer ao desenvolvedor um mecanismo para selecionar os arquivos contendo a descrição semântica dos serviços candidatos a interoperabilidade (RF1).
Clicar no botão "interoperate" após seleção dos arquivos com extensão XML usados pelo TheSIS. Vide Figura 3.3	Exibir os resultados obtidos pelo serviço TheSIS. Vide Figura 3.4.	Conforme o esperado.	A WGWSOA deve oferecer um meio do desenvolvedor e usuário usar os serviços OntologyMediator e TheSIS já existentes na infraestrutura para apoiar a interoperabilidade semântica (RF3).
Clicar no botão "interoperate" após seleção dos arquivos OWL usados pelo OntologyMediator. Vide Figura 3.3.	Exibir os resultados obtidos pelo serviço OntologyMediator. Vide Figura 3.4.	Conforme o esperado.	A WGWSOA deve oferecer um meio do desenvolvedor e usuário usar os serviços OntologyMediator e TheSIS já existentes na infraestrutura para apoiar a interoperabilidade semântica (RF3).
Clicar no botão "interoperate" após seleção dos arquivos com extensão XML e OWL usados respectivamente pelo TheSIS e OntologyMediator. Vide Figura 3.3.	Exibir os resultados obtidos por ambos os serviços: TheSIS e OntologyMediator. Vide Figura 3.4.	Conforme o esperado.	A WGWSOA deve oferecer um meio do desenvolvedor e usuário usar os serviços OntologyMediator e TheSIS já existentes na infraestrutura para apoiar a interoperabilidade semântica (RF3).
Marcar os dois checkbox e clicar em "next". Vide Figura 3.4.	Exibir os resultados para análise obtidos pelos serviços: TheSIS e OntologyMediator. Vide Figura 3.5.	Conforme o esperado.	A WGWSOA deve oferecer ao desenvolvedor e ao usuário opções de visualização detalhadas sobre os resultados obtidos através do uso OntologyMediator e do TheSIS (RF4).
Marcar o checkbox referente ao uso do serviço TheSIS e clicar em "next". Vide Figura 3.4.	Exibir os resultados para análise obtido pelo serviço TheSIS. Vide Figura 3.5.	Conforme o esperado.	A WGWSOA deve oferecer ao desenvolvedor e ao usuário opções de visualização detalhadas sobre os resultados obtidos através do uso OntologyMediator e do TheSIS (RF4).
Marcar o checkbox referente ao uso do serviço OntologyMediator e clicar em "next". Vide Figura 3.4.	Exibir os resultados para análise obtido pelo serviço OntologyMediator. Vide Figura 3.5.	Conforme o esperado.	A WGWSOA deve oferecer ao desenvolvedor e ao usuário opções de visualização detalhadas sobre os resultados obtidos através do uso OntologyMediator e do TheSIS (RF4).
Editar os resultados de análise. Vide Figura 3.5.	Permitir a edição dos resultados obtidos através dos métodos escolhidos. Vide Figura 3.5.	Conforme o esperado.	A WGWSOA deve oferecer ao desenvolvedor mecanismos para registrar sua concordância ou não com os resultados obtidos através do uso OntologyMediator e do TheSIS (RF5).
Clicar em "next" na interface que disponibiliza os resultados obtidos para análise. Vide Figura 3.5.	Exibição dos tutoriais disponíveis. Vide Figura 3.6.	Conforme o esperado.	A WGWSOA deve oferecer ao desenvolvedor acesso a informações sobre como trabalhar na infraestrutura (RF6).
Clicar no link dos tutoriais disponíveis. Vide Figura 3.6	Exibição do tutorial completo.	Conforme o esperado.	A WGWSOA deve oferecer ao desenvolvedor acesso a informações sobre como trabalhar na infraestrutura (RF6).

Figura 3.8: Quadro de Casos de Teste do Componente SemanticMediationSupportView

Funcionalidade Testada	Resultados Esperados	Resultados Obtidos	Requisito Funcional Relacionado
Executar a aplicação que chama o serviço SemanticExecutionSupport	Exibição do log de resultados obtidos através dos serviços TheSIS e OntologyMediator no console.	Conforme o esperado.	- A WGWSOA deve oferecer um meio do desenvolvedor e usuário usar os serviços OntologyMediator e TheSIS já existentes na infraestrutura para apoiar interoperabilidade semântica (RF3). - A WGWSOA deve oferecer ao usuário um meio para monitorar a interoperabilidade semântica durante o uso de serviços da infraestrutura (RF2).

Figura 3.9: Quadro de Casos de Teste do Componente SemanticExecutionSupport

3.5 Considerações Finais do Capítulo

Neste capítulo, a solução proposta foi apresentada, compreendendo suas etapas de especificação, projeto, implementação e testes funcionais. Os testes funcionais, geraram

evidências que indicam que os requisitos funcionais levantados na etapa de especificação da solução proposta foram atendidos. Porém, apesar dos resultados positivos obtidos, existe ainda a necessidade de avaliar se a solução proposta é capaz de gerar evidências sobre o atendimento aos requisitos não funcionais demandados.

4 Avaliação da Solução Proposta

O uso de métodos experimentais para avaliar soluções produzidas na área de engenharia de software pode produzir alguns benefícios como: eliminar abordagens não fundamentadas, suposições errôneas e modismos, e, auxiliar na identificação de linhas de pesquisa promissoras. As principais estratégias experimentais encontradas na literatura são: *survey*, estudo de caso e experimento (Kalinowski, 2011). Em Wohlin et al. (2012) uma descrição detalhada dessas três estratégias pode ser encontrada.

No tocante às aplicações de sistemas colaborativos, não existe um consenso sobre o método experimental mais adequado para avaliar as soluções produzidas na área de engenharia de software. Em um estudo de caso o objetivo principal é avaliar situações atuais nas quais não é possível manipular eventos comportamentais. Como para a solução proposta neste trabalho é importante que se considere o contexto real de utilização envolvendo a realização de atividades contemporâneas, o estudo de caso foi a opção escolhida. Neste capítulo, é apresentado um estudo experimental constituído por dois estudos de caso com a finalidade de avaliar a solução proposta em relação ao atendimento aos requisitos não funcionais definidos no capítulo anterior.

4.1 Definição do Estudo Experimental

O escopo dos estudos de caso foi definido com o auxílio do método GQM (Basili et al., 1994) e se resume a estrutura a seguir:

“**Analisar** a solução proposta **com o propósito de** avaliar o suporte à interoperabilidade semântica oferecido pela solução **com respeito a** sua capacidade de composição, escalabilidade e reusabilidade **do ponto de vista de** usuários e desenvolvedores **no contexto da** realização de atividades que demandam interoperabilidade semântica na infraestrutura WGWSOA.”

Assim, as questões principais nestes estudos de caso se relacionam à investigação da capacidade de composição, escalabilidade e reusabilidade oferecida pela solução proposta.

Desta forma, três questões serão observadas durante os estudos:

- Q1: Como a utilização da solução proposta implica em uma maior capacidade de composição dos serviços relacionados à interoperabilidade semântica na WGWSOA, em comparação a sua não utilização?
- Q2: Como a solução proposta consegue atender ao requisito não funcional de escalabilidade?
- Q3: Como a solução proposta consegue atender ao requisito não funcional de reusabilidade?

A forma com que o estudo experimental será planejado para responder a essas questões é explicado na próxima seção.

4.2 Planejamento do Estudo Experimental

Segundo Wohlin et al. (2012), o planejamento do estudo experimental é um processo que busca preparar *como* o estudo será conduzido. O autor divide a fase de planejamento em sete etapas que serão seguidas neste estudo. As etapas consistem em: (i) seleção do contexto - pode ser caracterizado segundo quatro dimensões como, *off-line* vs. *on-line*, estudantes vs. profissionais, real vs. modelado e específico vs. generalizado; (ii) formulação das hipóteses - podem ser categorizadas como nulas e alternativas. De forma geral, as hipóteses nulas são aquelas que queremos rejeitar, enquanto as alternativas são aquelas que pretendemos comprovar; (iii) seleção das variáveis - as variáveis são categorizadas em independentes e dependentes. As variáveis independentes são aquelas que podemos controlar e modificar, já as dependentes, são afetadas por modificações nas variáveis independentes e muitas vezes podem ser derivadas diretamente da hipótese; (iv) seleção de indivíduos - compreende a técnica que será usada para definir a amostra da população que participará do estudo; (v) projeto do experimento - descreve como os testes serão organizados e executados no estudo; (vi) instrumentação - seu objetivo é fornecer meios de executar e monitorar o estudo e; (vii) avaliação da validade - é importante considerar a

questão da validade já na fase de planejamento, pois isto ajuda a planejar uma validação adequada para os resultados que se deseja obter.

A execução de cada etapa do planejamento do estudo experimental que buscará avaliar o suporte à interoperabilidade semântica da solução oferecida se encontra a seguir:

Seleção do Contexto. Foi definido que os estudos de caso serão realizados *on-line*, uma vez que envolve a realização de atividades que usam serviços de apoio à interoperabilidade semântica. Os participantes selecionados serão estudantes e atuarão em problemas especificamente modelados para este estudo experimental, cujo foco está em um domínio específico de sistemas colaborativos, pois as evidências produzidas estarão sujeitas ao ambiente da WGWSOA.

Formulação das Hipóteses. Para este estudo experimental foram formuladas três hipóteses nulas e três hipóteses alternativas. São elas:

1. Hipóteses Nulas: (1.1) A solução proposta não aumenta a capacidade de composição dos serviços relacionados à interoperabilidade semântica na WGWSOA; (1.2) A solução proposta não é escalável. (1.3) A solução proposta não é reutilizável.
2. Hipóteses alternativas: (2.1) A solução proposta melhora a capacidade de composição dos serviços relacionados à interoperabilidade semântica na WGWSOA; (2.2) A solução proposta é escalável; (2.3) A solução proposta é capaz de ser reutilizada.

As etapas que compreendem a seleção de variáveis e definição do projeto são flexíveis em um estudo de caso. Isso se deve ao fato de não haver um controle rígido como em um experimento. Além disso, a coleta e análise dos dados pode ser conduzida de forma incremental e, a cada incremento, pode haver modificações nessas etapas do planejamento (Wohlin et al., 2012).

Seleção de Indivíduos. Na seleção de indivíduos para participarem do estudo experimental, foram selecionados alunos que cursavam graduação em Ciência da Computação ou áreas afins e possuíam conhecimentos na linguagem Java.

Projeto do Estudo Experimental. Nesta etapa, ficou definido que em cada estudo de caso haverá uma mesma quantidade de participantes com conhecimentos similares

nos seguintes assuntos: programação Java, desenvolvimento de sistemas distribuídos e em tecnologias que se baseiam nos conceitos de SOA. Para tal, os candidatos à participantes responderão a um questionário de caracterização que buscará analisar seus conhecimentos.

Para ser possível comprovar ou rejeitar as hipóteses formuladas, os participantes realizarão algumas tarefas que envolvam o uso de serviços de apoio a interoperabilidade semântica. Após realizá-las, serão aplicados questionários com o objetivo de coletar dados referentes à realização das tarefas e esses dados serão analisados.

Instrumentação. Para a realização deste estudo experimental, os seguintes instrumentos foram selecionados:

- Documento contendo um tutorial de criação de serviços na WGWSOA.
- Máquina configurada com o ambiente de uso da infraestrutura WGWSOA.
- Documento contendo a especificação e projeto das tarefas a serem realizadas na WGWSOA.
- Um arquivo XML e um arquivo OWL, ambos contendo a descrição de serviços da WGWSOA.
- Questionário de caracterização do participante.
- Questionário de *follow-up* para coletar dados após a realização das tarefas feitas no estudo experimental.
- Questionário de consentimento do participante para a publicação neste trabalho dos dados coletados no estudo experimental.

Avaliação da Validade. As ameaças à validade deste estudo experimental estão descritas a seguir:

- Validade Interna: Uma ameaça à validade interna deste estudo está associada aos conhecimentos dos participantes na linguagem Java, em desenvolvimento de sistemas distribuídos e em tecnologias baseadas no conceito de SOA. Um conhecimento inadequado por parte dos participantes ou então, disparidades significativas entre o nível de conhecimentos dos mesmos podem influenciar nos resultados obtidos neste

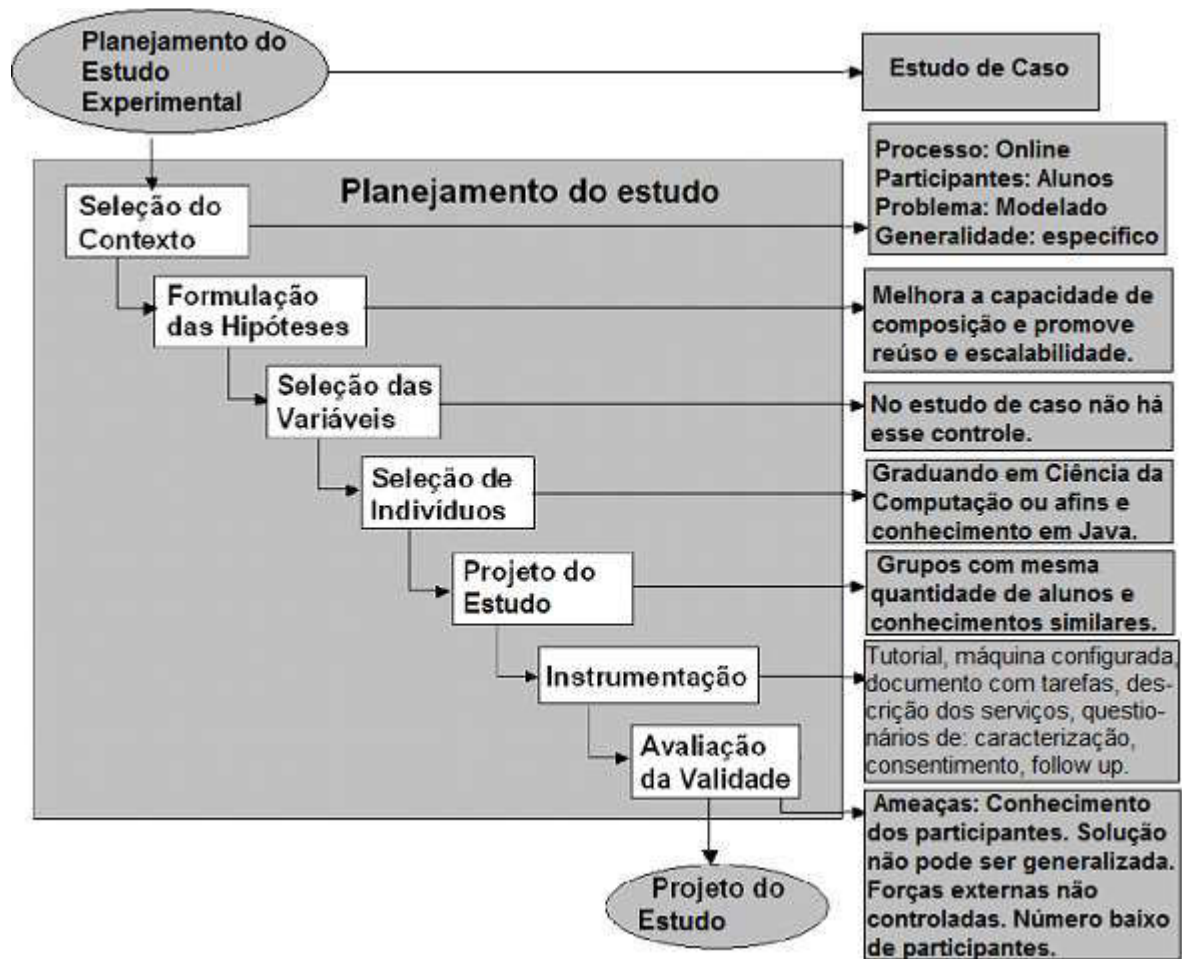


Figura 4.1: Etapas da Fase de Planejamento de um Estudo Experimental

estudo experimental. Para mitigar essas ameaças, os participantes preencherão um questionário de caracterização e além disso, um treinamento abordando ensinamentos necessários a realização das tarefas do estudo experimental será oferecido aos participantes. Outra ameaça é a influência que um participante poderia exercer sobre os outros, direta ou indiretamente, durante a realização do estudo. Para evitar que isso ocorresse, optou-se por executar individualmente um estudo de caso com cada participante.

- **Validade Externa:** O estudo experimental avaliará apenas o uso da solução proposta no contexto da WGWSOA. Desta forma, qualquer evidência produzida por este estudo não possui nenhuma validade de generalização em outras infraestruturas que apoiem o desenvolvimento de sistemas colaborativos.
- **Validade de Construção:** O uso do método de estudo de caso ameaça a validade

de construção. No estudo de caso, as etapas de seleção de variáveis e definição do projeto são flexíveis pois, não há um controle rigoroso durante o estudo. Sendo assim, diversas forças externas que podem influenciar no resultado obtido, não serão monitoradas e gerenciadas.

- **Validade de Conclusão:** A validade de conclusão está relacionada a possibilidade de gerar conclusões com significância estatística. Como este estudo contou com um número pequeno de participantes, a validade de conclusão foi ameaçada, pois não é possível gerar dados estatisticamente significativos.

Os resultados gerados pela execução das sete etapas acima, definem o projeto deste estudo experimental. Com base nesse projeto, os estudos de caso da próxima seção foram realizados. Um resumo dos produtos obtidos por cada fase pode ser visto na Figura 4.1.

4.3 Operação: Primeiro Estudo de Caso

4.3.1 Objetivo

O objetivo deste estudo de caso foi de avaliar a solução proposta em relação à capacidade de composição dos serviços de apoio à interoperabilidade semântica através da realização de tarefas que ora considerem seu uso e ora não. Além disso, deseja-se observar a capacidade de reúso e escalabilidade da solução proposta.

4.3.2 Caracterização do Participante

A caracterização do participante foi obtida através da aplicação de um questionário (APÊNDICE A). Para este estudo de caso foi selecionado um estudante de graduação. Esse estudante cursava Ciência da Computação e possuía um bom conhecimento no desenvolvimento de sistemas utilizando a linguagem Java. Porém, possuía um conhecimento baixo em relação ao desenvolvimento de sistemas distribuídos e em relação a tecnologias que utilizam conceitos de SOA. A avaliação durou aproximadamente duas horas e trinta minutos e foi dado previamente ao participante um treinamento de quarenta minutos. O

treinamento buscou abordar ensinamentos sobre os objetivos e o funcionamento da WGWSOA. Com relação ao contexto da WGWSOA, foram passadas informações sobre o uso de RMI e os conceitos de SOA. O desenvolvimento e execução de serviços na infraestrutura também foi contemplado no treinamento.

4.3.3 Cenário

Após o término do treinamento, foram passadas três tarefas para serem realizadas pelo participante. Cada tarefa tinha como objetivo gerar evidências que endereçassem um ou mais dos requisitos não funcionais que estão sendo observados. Durante a realização das tarefas o participante assumia ora o papel de usuário de serviços, ora de desenvolvedor através da WGWSOA.

A primeira tarefa propôs ao participante usar as descrições de dois serviços fornecidos para verificar as informações sobre a interoperabilidade semântica. Para isso, o participante deveria primeiro fazer esta verificação usando cada um dos serviços ThesIS e OntologyMediator da WGWSOA. Em seguida, a mesma verificação seria realizada usando a funcionalidade SemanticMediationSupportView que faz parte da solução proposta. Essa tarefa tinha como objetivo analisar como a capacidade de composição está sendo atendida.

A segunda tarefa dada ao participante foi monitorar a interoperabilidade semântica durante a execução de algum serviço da WGWSOA. Para isso, foi solicitado ao participante que ele desenvolvesse um serviço formado pela composição de outros dois já existentes na WGWSOA: chat e lista de usuários. O serviço criado deveria ser executado pelo participante inicialmente utilizando o serviço SemanticExecutionSupport, que faz parte da solução proposta, e em seguida sem o uso deste serviço.

A terceira e última tarefa que deveria ser realizada pelo usuário era criar um novo serviço na infraestrutura WGWSOA que simulasse uma nova forma de apoiar a interoperabilidade semântica entre serviços. Para a implementação desse serviço simulador, foi indicado ao participante criar uma lógica de implementação que retornasse aleatoriamente uma resposta positiva ou negativa sobre a interoperabilidade semântica entre dois serviços. O serviço simulador criado deveria ser adicionado ao serviço SemanticMediationSupport, que faz parte da solução proposta, com o objetivo de constituir uma nova

forma de interoperabilidade semântica contemplada pela WGWSOA. O entendimento das tarefas demorou aproximadamente trinta minutos e sua execução em torno de duas horas.

4.3.4 Fontes de Evidências

Para a coleta de dados, foi utilizado um questionário (APÊNDICE B) e observações diretas. As observações diretas constituíram uma fonte de dados importante para o estudo experimental, uma vez que foi capaz de revelar informações relacionadas às dificuldades de uso e limitações da solução que não foram capturadas através da aplicação dos questionários.

4.3.5 Lições Aprendidas

Com a execução deste primeiro estudo de caso, às observações diretas e a seção “Com relação às dificuldades encontradas durante a realização das tarefas” do questionário de *follow-up* foram utilizadas para fornecer lições no sentido de aprimorar o próximo estudo de caso. O participante relatou dificuldades em relação ao entendimento dos resultados apresentados em uma das interfaces de usuário da funcionalidade SemanticMediationSupportView e também considerou a abordagem sobre como realizar a composição de serviços na WGWSOA insuficiente durante o treinamento. As observações diretas comprovaram as considerações do participante, uma vez que foram percebidas essas dificuldades durante a realização das tarefas. Além disso, as observações revelaram que os serviços TheSIS e OntologyMediator também deveriam ser melhor explicados durante o treinamento pois, a dificuldade de compreensão de uma das interfaces de usuário da funcionalidade SemanticMediationSupportView poderia ser fruto dessa deficiência.

O uso de um participante com bons conhecimento em Java neste primeiro estudo de caso auxiliou a calibrar a operação do próximo estudo de caso, pois diminuiu a interferência de possíveis problemas técnicos que poderiam influenciar na percepção de falhas do estudo de caso que deveriam ser ajustadas.

4.4 Operação: Segundo Estudo de Caso

4.4.1 Objetivo

Este estudo teve o mesmo objetivo do estudo anterior, porém buscou-se corrigir as falhas reveladas durante a análise dos dados coletados no estudo de caso anterior.

4.4.2 Caracterização do Participante

A caracterização do participante foi obtida através do mesmo questionário do estudo de caso anterior (APÊNDICE A). O único ajuste feito foi referente a quarta questão que passou a ser descrita desta maneira: “Como você avaliaria seus conhecimentos em relação a tecnologias que utilizam conceitos de SOA? (ex: *Web Services* e RMI)”.

Para este estudo de caso foi selecionado um participante de graduação do curso Sistemas de Informação, que possuía um conhecimento regular sobre a linguagem Java e um conhecimento baixo em relação ao desenvolvimento de sistemas distribuídos e às tecnologias que se baseiam no conceito de SOA. Foi oferecido ao participante um treinamento de aproximadamente uma hora, diferente do treinamento do estudo de caso anterior, neste treinamento foi dada uma explicação mais detalhada sobre o funcionamento dos serviços ThesIS e OntologyMediator e da funcionalidades SemanticMediationSupportView. Além disso, foi incluído um exercício prático para demonstrar como pode ser feita a composição de serviços na WGWSOA. A avaliação deste estudo durou aproximadamente três horas.

4.4.3 Cenário

O cenário deste estudo de caso foi o mesmo do estudo anterior. Ao participante foram passadas as mesmas três tarefas do estudo de caso anterior para serem realizadas no ambiente WGWSOA. A leitura e entendimento das tarefas duraram aproximadamente vinte minutos e a execução das mesmas durou aproximadamente duas horas e trinta minutos.

4.4.4 Fontes de Evidências

As fontes de evidências para este estudo de caso foram as mesmas do estudo de caso anterior. Os dados foram coletados através de observação direta e o questionário de *follow-up* foi aplicado ao participante logo após o término da realização das tarefas (APÊNDICE B).

4.4.5 Lições Aprendidas

Neste estudo de caso, como feito no estudo anterior, foram utilizadas as observações diretas e a seção “Com relação às dificuldades encontradas durante a realização das tarefas” do questionário de *follow-up* para fornecer as lições aprendidas. O participante deste estudo de caso não relatou ter enfrentado problemas técnicos durante a realização das tarefas e considerou o treinamento dado suficiente para a realização das tarefas. As observações diretas comprovaram que houve um maior entendimento sobre o funcionamento dos serviços ThesIS, OntologyMediator e da funcionalidade SemanticMediationSupport-View. Além disso, não foram notadas dificuldades sobre como realizar a composição de serviços na WGWSOA durante as tarefas. Essa melhoria na qualidade de compreensão pode ser atribuída aos ajustes feitos no treinamento após o primeiro estudo de caso. Apesar disso, as observações diretas também constataram que houve uma demora maior para execução das tarefas, que parecia ser causada por uma menor experiência de desenvolvimento com a linguagem Java.

Após a execução dos dois estudos de caso, é possível considerar que a falta de conhecimento em desenvolvimento de sistemas distribuídos e em tecnologias que utilizam conceitos de SOA, não foram impeditivos para a realização das tarefas. Porém, o treinamento foi um instrumento essencial para suprir essas deficiência de conhecimento e tornar possível a execução dos estudos de caso.

4.5 Resultado e Análise dos Dados Coletados

Nesta seção serão apresentados os resultados quantitativos e qualitativos obtidos através do estudo experimental. Os resultados quantitativos, foram gerados a partir da análise das

questões fechadas do questionário *follow-up*. Para organização dos dados, as questões foram divididas em grupos de acordo com sua relação com um dos requisitos não funcionais que estavam sendo investigados (capacidade de composição, reusabilidade e escalabilidade). As opções de resposta de cada questão foram associadas a uma nota que variava em um intervalo fechado e discreto de um até cinco. Para quantificar o atendimento da solução proposta a um determinado requisito não funcional, foi contabilizada a média aritmética das notas obtidas pelas questões do grupo associado ao requisito não funcional em questão. Portanto, cada requisito não funcional poderia receber uma nota máxima de cinco pontos e mínima de um ponto em relação ao seu atendimento através da solução proposta. A divisão dos pontos por questão ficou da seguinte forma:

- Grupo 1 - Capacidade de composição.

Questão 1, 2 e 3: Discordo Fortemente (1 ponto), Discordo (2 pontos), Não Deseja e/ou não se sente capaz de opinar (3 pontos), Concordo (4 pontos), Concordo Fortemente (5 pontos).

- Grupo 2 - Reusabilidade.

Questão 1: Discordo Fortemente (5 pontos), Discordo (4 pontos), Não Deseja e/ou não se sente capaz de opinar (3 pontos), Concordo (2 pontos), Concordo Fortemente (1 ponto).

Questão 2: Discordo Fortemente (1 ponto), Discordo (2 pontos), Não Deseja e/ou não se sente capaz de opinar (3 pontos), Concordo (4 pontos), Concordo Fortemente (5 pontos).

- Grupo 3 - Escalabilidade.

Questão 1: Discordo Fortemente (1 ponto), Discordo (2 pontos), Não Deseja e/ou não se sente capaz de opinar (3 pontos), Concordo (4 pontos), Concordo Fortemente (5 pontos).

Questão 2: Discordo Fortemente (5 pontos), Discordo (4 pontos), Não Deseja e/ou não se sente capaz de opinar (3 pontos), Concordo (2 pontos), Concordo Fortemente (1 ponto).

Como pode ser visto na Figura 4.2, os resultados obtidos indicaram que o estudo experimental, constituído pelos dois estudos de caso, gerou evidências de que a solução proposta conseguiu atender aos requisitos não funcionais, uma vez que as notas chegaram próximo do valor máximo.

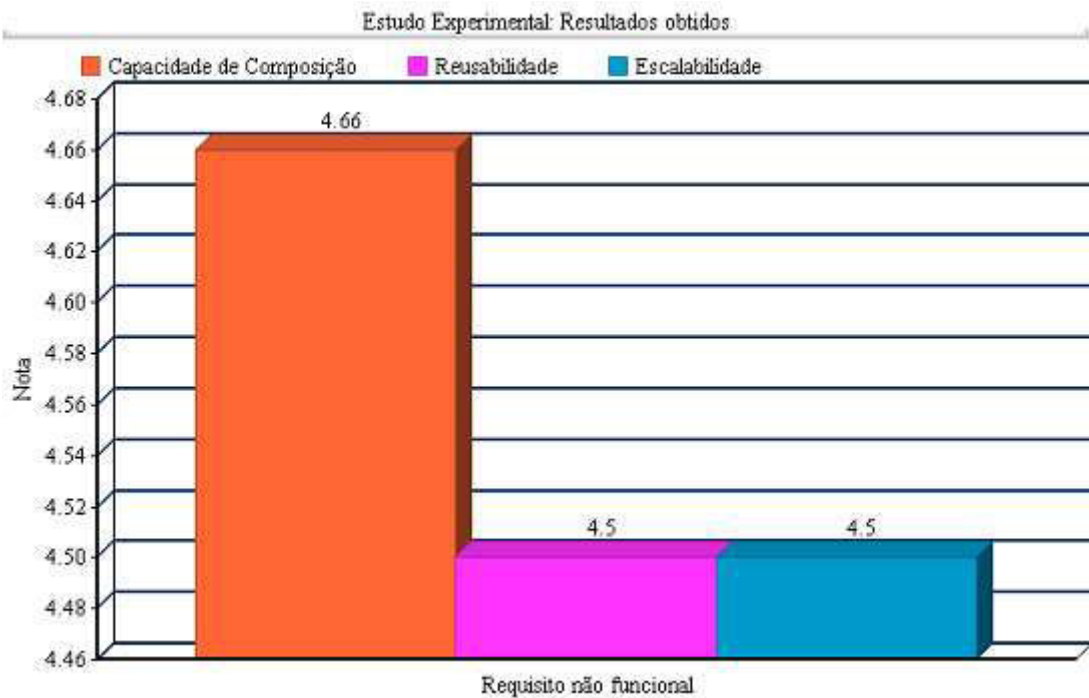


Figura 4.2: Resultados Obtidos Através dos Dados Coletados no Estudo Experimental.

Na Figura 4.3 é possível observar uma comparação entre os resultados obtidos no primeiro e no segundo estudos de caso. Apesar dos dados coletados resultarem em uma média de notas parecidas nos dois estudos de caso, percebe-se resultados mais positivos obtidos no segundo estudo de caso. É possível que os ajustes no treinamento para o segundo estudo de caso tenham contribuído para esta melhoria nas notas.

Para realizar a análise qualitativa do estudo experimental, as questões abertas do questionário *follow-up* foram analisadas. Essas questões tinham como objetivo principal identificar se o uso da solução proposta trazia benefícios para a capacidade de composição relacionada aos serviços de apoio a interoperabilidade semântica da WGWSOA. Em ambos os estudos de caso, os dados qualitativos coletados geraram evidências que o uso da solução proposta facilita a utilização dos serviços de interoperabilidade semântica em tempo de execução e desenvolvimento na WGWSOA. Os participantes consideraram que o uso da ferramenta tornou as atividades que demandaram suporte à interoperabilidade

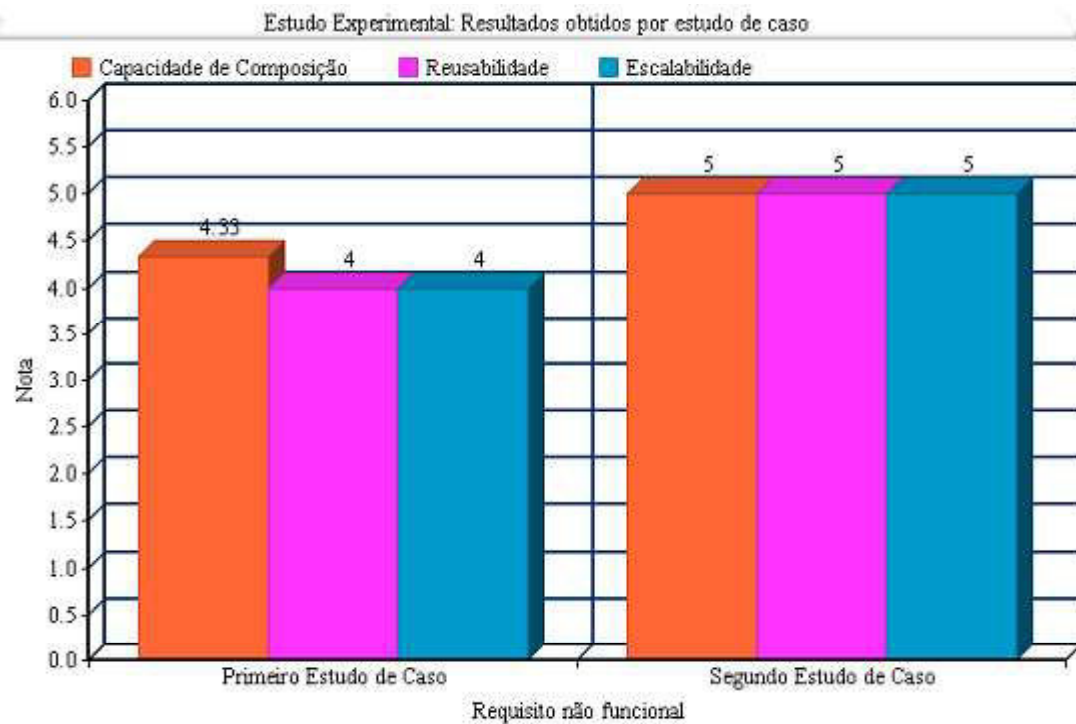


Figura 4.3: Comparativo dos Resultados Obtidos por Estudo de Caso.

semântica mais ágeis e simples por abstraírem complexidades de programação e fornecerem interfaces de usuário para execução das atividades. Além disso, os participantes tiveram mais facilidade de obter e compreender as informações sobre a interoperabilidade semântica em tempo de desenvolvimento e execução, sendo que em tempo de execução, o monitoramento da interoperabilidade semântica só foi possível com o uso da solução proposta.

4.6 Considerações Finais do Capítulo

A avaliação da solução proposta apresentada neste capítulo gerou evidências de que os seguintes requisitos não funcionais foram atendidos: capacidade de composição, reusabilidade e escalabilidade. Apesar destas evidências, existem ameaças a validade da avaliação, que também foram discutidas neste capítulo, como por exemplo, o número pequeno de participantes que impossibilita a geração de resultados com significância estatística. Dessa forma, outros estudos que mitiguem essas ameaças devem ser conduzidos.

5 Conclusões

Grupos de trabalho e suas atividades têm sido distribuídos geograficamente, isso gerou a necessidade de sistemas que apoiem a colaboração entre esses grupos. Os sistemas colaborativos trazem diversos desafios com ele, como por exemplo, dificuldades relacionadas à interoperabilidade. Com o objetivo de promover essa interoperabilidade nos níveis sintático, semântico e pragmático, necessários à uma colaboração eficiente, foi construída uma infraestrutura orientada à serviços de *middleware* chamada *Web-based Groupware Service-Oriented Architecture* (WGWSOA).

A WGWSOA contava com dois serviços para apoiar a interoperabilidade semântica: ThesIS e OntologyMediator. Porém, havia limitações no oferecimento desses serviços, como por exemplo, a ausência de um serviço que fosse capaz de compor com o ThesIS e o OntologyMediator, de maneira à facilitar seu uso dentro da infraestrutura, abstraindo complexidades de programação.

Desta maneira, este trabalho propôs uma alternativa para ampliar o apoio à interoperabilidade semântica entre serviços colaborativos em tempo de execução e desenvolvimento. Para isso, um serviço e uma funcionalidade foram desenvolvidos considerando a infraestrutura WGWSOA. O serviço desenvolvido buscou solucionar as limitações do apoio à interoperabilidade semântica promovido pelo uso do ThesIS e do OntologyMediator, oferecendo um meio de monitorá-los em tempo de execução, através do oferecimento de um *log* e da composição com esses serviços. Já a funcionalidade desenvolvida, tratou o suporte à interoperabilidade semântica em tempo de desenvolvimento e consistiu na construção de interfaces de usuário que permitem o uso combinado dos serviços ThesIS e OntologyMediator, sem que para isso seja necessário esforço de programação.

A avaliação da solução proposta neste trabalho incluiu a realização de dois estudos de caso. Esses estudos tiveram como objetivo analisar as seguintes hipóteses: (i) a solução proposta melhora a capacidade de composição dos serviços relacionados à interoperabilidade semântica na WGWSOA; (ii) a solução proposta é escalável; (iii) a solução proposta é capaz de ser reutilizada. As evidências produzidas através dos estudos de caso

indicaram a confirmação das hipóteses. Entretanto, experimentos adicionais necessitam ser realizados.

Ao longo do trabalho as dificuldades foram diversas, começando pela configuração do ambiente da WGWSOA. A falta de alguns documentos que indicassem os requisitos de software e hardware, geraram a necessidade de contactar alguns pesquisadores e expesquisadores da WGWSOA para conseguir essas informações. Para sanar essa dificuldade para os futuros pesquisadores da WGWSOA, os documentos foram melhorados e a configuração do ambiente foi feita utilizando uma máquina virtual que ficará a disposição da pesquisa. Outra dificuldade foi gerenciar as versões da WGWSOA e conseguir trabalhar em conjunto com grupos geograficamente dispersos de outras instituições, como o da Universidade Federal da Bahia (UFBA) que também desenvolve pesquisas usando a WGWSOA. Para mitigar esta dificuldade, optou-se pelo uso de um sistema de controle de versão conhecido como Apache Subversion⁸ (SVN).

Como trabalhos futuros relacionados à ampliação da interoperabilidade na WGWSOA, cabe citar: (i) Realizar estudos experimentais que gerem evidências com significância estatística sobre a eficiência dos serviços de apoio à interoperabilidade semântica; (ii) Investigar novas formas de apoiar à interoperabilidade semântica entre serviços, além dos métodos já existentes baseados em tesauros e ontologias; (iii) Investigar mecanismos que sejam capazes de apoiar à interoperabilidade pragmática, uma vez que a infraestrutura não oferece serviços dessa natureza.

Existe também o interesse por pesquisar como os diferentes níveis de interoperabilidade têm sido abordados no contexto de Linha de Produtos de Software (LPS). LPS é uma solução que têm sido cada vez mais adotada para aumentar a qualidade e produtividade do software e diminuir os custos de sua produção. Inclusive, há pesquisas em andamento que estudam a evolução da WGWSOA para uma linha de produto de software.

⁸<http://subversion.apache.org/>

Referências Bibliográficas

- Almeida, T. R.; David, J. M. N. ; Maciel, R. S. P. **Ampliando o suporte à interoperabilidade em uma arquitetura orientada a serviços de middleware para sistemas colaborativos**. In: Proceedings of the 2011 Simpósio Brasileiro de Sistemas Colaborativos, SBSC '11, p. 68–75, 2011.
- Almeida, T. R. **Contexto em groupware e interoperabilidade entre serviços de colaboração**. 2013. Dissertação de Mestrado - Mestrado em Sistemas e Computação - Universidade Federal da Bahia (UFBA).
- Arize, D.; Maciel, R. S. P. ; David, J. M. N. **ThesIS: A semantic interoperability service for a middleware service oriented architecture**. In: 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD), CSWD '13, 2013.
- Basili, V.; Caldiera, G. ; Rombach, H. The goal question metric approach. **Encyclopedia of Software Engineering**, v.1, p. 528–532, 1994.
- Bernstein, P. **Middleware: an architecture for distributed system services**. Technical report series. Digital Equipment Corporation, Cambridge Research Laboratory, 1993.
- Bernstein, P. Middleware: a model for distributed system services. **Communications of the ACM**, v.39, n.2, p. 86–98, 1996.
- Berners-Lee, T.; Hendler, J. ; Lassila, O. The Semantic Web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. **Scientific American**, May 2001.
- David, J. M. N. **Um Serviço de Percepção para uma Infra-estrutura de Desenvolvimento de Groupware**. 2004. Tese de Doutorado - Doutorado em Engenharia de Sistemas e Computação - Universidade Federal do Rio de Janeiro (UFRJ).
- David, J. M. N.; Maciel, R. S. P. **WGWSOA - Implementing collaboration services in a middleware infrastructure**. In: Proceedings of the 2009 Simpósio Brasileiro de Sistemas Colaborativos, SBSC '09, p. 193–201. IEEE Computer Society, 2009.
- Ellis, C. A.; Gibbs, S. J. ; Rein, G. **Groupware: Some issues and experiences**. In: Communications of the ACM, p. 38–58. ACM, 1991.
- Erl, T. **SOA Principios De Design De Serviços**. Prentice Hall Brasil, 2008.
- Fuks, H.; Raposo, A. B.; Gerosa, M. A. ; Lucena, C. J. P. Applying the 3C model to groupware development. **International Journal of Cooperative Information Systems (IJCIS)**, v.14, n.2-3, p. 299–328, 2005.
- Fuks, H.; Pimentel, M. **Sistemas Colaborativos**. Elsevier, 2012.
- ISO2788. **Guidelines for the establishment and development of monolingual thesauri**, 1986.

- Kalinowski, M. **Uma abordagem para prevenção de defeitos provenientes de inspeções para apoiar a melhoria dos processos de engenharia do software.** 2011. Tese de Doutorado - Doutorado em Engenharia de Sistemas e Computação - Universidade Federal do Rio de Janeiro (UFRJ).
- Li, W.; Li, S. **Improve the semantic interoperability of information.** In: Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference, volume 2, p. 591–59, 2004.
- Maciel, R. S. P.; David, J. M. N. **WGWSOA: A service-oriented middleware architecture to support groupware interoperability.** In: Computer Supported Cooperative Work in Design, 2007. CSCWD 2007. 11th International Conference on, p. 556–561, 2007.
- Maciel, R. S. P.; David, J. M. N.; Oei, M. R.; de Oliveira Bastos, A. A. ; de Oliveira Menezes, L. Supporting awareness in groupware through an aspect-oriented middleware service. **Journal UCS**, v.15, p. 1945–1969, 2009.
- Mucheroni, M.; Silva, J. A interoperabilidade dos sistemas de informação sob o enfoque da análise sintática e semântica de dados na web. **Ponto de Acesso**, v.5, n.1, 2011.
- Papazoglou, M. P.; Traverso, P. Service-oriented computing: State of the art and research challenges. **IEEE Computer**, v.40, n.11, p. 38–45, 2007.
- Pohl, K.; Rupp, C. **Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB compliant.** Rocky Nook, 2011.
- Pokraev, S.; Reichert, M.; Steen, M. ; Wieringa, R. **Semantic and pragmatic interoperability: A model for understanding.** In: Proceedings of the Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability (EMOI - INTEROP'05), volume 160 de **CEUR Workshop Proceedings**, p. 1–5. CEUR-WS.org, 2005.
- Pollock, J.; Hodgson, R. **Adaptive Information: Improving Business Through Semantic Interoperability, Grid Computing, and Enterprise Integration.** Wiley Series in Systems Engineering and Management. Wiley, 2004.
- Schantz, R. E.; Schmidt, D. C. **Middleware for distributed systems - Evolving the common structure for network-centric applications**, 2001.
- Sommerville, I. **Software Engineering: (8th Edition).** 8. ed., Addison Wesley, 2006.
- Tolk, A.; Diallo, S. ; Turnitsa, C. Applying the levels of conceptual interoperability model in support of integratability, interoperability, and composability for system-of-systems engineering. **Systemics, Cybernetics and Informatics**, v.5, n.5, p. 65–74, 2007.
- Wang, W.; Tolk, A. ; Weiping, W. **The levels of conceptual interoperability model: applying systems engineering principles to mes.** In: SpringSim. SCS/ACM, 2009.
- Wohlin, C.; Runeson, P.; Höst, M.; Ohlsson, M.; Regnell, B. ; Wesslén, A. **Experimentation in Software Engineering.** Computer Science. Springer, 2012.

A APÊNDICE - Questionário de Caracterização do Participante

O questionário de caracterização do participante aplicado no estudo de caso possuía perguntas fechadas que seguiam a escala: muito pouco, pouco, regular, alto e muito alto. A seguir, estão descritas as perguntas contidas no questionário de caracterização:

- Dados Pessoais

Nome:

Idade:

E-mail:

Escolaridade:

Curso:

Instituição de Ensino:

- Questionário

1. Como você avaliaria seus conhecimentos em relação à linguagem Java?

() Muito Pouco

() Pouco

() Regular

() Alto

() Muito Alto

2. Como você avaliaria seus conhecimentos em relação ao desenvolvimento de sistemas?

() Muito Pouco

() Pouco

() Regular

☐ Alto

☐ Muito Alto

3. Como você avaliaria seus conhecimentos em relação ao desenvolvimento de sistemas distribuídos?

☐ Muito Pouco

☐ Pouco

☐ Regular

☐ Alto

☐ Muito Alto

4. Como você avaliaria seus conhecimentos em relação a tecnologias que utilizam conceitos de SOA?

☐ Muito Pouco

☐ Pouco

☐ Regular

☐ Alto

☐ Muito Alto

B APÊNDICE - Questionário de Avaliação do Estudo de Caso

O questionário teve suas perguntas divididas por tarefa e por requisito não funcional relacionado. Há respostas fechadas e abertas, sendo que as respostas fechadas seguem a escala *likert* com as opções: discordo fortemente, discordo, não deseja e/ou não se sente capaz de opinar, concordo e concordo fortemente. A descrição das perguntas pode ser encontrada a seguir:

- Tarefa 1 -Usar as descrições de dois serviços fornecidos para verificar informações sobre a interoperabilidade semântica. O participante deve primeiro fazer esta verificação usando os serviços ThesIS e OntologyMediator e depois, usando a funcionalidade SemanticMediationSupportView.

Sobre a capacidade de composição:

1) O uso da funcionalidade SemanticMediationSupportView facilitou a obtenção de informações relacionadas à interoperabilidade semântica, em comparação com a execução da mesma tarefa sem usá-la.

☐ Discordo fortemente

☐ Discordo

☐ Não deseja e/ou não se sente capaz de opinar

☐ Concordo

☐ Concordo fortemente

2) O uso da funcionalidade SemanticMediationSupportView facilitou a compreensão de informações relacionadas à interoperabilidade semântica, em comparação com a execução da mesma tarefa sem usá-la.

☐ Discordo fortemente

☐ Discordo

☐ Não deseja e/ou não se sente capaz de opinar

☐ Concordo

☐ Concordo fortemente

3) Houve menos esforço de programação ou consulta a código fonte para executar a tarefa usando a funcionalidade `SemanticMediationSupportView`, em comparação com a execução da mesma tarefa sem usá-la.

☐ Discordo fortemente

☐ Discordo

☐ Não deseja e/ou não se sente capaz de opinar

☐ Concordo

☐ Concordo fortemente

4) Para execução desta tarefa, qual forma você considerou mais vantajosa? Por que?

- Tarefa 2 - O objetivo desta tarefa é monitorar a interoperabilidade semântica durante a execução de algum serviço. Para isso, deve ser criado um serviço formado pela composição de outros dois serviços já existentes na WGWSOA: chat e lista de usuários. O serviço criado deve ser executado uma vez utilizando o serviço `SemanticExecutionSupport` e outra vez sem usar o serviço.

Sobre a reusabilidade:

1) Houve dificuldade na reutilização do serviço `SemanticExecutionSupport` para a implementação do serviço solicitado.

☐ Discordo fortemente

☐ Discordo

☐ Não deseja e/ou não se sente capaz de opinar

☐ Concordo

☐ Concordo fortemente

2) O uso do serviço SemanticExecutionSupport facilitou a execução da tarefa, se comparado a execução sem seu uso.

☐ Discordo fortemente

☐ Discordo

☐ Não deseja e/ou não se sente capaz de opinar

☐ Concordo

☐ Concordo fortemente

3) Para execução desta tarefa, qual forma você considerou mais vantajosa? Por que?

- Tarefa 3 - Criar um novo serviço na infraestrutura que simule uma nova forma de apoiar à interoperabilidade semântica entre serviço. Para a implementação do serviço simulador, o participante deve criar uma lógica que retorne aleatoriamente uma resposta positiva ou negativa sobre a interoperabilidade dos serviços candidatos. O serviço simulador deve ser adicionado ao serviço SemanticMediationSupport para constituir uma nova forma de interoperabilidade semântica abordada.

Com relação à escalabilidade:

1) Foi possível adicionar o serviço simulador solicitado ao serviço SemanticMediationSupport.

☐ Discordo fortemente

☐ Discordo

☐ Não deseja e/ou não se sente capaz de opinar

☐ Concordo

☐ Concordo fortemente

2) Houve dificuldade para adicionar o serviço simulador solicitado ao serviço SemanticMediationSupport.

☐ Discordo fortemente

☐ Discordo

() Não deseja e/ou não se sente capaz de opinar

() Concordo

() Concordo fortemente

• Com relação às dificuldades encontradas durante a realização das tarefas

1) Que problemas técnicos ocorreram durante o trabalho?

2) O treinamento dado foi suficiente para executar as tarefas? Justifique.

C APÊNDICE - Termo de Consentimento

Livre e Esclarecido

Pesquisa: “Solução Proposta para ampliar o suporte à interoperabilidade semântica em uma infraestrutura de middleware”

Prezado Senhor (a),

Infraestruturas de *middleware* têm sido largamente utilizadas para apoiar o desenvolvimento e a execução de sistemas colaborativos. Com estes sistemas diversos desafios surgiram, entre eles os relacionados à interoperabilidade semântica ganharam notável destaque. Este trabalho tem como objetivo ampliar o suporte à interoperabilidade semântica em uma infraestrutura chamada *Web-based Groupware Service-Oriented Architecture* (WGWSOA) que utiliza serviços de *middleware* para apoiar o desenvolvimento e a execução de sistemas colaborativos. O objetivo da pesquisa é analisar uma solução proposta com o propósito de avaliar o suporte à interoperabilidade semântica oferecido por ela com respeito a sua capacidade de composição, escalabilidade e reusabilidade do ponto de vista de usuários e desenvolvedores no contexto da realização de atividades que demandam interoperabilidade semântica na infraestrutura WGWSOA.

1 - Procedimento

O ambiente da WGWSOA será utilizado para o participante realizar algumas tarefas que utilizem a solução proposta para apoiar a interoperabilidade semântica. Para participar deste estudo solicito a sua especial colaboração em: (1) permitir que os dados resultantes da sua avaliação sejam estudados, (2) participar de- entrevista e/ou responder um questionário. Quando os dados forem coletados, seu nome será removido dos mesmos e não será utilizado em nenhum momento durante a análise ou apresentação dos resultados.

2 - Tratamento de possíveis riscos e desconfortos Serão tomadas todas as providências durante a coleta de dados de forma a garantir a sua privacidade e seu anonimato. Os dados coletados durante o estudo destinam-se estritamente a atividades de pesquisa relacionadas à solução proposta, não sendo utilizados em qualquer forma de

avaliação profissional ou pessoal.

3 - Benefícios e Custos

Espera-se que, como resultado deste estudo, você possa aumentar seu conhecimento sobre sistemas colaborativos, de maneira a contribuir para o aumento dos seus conhecimentos. Este estudo também contribuirá com resultados importantes para a pesquisa de um modo geral nas áreas de Engenharia de Software e Sistemas Colaborativos. Você não terá nenhum gasto ou ônus com a sua participação no estudo e também não receberá qualquer espécie de reembolso ou gratificação devido à participação na pesquisa.

4 - Confidencialidade da Pesquisa

Toda informação coletada neste estudo é confidencial e seu nome e o da sua organização não serão identificados de modo algum, a não ser em caso de autorização explícita para esse fim.

5 - Participação

Sua participação neste estudo é muito importante e voluntária. Você tem o direito de não querer participar ou de sair deste estudo a qualquer momento, sem penalidades. Em caso de você decidir se retirar do estudo, favor notificar um pesquisador responsável.

Os pesquisadores responsáveis pelo estudo poderão fornecer qualquer esclarecimento sobre o mesmo, assim como tirar dúvidas, bastando entrar em contato pelos seguintes *emails*:

Pesquisadora: Frâncila Weidt Neiva - fran.weidt@gmail.com

Professor orientador: José Maria Nazar David - jmndavid@gmail.com

6 - Declaração de Consentimento Li ou alguém leu para mim as informações contidas neste documento antes de assinar este termo de consentimento. Declaro que toda a linguagem técnica utilizada na descrição deste estudo de pesquisa foi explicada satisfatoriamente e que recebi respostas para todas as minhas dúvidas. Confirmando também que recebi uma cópia deste Termo de Consentimento Livre e Esclarecido. Compreendo que sou livre para me retirar do estudo em qualquer momento, sem qualquer penalidade. Declaro ter mais de 18 anos e dou meu consentimento de livre e espontânea vontade para participar deste estudo.

Local e Data:

Nome do Participante:

Assinatura do Participante:

Nome do Pesquisador:

Assinatura do Pesquisador:

D APÊNDICE - Protótipos da

Funcionalidade

SemanticMediationSupportView

Antes de partir para a fase de implementação da solução proposta, alguns protótipos foram elaborados para auxiliar no processo de desenvolvimento. Os protótipos, bem como suas descrições podem ser encontrados a seguir:

Interface de usuário 1: Nesta interface, representada pela Figura D.1, o usuário deve escolher o caminho dos arquivos que serão analisados, contendo os dados dos serviços candidatos a interoperabilidade. O usuário poderá incluir os arquivos .xml e/ou os arquivos .owl (Não é obrigatório o preenchimento das duas extensões de arquivo, ou seja, posso preencher apenas os arquivos .owl)

A interface é uma janela com o título "Verificação da Interoperabilidade Semântica". Ela contém duas seções principais para a seleção de arquivos. A primeira seção, para arquivos .xml, possui o rótulo "Desejo interoperar o serviço (.xml):" seguido de um campo de texto e um botão "Buscar", e o rótulo "Com o serviço (.xml):" seguido de outro campo de texto e botão "Buscar". Abaixo, há uma explicação: "O arquivo .xml é usado para interoperabilidade através do Thesis". A segunda seção, para arquivos .owl, possui o rótulo "Desejo interoperar o serviço (.owl):" seguido de um campo de texto e um botão "Buscar", e o rótulo "Com o serviço (.owl):" seguido de outro campo de texto e botão "Buscar". Abaixo, há uma explicação: "O arquivo .owl é usado para interoperabilidade através do Ontology Mediator". No rodapé da janela, há dois botões: "Interoperar" e "Cancelar".

Figura D.1: Verificação da Interoperabilidade Semântica

Após clicar no botão “interoperar” será exibido os resultados gerais de cada técnica e um gráfico comparativo entre elas, como pode ser visto na Figura D.2:

Ao clicar no botão “próximo” na interface com os dados da interoperabilidade semântica, será exibida uma outra interface de usuário para possibilitar uma análise re-

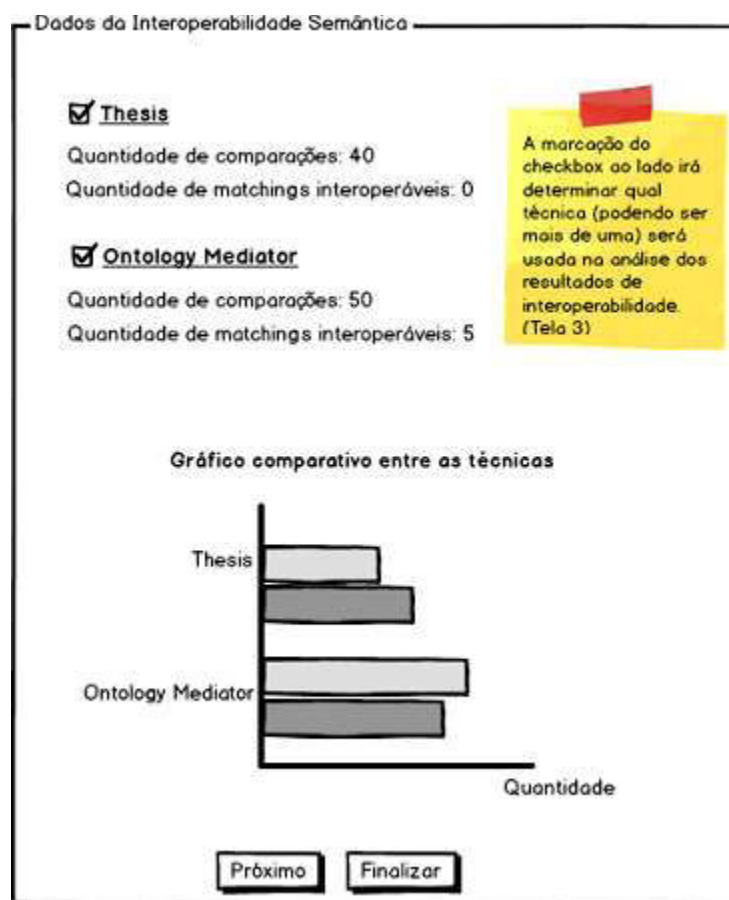


Figura D.2: Dados da interoperabilidade semântica

finada dos *matchings* interoperáveis de acordo com a técnica selecionada no *checkbox* da interface representada pela Figura D.2.

Análise de Conceitos

Conceitos interoperáveis segundo:

Thesis
Não existem conceitos interoperáveis.

Ontology Mediator
☒ Carro - Veículo
☐ Automóvel - Viatura
☐ etc. - etc.

Próximo Finalizar

Figura D.3: Dados da Interoperabilidade Semântica

Na interface de usuário representada na Figura D.3, o usuário poderá marcar quais *matchings* considerados interoperáveis pela técnica ele concorda. Ao clicar em próximo, será exibida uma quarta interface de usuário contendo tutoriais para auxiliar as atividades de desenvolvimento na WGWSOA.