

Universidade Federal de Juiz de Fora
Programa de Mestrado em Ciência da Computação

Tadeu Moreira de Classe

**UMA REDE PONTO A PONTO SEMÂNTICA PARA COMPOSIÇÃO DE SERVIÇOS
EM DOMÍNIOS CIENTÍFICOS**

Juiz de Fora
2014

**UMA REDE PONTO A PONTO SEMÂNTICA PARA COMPOSIÇÃO DE SERVIÇOS
EM DOMÍNIOS CIENTÍFICOS**

Tadeu Moreira de Classe

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Juiz de Fora, como parte dos requisitos necessários à obtenção do grau de Mestre em Ciência da Computação.

Orientador: Regina Maria Maciel Braga Villela

Juiz de Fora
Março de 2014

Tadeu Moreira de Classe

**UMA REDE PONTO A PONTO SEMÂNTICA PARA COMPOSIÇÃO DE SERVIÇOS
EM DOMÍNIOS CIENTÍFICOS**

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre.

Aprovada em 13 de março de 2014.

BANCA EXAMINADORA

Prof.^a Dr.^a Regina Maria Maciel Braga Villela – Orientadora
Universidade Federal de Juiz de Fora

Prof. Dr. José Maria Nazar David
Universidade Federal de Juiz de Fora

Prof. Dr. Marco Antônio Pereira Araújo
Instituto Federal de Educação, Ciência e Tecnologia
do Sudeste de Minas Gerais

Prof.^a Dr.^a Renata Mendes de Araujo
Universidade Federal do Estado do Rio de Janeiro

Para as pessoas que amo.

AGRADECIMENTOS

A Deus por me dar a sabedoria e serenidade necessária para encontrar a solução em momentos que mais precisei.

Aos meus familiares e minha namorada por estarem sempre ao meu lado, incentivando, tendo paciência e acreditando em meu potencial em momentos em que eu mesmo achava que não havia mais o que fazer.

A minha orientadora Prof.^a Regina Braga pelo apoio, paciência, dedicação e incentivo ao decorrer deste Mestrado.

Aos professores José Maria David e Fernanda Campos pelo apoio e dedicação, ajudando na correção dos artigos e dando conselhos.

Aos membros da Banca Examinadora pela avaliação deste trabalho.

E a todos que contribuíram, direta ou indiretamente, na realização deste trabalho.

Resumo da Dissertação apresentada à Universidade Federal de Juiz de Fora como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciência da Computação

Grandes mudanças estão acontecendo no contexto de pesquisas científicas. Em particular, existe uma maior colaboração entre os pesquisadores, o que leva ao aumento do uso de técnicas de processamento e a necessidade de compartilhar resultados e observações, e a partir disso, construir e executar modelos científicos. Entretanto, encontrar e/ou especificar aplicações científicas não é tão trivial e muitos cientistas não possuem o conhecimento computacional necessário para compor seus estudos experimentais.

Nesse sentido, utilizando tecnologias como web semântica e redes ponto a ponto aplicados ao contexto de e-Science, o objetivo desta dissertação é desenvolver uma plataforma de acesso onde os pesquisadores possam trabalhar com informações heterogêneas, acessando comunidades científicas de acordo com seus interesses, e criar novos experimentos baseados na composição de serviços web semânticos.

Este trabalho apresenta a abordagem e-ScienceNet, que é uma rede ponto a ponto semântica para auxiliar os cientistas em suas pesquisas. A partir da proposta arquitetural, um protótipo foi especificado e utilizado em quatro provas de conceito.

Por meio das provas de conceito foi possível obter indícios da viabilidade do uso da e-ScienceNet, auxiliando e facilitando o desenvolvimento de experimentos científicos pelos cientistas, fornecendo ainda, mecanismos de compartilhamento e pesquisa de recursos. Porém, uma avaliação criteriosa em um ambiente real ainda é necessária.

Abstract of Dissertation presented to Federal University of Juiz de Fora as a partial fulfillment of the requirements for the degree of Master of Computer Science

Significant changes are occurring in scientific scenario. In particular, there is a higher collaboration among the researchers, which leads to increase the use of processing techniques and the necessity of sharing results and observations, and with that, building and executing scientific experiments. However, finding and/or specifying scientific applications is not trivial and many scientists lack the computational knowledge necessary to compose their experiments

Using technologies as semantic web and peer to peer networks in the e-Science context, the objective of this work is to develop a platform where the researchers can work with heterogeneous information, accessing scientific communities according to their interests, and creating new experiments based on composition of semantic web services.

This work presents the e-ScienceNet approach, which is a semantic peer to peer network, to help scientists in their research. Considering the architectural model, a prototype was specified, which was used in four "Proof of Concept" (PoC). These proofs of concepts aims to obtain evidences of the viability of the use of this proposal on scientific scenarios.

From these proofs of concepts, it was possible to obtain evidences of the viability of use of the e-ScienceNet, assisting and facilitating the development of scientific experiments while still providing mechanisms for sharing and searching for resources. However, an evaluation in a real environment is still needed.

LISTA DE FIGURAS

Figura 2.1: Ontologia Service [OWL-S, 2014].....	28
Figura 2.2: Elementos de uma rede de Petri [DA SILVA, BRAGA e CAMPOS, 2012]	32
Figura 2.3: Roteamento sequencial [VAN DER AALST, 1998].....	34
Figura 2.4: Roteamento paralelo [VAN DER AALST, 1998]	34
Figura 2.5: Roteamento condicional [VAN DER AALST, 1998]	35
Figura 2.6: Roteamento interativo [VAN DER AALST, 1998].....	35
Figura 3.1: Arquitetura da e-ScienceNet	50
Figura 3.2: Arquitetura Funcional da e-ScienceNet	52
Figura 3.3: Tela Principal da e-ScienceNet	53
Figura 3.4: Tela de Visualização dos Peers Conectados no Grupo.....	54
Figura 3.5: Menu Principal da e-ScienceNet	54
Figura 3.6: Arquitetura Funcional do Gerente de Ferramentas.....	55
Figura 3.7: Tela de Configuração do JXTA na e-ScienceNet	56
Figura 3.8: Gerente de Semântica	57
Figura 3.9: A) PeerOntology modelo criado. B) PeerOntology modelo inferido	58
Figura 3.10: Informações sobre o peer contida na PeerOntology.....	59
Figura 3.11: Exemplo da PeerOntology de um nó	59
Figura 3.12: Exemplo de consulta SPARQL para verificar peers.	60
Figura 3.13: Fluxo de Controle do Gerente de Interesse.....	61
Figura 3.14: Pesquisa semântica de grupos de interesse	62
Figura 3.15: Parte do código de busca de grupos semânticos	63
Figura 3.16: Interface de seleção de grupo semântico	64
Figura 3.17: Criação de novos grupos semânticos.....	64
Figura 3.18: Ontologias de domínio disponíveis para download no Super nó da comunidade semântica.....	65
Figura 3.19: Cadastro de nova ontologia de domínio e upload para o super nó.....	66
Figura 3.20: Download e Upload de Ontologias de Domínio	67
Figura 3.21: Comunicação do Gerente de Dados	68
Figura 3.22: Interface de inclusão de documentos	68
Figura 3.23: Informações sobre arquivo persistido na PeerOntology	69
Figura 3.24: Interface de gerenciamento de arquivos.....	69

Figura 3.25: Modelo de Comunicação do Gerente de Serviço	70
Figura 3.26: Interface de inclusão de serviços web	71
Figura 3.27: Anotações	72
Figura 3.28: A) Anotação semântica por termos sugeridos. B) Anotação semântica por seleção manual	72
Figura 3.29: Anotação semântica de parâmetros	73
Figura 3.30: Anotação de parâmetros por XSD.....	73
Figura 3.31: Código de verificação de consistência de OWL-S.....	74
Figura 3.32: Estrutura OWL-S de um serviço incluso na e-ScienceNet.....	75
Figura 3.33: Interface de gerenciamento de serviços disponíveis no peer.	75
Figura 3.34: Data Property domainTerm no OWL-SE	76
Figura 3.35: Data Property serviceRealName no OWL-SE	77
Figura 3.36: Data Property parameterTypeXDS no OWL-SE.....	77
Figura 3.37: Classe SubParameter no OWL-SE	78
Figura 3.38: hasSubParameter e isSubParameterOf da OWL-SE	78
Figura 3.39: Interface de anotação semântica de sub-parâmetros	79
Figura 3.40: A) Sub-Parâmetros do parâmetro parameters. B) Sub-Parâmetro do sub-parâmetro database.....	80
Figura 3.41: Inclusão de conector na e-ScienceNet.....	81
Figura 3.42: Associação de conector com um serviço web	82
Figura 3.43: Modelo de comunicação do Gerente de Pesquisa	82
Figura 3.44: Query SPARQL para consultas de arquivos na PeerOntology	83
Figura 3.45: Resultado de um busca de arquivos pelo Gerente de Pesquisa.....	84
Figura 3.46: Variações dos termos de pesquisa através da inferência na ontologia de domínio.....	85
Figura 3.47: Serviços retornados pela busca	85
Figura 3.48: Fases de execução do Gerente de Composição	87
Figura 3.49: Criação do modelo de workflow abstrato.....	89
Figura 3.50: Interface de associação da conexão lógica JOIN	90
Figura 3.51: Diferenças entre os parâmetros de um serviço antes e depois de se unir a um conector.....	91
Figura 3.52: Conectores na e-ScienceNet.....	92
Figura 3.53: Associação de parâmetro do serviço com o conector.....	92
Figura 3.54: Conexão de parâmetros entre serviços e inclusão de valores.....	93

Figura 3.55: (A) Inclusão de valores manuais em um parâmetro. (B) Conexão de múltiplas saídas em uma entrada	93
Figura 3.56: Interface de criação de restrições para parâmetros.....	94
Figura 3.57: Execução do Workflow Científico	95
Figura 3.58: Código para a execução de um web service	95
Figura 3.59: Visualização de resultados obtidos na execução do workflow.....	96
Figura 3.60: Execução de serviço único.....	97
Figura 4.1: Esquema de divisão dos peers nos grupos semânticos no ambiente...	102
Figura 4.2 A) Busca de arquivo Tadeu_PEER. B) Busca de arquivo VIRTUALXP .	102
Figura 4.3: Ambiente colaborativo para Mathematics.....	103
Figura 4.4: A) Representação geométrica. B) Representação algébrica	104
Figura 4.5: SimpleMathOntology	105
Figura 4.6: Modelo Inferido nas operações de SimpleMathOntology.....	105
Figura 4.7: Restrição workflow_task entre classes MathFormule e MathOperations	106
Figura 4.8: Operações relacionadas ao teorema de Pitágoras.....	106
Figura 4.9: Anotação semântica do serviço Read	107
Figura 4.10: Anotação semântica do parâmetro return do serviço Read	107
Figura 4.11: Inclusão do serviço Pot na e-ScienceNet	108
Figura 4.12: Anotação semântica do serviço Pot	108
Figura 4.13: Inserindo o conector SplitNumbers.....	110
Figura 4.14: Inclusão da tarefa Read no AWF	111
Figura 4.15: inclusão das tarefas Pot e sua conexão AND-SPLIT com a tarefa Read	112
Figura 4.16: Inclusão da tarefa Sum e a conexão AND-JOIN entre os catetos	112
Figura 4.17: Inclusão da tarefa Equals e a conexão AND-JOIN com a hipotenusa	113
Figura 4.18: Finalização da composição de Pitágoras	113
Figura 4.19: Carregando o modelo abstrato no gerente de pesquisa de serviços ..	114
Figura 4.20: Busca dos serviços relacionados a tarefa Read	115
Figura 4.21: Busca do conector Split para a tarefa Read	115
Figura 4.22: Busca de serviços para a tarefa Pot.....	116
Figura 4.23: Busca de serviços relacionados a tarefa Sum.....	116
Figura 4.24: Busca de serviços relacionados a tarefa Equals	117
Figura 4.25: Configuração das conexões entre os serviços.	118

Figura 4.26: Execução da composição de Pitágoras.....	119
Figura 4.27: Execução do workflow funcional de Pitágoras em outro peer da e- ScienceNet	120
Figura 4.28: Workflow para alinhamento de sequencias proposto por DA SILVA, BRAGA e CAMPOS [2012].....	122
Figura 4.29: LaryssaSampe workflows_possible_tasks em SequenceAligningOntology	123
Figura 4.30: Anotação semântica para a tarefa GetSequence	124
Figura 4.31: Anotação semântica para Read and Format Sequences.....	125
Figura 4.32: Anotação semântica para o serviço de Alignment	126
Figura 4.33: Inclusão do conector getStatus para Clustal Omega	127
Figura 4.34: Inclusão do conector getResult para Clustal Omega.	128
Figura 4.35: Inclusão da tarefa Aligning e finalização do modelo abstrato	129
Figura 4.36: Ambiente colaborativo para Bioinformatics.....	130
Figura 4.37: Inferência de serviços relacionados com a tarefa <i>aligning</i>	130
Figura 4.38: Busca dos serviços relacionados a tarefa <i>getSequence</i>	131
Figura 4.39: Busca do serviço relacionado a tarefa <i>read and format sequences</i>	131
Figura 4.40: Busca de serviço relacionado a tarefa <i>aligning</i>	132
Figura 4.41: Busca do conector <i>getStatus</i> para a tarefa <i>aligning</i>	132
Figura 4.42: Busca do conector <i>getResult</i> para a tarefa <i>aligning</i>	133
Figura 4.43: Inclusão estrutura de repetição no conector getStatus	134
Figura 4.44: Resultado final do workflow gerado pela proposta de DA SILVA, BRAGA e CAMPOS [2012]	135
Figura 4.45: Workflow Blast, Clustal, EMBOSS gerado pelo Taverna	137
Figura 4.46: Modelo de conexão de web services.....	138
Figura 4.47: Pesquisa para o serviço blast.....	139
Figura 4.48: Conector de entrada para o serviço blast (fetchBatch)	140
Figura 4.49: Conexão e inclusão de valores nos parâmetros dos serviços	141
Figura 4.50: Resultado final de Blast, Clustal e Emboss	141

LISTA DE TABELAS

Tabela 1: Comparação de Trabalhos Relacionados.....	46
Tabela 2: Parâmetros do serviço Read	107
Tabela 3: Parâmetros do Serviço Pot.....	109
Tabela 4: Parâmetros do serviço Sum	109
Tabela 5: Parâmetros do serviço Equals.....	109
Tabela 6: Parâmetros de SplitNumbers.....	110
Tabela 7: Conexão entre parâmetros e valores manuais	118
Tabela 8: Parâmentros de GetSequence	124
Tabela 9: Parâmetros de Read and Format Sequences.....	125
Tabela 10: Parâmetros de Clustal Omega	126
Tabela 11: Conexão entre os parâmetros do serviço	133

LISTA DE ABREVIações

API	APPLICATION PROGRAMMING INTERFACE
DDBJ	DNA DATA BANK OF JAPAN
DHT	DATA HASH TABLE
EBI	EUROPEAN BIOINFORMATICS INSTITUTE
GQM	GOAL/QUESTION/METRIC
HTTP	HYPertext TRANSFER PROTOCOL
IP	INTERNET PROTOCOL
KEGG	KYOTO ENCYCLOPEDIA OF GENES AND GENOMES
NCBI	NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION
NENC	NÚCLEO DE ENGENHARIA DO CONHECIMENTO
NGS	NEXT-GENERATION SEQUENCING
OWL	WEB ONTOLOGY LANGUAGE
P2P	PONTO A PONTO OU PEER TO PEER
PDB	PROTEIN DATA BANK
PIR	PROTEIN INFORMATION RESOURCE
POC	PROVA DE CONCEITO OU PROOF OF CONCEPT
RDF	RESOURCE DESCRIPTION FRAMEWORK
SGWC	SISTEMA DE GERENCIALMENTO DE WORKFLOWS CIENTÍFICOS
SOA	SERVICE-ORIENTED ARCHITECTURE
SOAP	SIMPLE OBJECT ACCESS PROTOCOL
TCP	TRANSMISSION CONTROL PROTOCOL
UDDI	UNIVERSAL DESCRIPTION, DISCOVERY AND INTEGRATION
UFJF	UNIVERSIDADE FEDERAL DE JUIZ DE FORA
URI	UNIFORM RESOURCE IDENTIFIER
URL	UNIFORM RESOURCE LOCATOR
W3C	WORLD WIDE WEB CONSORTIUM
WSDL	WEB SERVICES DESCRIPTION LANGUAGE
WSMO	WEB SERVICE MODELING ONTOLOGY
XML	EXTENSIBLE MARKUP LANGUAGE
XSD	XML SCHEMA DEFINITION

SUMÁRIO

1	INTRODUÇÃO	16
1.1	MOTIVAÇÃO	16
1.2	JUSTIFICATIVA	16
1.3	OBJETIVOS	18
1.4	ESTRUTURA DO TRABALHO	19
2	PRESSUPOSTOS TEÓRICOS	20
2.1	E-SCIENCE E SOFTWARE CIENTÍFICO.....	20
2.2	REDES PONTO A PONTO.....	23
2.3	WEB SEMÂNTICA.....	25
2.4	SERVIÇOS WEB SEMÂNTICOS.....	27
2.5	COMPOSIÇÃO DE SERVIÇOS.....	29
2.5.1	Redes de Petri	31
2.6	BIOINFORMÁTICA: SEQUENCIAMENTO E ALINHAMENTO GENÉTICO	35
2.7	TRABALHOS RELACIONADOS.....	39
2.7.1	Análise dos Trabalhos Relacionados	41
2.8	CONSIDERAÇÕES FINAIS DO CAPÍTULO	46
3	E-SCIENCENET	48
3.1	INTRODUÇÃO	48
3.2	A ARQUITETURA DA E-SCIENCENET	49
3.2.1	Gerente De Ferramentas (Tool Manager).....	52
3.2.2	Gerente De Semântica (Semantic Manager)	57
3.2.3	Gerente De Interesse (Interest Manager)	60
3.2.3.1	Grupos Semânticos	61
3.2.3.2	Ontologias De Domínio.....	65
3.2.4	Gerente De Recursos (Resource Manager).....	67
3.2.5	Gerente De Dados (Data Manager).....	67
3.2.6	Gerente De Serviços (Service Manager)	69
3.2.6.1	Owl-Sextended	76
3.2.6.2	Conectores	80

3.2.7	Gerente De Pesquisa (Search Manager).....	82
3.2.8	Gerente De Composição (Composition Manager)	86
3.3	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	97
4	PROVA DE CONCEITO: USO DA E-SCIENCENET EM CENÁRIOS CIENTÍFICOS	99
4.1	CONTEXTO GERAL.....	100
4.2	PROVA DE CONCEITO 1	102
4.3	PROVA DE CONCEITO 2	121
4.4	PROVA DE CONCEITO 3	136
4.5	ANÁLISE DAS PROVAS DE CONCEITO.....	142
4.6	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	143
5	CONCLUSÕES E TRABALHOS FUTUROS.....	145
	REFERÊNCIAS	149
	APÊNDICE A – XML DE COMUNICAÇÃO DE GRUPOS SEMÂNTICOS.....	159
	APÊNDICE B – ALGORITMO DE INFERÊNCIA DE TERMOS RELACIONADOS PARA A PESQUISA DE SERVIÇOS.....	160
	APÊNDICE C – BUSCA DE INFORMAÇÕES SOBRE SERVIÇOS NA OWL-SE .	161
	APÊNDICE D – MENSAGEM SOAP PARA REQUISIÇÃO DE WEB SERVICE FETCHBATCH	163
	APÊNDICE E – MENSAGEM SOAP DE RESPOSTA DO WEB SERVICE FETCHBATCH	164

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Nos últimos anos o contexto de pesquisas científicas vem sofrendo mudanças substanciais. Em particular, existe uma maior colaboração entre grupos de pesquisadores, o que leva a um aumento no uso de técnicas de processamento de informação, e, com isso, a necessidade de compartilhar resultados e observações entre os participantes de uma pesquisa [MEDJAHED E BOUGUETAYA, 2011]. Entretanto, o uso de recursos computacionais em caráter científico não é um processo simples, pois, em geral, os cientistas de diferentes áreas não possuem habilidades computacionais adequadas e muitas vezes, por falta de conhecimento, não compartilham e nem processam colaborativamente seus experimentos. Além disso, existe também uma grande dificuldade em encontrar aplicações científicas adequadas para especificar as tarefas necessárias na execução de experimentos [DA SILVA, BRAGA e CAMPOS, 2012].

Para a execução de um dado experimento, geralmente são necessários um conjunto de aplicativos/algoritmos/serviços. No entanto, encontrar esses artefatos é uma tarefa cada vez mais complexa. Além disso, é necessária a composição correta destes aplicativos para a execução do experimento. Assim, analisando a necessidade de pesquisa por aplicações científicas e sua composição de um experimento, constata-se que para facilitar o processo de criação dessas ferramentas, um suporte semântico pode ser adequado [COSTA et al., 2013].

1.2 JUSTIFICATIVA

Devido a características de distribuição, baixo custo de implementação e possibilidade de extensão, as redes ponto a ponto vêm sendo consideradas como uma abordagem adequada para o suporte às aplicações científicas.

Aliadas a tecnologias da web semântica como ontologias, motores de inferência, serviços web semânticos e linguagens de consultas semânticas, por exemplo, SPARQL, entre outros, as redes ponto a ponto podem facilitar o acesso ao conhecimento necessário para o processamento adequado de um experimento científico.

A junção destas tecnologias permite, portanto, a especificação de redes ponto a ponto semânticas, viabilizando a recuperação de recursos, como documentos, serviços e etc. considerando a semântica, englobando assim o significado que estes recursos têm em um dado domínio de aplicação, através da anotação destes, com termos ontológicos. Neste contexto, podem-se agregar serviços web como recursos disponíveis. Tais serviços podem ser semanticamente anotados através de termos das ontologias de domínios, sendo gerados os serviços web semânticos. Assim, o uso de redes ponto a ponto e da web semântica, podem fornecer o suporte necessário para facilitar o processo de desenvolvimento de experimentos científicos colaborativos.

A hipótese levantada neste trabalho é que “a especificação de uma infraestrutura distribuída, baseada na tecnologia de redes ponto a ponto e web semântica, pode facilitar o desenvolvimento de experimentos científicos colaborativos”.

A partir dessa hipótese, pretende-se utilizar a abordagem de redes ponto a ponto, a qual se torna responsável por gerar os mecanismos de distribuição de recursos, apoiados em web semântica, responsável por prover conhecimento e auxiliar no desenvolvimento de composições. Deste modo, uma arquitetura assim se torna necessária na medida em que encontrar artefatos para o desenvolvimento de experimentos nem sempre é uma tarefa simples. Neste sentido, a arquitetura e-ScienceNet é apresentada para fornecer um ambiente integrado, baseado em rede ponto a ponto semântica, onde os cientistas consigam contribuir entre si na confecção de experimentos científicos e no compartilhamento de recursos, colaborando entre si no desenvolvimento da ciência.

1.3 OBJETIVOS

Baseando-se nos benefícios que a utilização de redes ponto a ponto semânticas pode oferecer para facilitar o desenvolvimento de experimentos científicos e o compartilhamento de conhecimento entre os pesquisadores, essa dissertação tem como objetivo principal propor uma arquitetura de apoio ao processamento de experimentos científicos distribuídos, como suporte a especificação dos chamados laboratórios colaborativos [OLSON et al., 2009].

Segundo OLSON et al. [2009], laboratórios colaborativos são uma importante e emergente infraestrutura computacional a qual permite a distribuição geográfica de projetos de pesquisa. Diferente dos laboratórios tradicionais, os quais dependem de uma estrutura física como equipamentos de pesquisa e profissionais em um mesmo ambiente, os laboratórios colaborativos, independente de sua localização, tempo e estado, conseguem que os cientistas compartilhem informações, troquem resultados e utilizem ferramentas de maneira distribuída, criando novas possibilidades para o trabalho científico [FINHOUT, 2002].

Assim, essa arquitetura, denominada e-ScienceNet provê mecanismos para que os cientistas possam pesquisar recursos para a criação de experimentos.

O objetivo principal deste trabalho pode ser subdividido em:

- i) Propor uma arquitetura de redes ponto a ponto semântica para fornecer suporte a *e-Science*;
- ii) Especificar experimentos científicos distribuídos;
- iii) Implementar mecanismos de pesquisa por serviços web baseados em modelos semânticos;
- iv) Possibilitar a execução de experimentos científicos;
- v) Fornecer resultados e compartilhar modelos que possam ser adaptados e reutilizados pelos pesquisadores em outros experimentos.

Diante de tais elementos, foi desenvolvido um protótipo para teste das ideias propostas na arquitetura. E, a partir desse protótipo, foram especificadas e consideradas três provas de conceito.

1.4 ESTRUTURA DO TRABALHO

Este trabalho encontra-se organizado em 4 capítulos, além desta introdução.

No capítulo 2, são apresentados os principais conceitos relacionados a esta dissertação como: *e-Science*, software científico, redes ponto a ponto e web semântica. Além disso, este capítulo discute alguns trabalhos relacionados.

No capítulo 3, é apresentada a arquitetura da *e-ScienceNet*. Neste capítulo são enfatizados cada um dos gerentes e as principais características de cada um deles.

No capítulo 4 são apresentadas quatro Provas de Conceito, com o intuito de se obter indícios sobre a viabilidade da proposta.

Finalmente, no capítulo 5 são apresentadas as conclusões, bem como indicações para trabalhos futuros.

2 PRESSUPOSTOS TEÓRICOS

Neste capítulo serão apresentadas as principais tecnologias relacionadas ao contexto deste trabalho. A seção 2.1 detalha conceitos relacionados a *e-Science*, software científico e bioinformática, considerando o subdomínio de alinhamento e o sequenciamento de DNA. Na seção 2.2, são delineados os conceitos relacionados a redes ponto a ponto. A seção 2.3 detalha web semântica, com ênfase no uso de ontologias. Na seção 2.4, serviços web semânticos são apresentados e na seção 2.5, a composição de serviços é abordada além do detalhamento do formalismo de redes de Petri (seção 2.5.1). Na seção 2.6, são discutidos os principais trabalhos relacionados a proposta, bem como é realizada uma análise comparativa com a *e-ScienceNet* (2.6.1). Por fim, na seção 2.7 são assinaladas as considerações finais do capítulo.

2.1 E-SCIENCE E SOFTWARE CIENTÍFICO

Na medida em que as técnicas computacionais evoluem, diversas áreas da ciência se beneficiam, uma vez que se considera a possibilidade de realização de experimentos computacionais ao invés de experimentos físicos, ou seja, aqueles que precisam de laboratórios e bancadas para a sua realização. Porém, à medida que esse conhecimento avança, um grande volume de informação é gerado, necessitando de maior poder computacional para processá-la [MATTOSO et al., 2008], além disso, cabe ressaltar a necessidade do conhecimento de técnicas computacionais pelos cientistas.

Na Engenharia de Software é fundamental o uso de modelos e metodologias para o desenvolvimento de aplicações computacionais. Entretanto, quando se considera o desenvolvimento de aplicações científicas, o uso de modelos e metodologias nem sempre é enfatizado. Os desenvolvedores de aplicações científicas argumentam que estas, são desenvolvidas para domínios específicos e, portanto, nem todas as técnicas disponíveis para o desenvolvimento de software, de uma maneira geral, trazem resultados satisfatórios [SLETHOLT et al., 2011]. Isso faz com que pesquisas sobre

técnicas relacionadas à Engenharia de Software no contexto de software científico sejam importantes.

De tal modo, cabe assinalar que Software científicos são complexos, geralmente utilizados para modelar, simular ou testar uma teoria, podendo evoluir ao longo dos anos, fazendo com que novas funcionalidades sejam sempre acrescentadas [COSTA et al., 2013]. Neste contexto, SOUZA [2011 apud COSTA et al., 2013] ressalta que os softwares científicos são diferentes dos softwares tradicionais devido aos seguintes aspectos: 1) há informalidade no processo de desenvolvimento de um software científico; 2) geralmente são os próprios cientistas que desenvolvem o software; e 3) o levantamento e a especificação de requisitos não são apresentados de forma clara, ou são desconhecidos no momento inicial da pesquisa.

O termo *e-Science* foi apresentado no Reino Unido para propor um novo meio de fazer ciência. HEY e TREFETHEN [2003] destacam que: “*e-Science* é uma colaboração global em áreas chave da ciência junto com a próxima geração de infraestrutura capaz de suportá-la”. Outro autor definiu *e-Science* enquanto uma reunião de conceitos relacionados a integração de computação em pesquisas científicas [HENDLER, 2003]. No entanto, considera-se que *e-Science* vai além deste conceito, visto que, se caracteriza pelo acesso a vastas coleções de dados, uso computacional em grande escala, heterogeneidade de recursos de diferentes repositórios, reusabilidade e uso de *workflows* [DA SILVA, BRAGA e CAMPOS, 2012].

Considerando a necessidade de distribuição de processamento e a integração de conhecimento, tecnologias como “*grids*” e redes ponto a ponto são apontadas como possíveis soluções para suportar ambientes compartilhados de experimentos computacionais, aliados a tecnologias como web semântica [CLASSE et al, 2013a].

De acordo com HENDLER [2003] existem alguns requisitos para uma arquitetura distribuída no contexto da *e-Science*: 1) Armazenamento: o cientista deve ser habilitado a armazenar e processar grandes volumes de dados independentemente de sua localização geográfica; 2) Gerenciamento de Propriedades: o cientista deve ser habilitado a manter seus próprios dados e serviços, fornecendo recursos apenas depois que outro cientista aceitar o seus

termos de uso; 3) Transparência: o cientista deve ser habilitado a descobrir, acessar e processar dados de maneira transparente, independentemente de onde esteja armazenado; 4) Comunidades: o cientista deve criar, manter e disponibilizar comunidades, sendo essas restritas ou não. Para isso, ele deve criar regras indicando as permissões de uso dentro delas; 5) Segurança: o cientista deve ser habilitado para compartilhar seus dados em segurança por meio de mecanismos de encriptação e autenticação, além do uso de privacidade de dados; 6) Mobilidade: o cientista deve ter acessibilidade a seus dados a partir de qualquer computador; 7) *Workflows*: o cientista deve ser capaz de criar e descrever um processo científico de forma clara e computacionalmente processável; 8) Proveniência: informações suficientes devem ser armazenadas em tempo de execução do *workflow*, fornecendo evidências concisas dos dados gerados e possibilitando a reprodução do experimento científico; 9) Notificação: os cientistas devem receber notificações sobre resultados, e novos recursos disponíveis de acordo com o seu interesse; 10) Decisão: os sistemas devem ser capazes de fornecer informações e sugestões para os cientistas de acordo com seu interesse; 11) Expansão: os sistemas devem prover suporte para o crescimento de sua infraestrutura; e 12) Componentes: os sistemas devem ser capazes de fazer uso de componentes no intuito de personalizar a necessidade de cada cientista.

Neste contexto, uma das áreas que necessitam de grande atenção é a bioinformática, pois muitos de seus problemas e oportunidades surgem da dependência de processos computacionais complexos. Diversas são as pesquisas neste campo, muitas delas envolvendo a composição de serviços para alinhamentos genéticos, processamentos de cadeias e outros [KONA et al., 2008] [CLASSE et al., 2013] [DA SILVA, BRAGA e CAMPOS, 2012], pois existem várias experiências que podem ser realizadas; A proveniência de dados em *workflows* científicos também é tema de muitos trabalhos [GREENWOOD et al., 2003] [STEVENS, ROBINSON e GOBLE, 2003] [ZHAO et al., 2008], os quais envolvem a captura de informações para a reprodutibilidade das experiências. Além disso, com o tempo, surgem ferramentas para análise de material genético [KUMAR e DUDLEY, 2007] além

de abordagens para conexão entre ferramentas já existentes [HEATH e RAMAKRISHNAN, 2002] [HOLLAND et al. 2008] [KUMAR et al. 2007].

2.2 REDES PONTO A PONTO

A tecnologia P2P (*peer to peer*) ou ponto a ponto, surgiu com uma proposta de mudança do já existente paradigma de cliente e servidor, fazendo com que cada computador integrante da rede (nós ou *peers*) tenha ao mesmo tempo o comportamento tanto de uma aplicação cliente, quanto o de um sistema servidor [ROCHA et al., 2004]. As redes ponto a ponto são consideradas sistemas computacionais comunitários, ou seja, são sistemas em que a informação é compartilhada entre seus *peers* de modo descentralizado [FREIRE, FERREIRA e FURTADO, 2006]. Tal sistema permite que recursos computacionais, arquivos, serviços e informações sejam compartilhados com os outros nós da rede através da Internet [ROCHA et al., 2004], [TANENBAUM e VAN STEEN, 2006]. Por meio desta tecnologia, qualquer dispositivo pode acessar recursos disponíveis nos outros *peers* sem que exista a necessidade de interferência de controladores centrais [ROCHA et al., 2004]. Assim, uma rede P2P pode conter diferentes equipamentos, com os mais distintos sistemas e arquiteturas computacionais, os quais são interconectados por uma rede heterogênea [DIAS et al., 2011].

ROCHA [2004] destaca em seu trabalho uma característica básica dos sistemas de rede ponto a ponto, que é a possibilidade ou não de formação de grupos de nós com os mesmos interesses. Esses grupos, baseando-se nos modelos de apresentação da rede, podem ser categorizados como: totalmente descentralizados, são sistemas onde não existe nenhum tipo de controlador e cada um dos *peers* possuem a mesma responsabilidade, sendo eles autônomos e responsáveis pela troca de recursos entre si; e semi-centralizado: são sistemas em que existe um *peer* (ou conjunto de *peers*) central responsável por algum tipo de controle sobre os demais. Este tipo especial de *peer* é denominado **super nó** (*super peers*) de uma rede P2P.

Outro tipo de classificação de um sistema ponto a ponto é pelo tipo de pesquisa por informações entre os *peers*. Para isso ROCHA [2004] destaca:

buscas centralizadas, são sistemas onde existe um *peer* (geralmente o super nó), onde os demais *peers* realizam consultas e trocam informações sobre recursos disponíveis na rede; buscas por inundação: são sistemas onde não existe controlador central, porém a busca é limitada entre os vizinhos, ou seja, cada *peer* conhece os recursos do seu respectivo vizinho, e vão passando a informação assim por diante; e busca por tabelas hash distribuídas ou busca por DHT (*Distributed Hash Table*), são sistemas onde os nós são autônomos e utilizam uma tabela hash para compartilhar recursos entre si.

O controle descentralizado entre os componentes da rede e o grande poder de processamento e compartilhamento de recursos, fizeram com que a proposta do usos de redes ponto a ponto se tornasse atraente para uma vasta gama de sistemas distribuídos [MAWLOOD-YUNIS, WEISS e SANTORO, 2011]. A partir desta característica e do crescimento do número de computadores domésticos, tais sistemas se popularizaram, estimulando as pessoas envolvidas a compartilharem cada vez mais recursos, arquivos e documentos umas com as outras. Tal popularização pode ser comprovada através de um retrospecto a alguns sistemas de compartilhamento de arquivos e mensagem que foram (ou são) sucesso entre os usuários da Internet. Sistemas como: KaZaa¹, eMule², Napster³, ICQ⁴, BitTorrent⁵ e outros tantos existentes, são exemplos da popularização da troca de arquivos na Internet [ROCHA et al., 2004].

É possível destacar, basicamente, a existência de três tipos de sistemas de redes ponto a ponto, sendo eles: 1) redes ponto a ponto com a finalidade para o compartilhamento de arquivos entre *peers* (documentos, músicas, vídeos e etc.); 2) redes ponto a ponto baseadas na disseminação de buscas por palavras-chave; e 3) redes ponto a ponto baseada em consultas semânticas de recursos [MODICA, TOMARCHIO e VITA, 2011]. Este terceiro tipo de redes P2P é no qual se baseia a proposta deste trabalho.

¹ <http://www.kazaa.com>

² <http://www.emule.com>

³ <http://www.napster.com>

⁴ <http://www.icq.com>

⁵ <http://www.bittorrent.com>

2.3 WEB SEMÂNTICA

A Web Semântica [BERNERS-LEE, HENDLER e LASSILA, 2001] é descrita como a extensão da web existente atualmente. Atualmente na web, a maioria das informações é designada para o entendimento humano e muitas delas são compostas somente por texto, sem conexão alguma entre si. O uso de semântica pode adicionar significado a estas informações, permitindo ainda que estas informações sejam processáveis por máquinas. Uma das formas de se adicionar semântica é através do uso de ontologias. As ontologias permitem a especificação de conhecimento, através do uso de regras e relacionamentos entre informações, permitindo assim o uso de semântica na busca por informações [JOUNG e CHUANG, 2009].

Com a ideia da Web Semântica proposta por BERNERS-LEE, HENDLER e LASSILA [2001] o uso de ontologias se popularizaram na área de software, pois elas podem ser empregadas em sistemas relacionadas ao gerenciamento e compartilhamento de conhecimento, inteligência artificial, recuperação e descoberta de informação, educação, agentes, bioinformática e sistemas biológicos [DA SILVA, BRAGA e CAMPOS, 2012], [COSTA et al., 2013], [CLASSE et al., 2013a]. Assim, ontologias consistem em um conjunto de conceitos e regras de inferência, as quais permitem capturar um conhecimento que não esteja implícito em sua modelagem por meio de máquinas de inferências (*reasoners*). Esses mecanismos de inferência fornecem algoritmos que conseguem derivar este novo conhecimento e relacionamentos nas ontologias [FERREIRA FILHO et al., 2012].

A fim de tornar a Web Semântica possível, o consórcio internacional W3C (*World Wide Web Consortium*) trabalha na construção de padrões abertos para o uso de ontologias, incentivando seu uso tanto no meio acadêmico quanto industrial [DA SILVA, BRAGA e CAMPOS, 2012]. De acordo com o W3C a linguagem padrão para a definição de ontologias é o OWL a qual está atualmente em sua versão 2.0 [OWL2.0, 2012].

O OWL foi projetado com o objetivo de suprir as necessidades de uma linguagem ontológica para a web semântica e resolver limitações das linguagens anteriores. Com base na lógica descritiva, essa linguagem

possibilita que mecanismos de inferências explicitem conhecimentos, ou seja, um documento OWL não deve ser considerado somente pelo seu ponto de vista de sintaxe, mas também por sua semântica. Uma ontologia feita com esta linguagem pode incluir descrições de classes, propriedades e instâncias, podendo estes, serem expressos por um único documento ou múltiplos documentos que foram combinados usando métodos definidos pelo OWL. O uso de ontologias também pode ser utilizado para a anotação semântica de serviços web [DA SILVA, BRAGA e CAMPOS, 2012]. Estas anotações semânticas podem ser baseadas em diferentes técnicas e tecnologias como: OWL-S [OWL-S, 2014], WSMO [WSMO, 2014], SA-WSDL [SAWSDL, 2014] e WSDL-S [WSDL-S, 2014], por exemplo.

No contexto de redes ponto a ponto, uma ontologia pode ter diferentes tipos de funcionalidades como: roteamentos entre *peers*; fontes de informação sobre os nós; mapeamentos de recursos disponíveis; anotações semânticas de recursos; descrições de domínios para comunidade semânticas [MODICA, TOMARCHIO e VITA, 2011] entre outros.

Uma rede ponto a ponto pode gerar um grande volume de tráfego entre os seus *peers* e muitas vezes essa quantidade de informações pode ocasionar problemas como lentidão e travamento do sistema. Para que estes problemas fossem solucionados algumas pesquisas [LUA et al. 2005], [MODICA et al. 2011], [NOVAIS, 2012], [CRESPO e GARCIA-MOLINA, 2005] sugerem a divisão da rede ponto a ponto em grupos. Tal técnica diminui o volume de informação transitada entre todos os *peers* da rede, devido à troca de dados ser em grupos.

Neste contexto, DI MODICA et al. [2011] propôs a criação das chamadas comunidades semânticas (ou grupos de interesse), onde os usuários (*peers*) são agrupados e se relacionam de acordo com seus interesses em suas áreas de pesquisa. Essas comunidades semânticas podem ser consideradas como agrupamentos de *peers* de uma rede ponto a ponto que tenham algum interesse em comum. Ou seja, em um grupo de interesse, as informações são compartilhadas apenas com seus integrantes, garantindo assim, que elas não se propaguem a todos os nós da rede e diminuindo, com isso, o volume de informação trafegada [NOVAIS, 2012].

É comum encontrar em abordagens que realizam agrupamentos de *peers* o conceito de Super Nós (*super peers*) [DI MODICA et al. 2011], [NOVAIS, 2012], o qual está associado a um *peer* responsável por gerenciar a comunidade. Em alguns trabalhos, estes super *peers* são associados às propriedades de *peers* de encontro (*rendezvous*) existente na plataforma JXTA [WILSON, 2002], [JXTA, 2013], sendo estes os responsáveis por controlar as ações de um grupo [DI MODICA et al. 2011]. Em outras abordagens, este conceito se aplica ao poder de processamento de um *peer*, geralmente sendo aquele que possui mais recursos para fornecer suporte ao demais [NOVAIS, 2012], [CHIRITA et al., 2006].

Uma comunidade semântica pode ser associada a uma ou mais ontologias de domínio. Essas ontologias são responsáveis por fornecer o conhecimento aos integrantes de um grupo sobre um domínio específico. A partir delas, recursos como documentos, arquivos e serviços podem ser associados à seus termos e pesquisas semânticas de recursos podem ser realizadas, aumentando a descrição dos resultados retornados [CLASSE et al. 2013]. Um *peer* pertencente a uma comunidade semântica pode apresentar diferentes interesses, podendo, pertencer a vários grupos diferentes, ou seja, um *peer* não precisa estar necessariamente somente em uma comunidade, podendo estar dentro de várias delas ao mesmo tempo [NOVAIS, 2012].

Considerando a especificação de uma rede de colaboração científica, estes grupos de interesse permitem aos cientistas buscarem e compartilharem recursos específicos, ou seja, recursos relacionados a sua pesquisa, sem que exista a preocupação em recuperar resultados relativos a outras áreas de atuação em grupos definidos.

2.4 SERVIÇOS WEB SEMÂNTICOS

Os serviços web ou *web services* são sistemas de softwares desenvolvidos para fornecer suporte à interação entre computadores baseando-se em tecnologias Web como HTTP (*Hypertext Transfer Protocol*) e mensagens XML (*eXtensible Markup Language*), podendo ser acessados por diferentes tipos de aplicações, independentemente de suas arquiteturas ou sistemas. Os

outros softwares comunicam com os serviços web por meio de tecnologias como: SOAP, WSDL e UDDI [CAVANAUGH, 2006].

Esses serviços são acessíveis através da internet e em geral são descritos pelo padrão WSDL (*Web Services Description Language*). Essa linguagem descreve informações operacionais sobre o serviço, sua localização na web, mensagens e parâmetros, fornecendo os dados básicos para a sua comunicação [CAVANAUGH, 2006]. Os serviços web semânticos são serviços web que, além de sua descrição normal, apresentam uma descrição semântica de sua estrutura [DA SILVA, BRAGA e CAMPOS, 2012].

Mesmo sendo o responsável por descrever um serviço web, a linguagem WSDL não é capaz de descrever seus recursos semânticamente. Desta maneira, a solução para o desenvolvimento de serviços web semânticos está no mapeamento de conceitos de domínios em uma especificação dos serviços. Entre os elementos do WSDL que podem ser anotados por termos de ontologias estão suas operações, mensagens, parâmetros e resultados. Dentre as abordagens para realizar a especificação de serviços web semânticos usando ontologias é possível destacar OWL-S [OWL-S, 2014], WSMO [WSMO, 2014] e WSDL-S [WSDL-S, 2014], por exemplo [DA SILVA, BRAGA e CAMPOS, 2012].

O OWL-S é uma ontologia de serviços onde é possível expressar as descrições semânticas de um serviço web. Essa ontologia é baseada na necessidade de fornecer três tipos básicos de conhecimento sobre um serviço [OWL-S, 2014] (Figura 2.1):

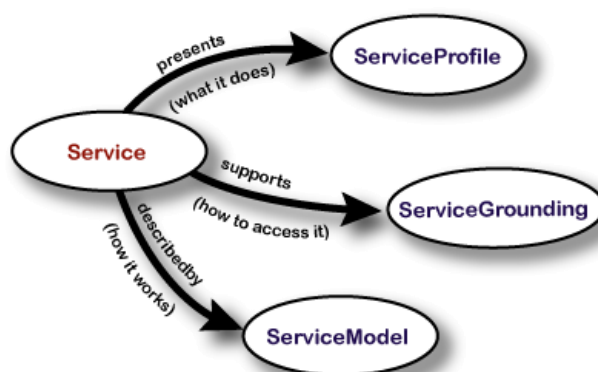


Figura 2.1: Ontologia Service [OWL-S, 2014]

- O que o serviço faz na perspectiva do cliente? Este conhecimento é apresentado pela sub-ontologia *ServiceProfile*. Assim, cada instância da classe *Service* apresenta (*presents*) um *ServiceProfile* associada a ela;
- Como o serviço é usado? Essa informação está presente na sub-ontologia *ServiceModel*. Com isso, uma instância de *Service* é descrita (*describedBy*) por uma *ServiceModel*;
- Como interagir com o serviço? Esse conhecimento é fornecido pela *ServiceGrounding*, a qual fornece as informações necessárias para a execução de um serviço. Assim, uma instância de *Service* suporta (*supports*) uma *ServiceGrounding*.

A classe *Service* representa a principal informação sobre o serviço e a cada operação publicada no *web service*, haverá uma instância *Service*. A Figura 2.1 detalha a comunicação com as classes *ServiceProfile*, *ServiceGrounding* e *ServiceModel*, através das propriedades, respectivamente, *presents*, *supports* e *describedBy*. A classe *ServiceProfile* apresenta informações (parâmetros, operações, URIs e outras) úteis para a descoberta de serviços. E as classes *ServiceModel* e *ServiceGrounding* fornecem informações sobre como um serviço pode ser utilizado, baseando em suas requisições, interfaces e protocolos.

2.5 COMPOSIÇÃO DE SERVIÇOS

Com o crescimento do número de serviços web e aplicações disponíveis na web, aumenta também o número de abordagens envolvendo o suporte a composição e reuso de serviços em larga escala [BOUHINI et al., 2010]. Essa perspectiva de uso computacional em larga escala, é vista pela ciência como possibilidade de significativos avanços em diferentes áreas, através da junção de diferentes tipos de serviços para a realização de uma tarefa comum.

No contexto de *e-Science*, os modelos computacionais gerados pelo processo de composição de serviços permitem a execução de experimentos com menor custo computacional e em um intervalo de tempo menor, pois não

necessita de equipamentos complexos, como os existentes em laboratórios de pesquisa [DA SILVA, BRAGA e CAMPOS, 2012].

Segundo GIL et al. [2007], *workflows* emergem como um forte paradigma para representação e gerenciamento de computações científicas distribuídas complexas, devido a existência de um modelo formal de experimentação. Em contraponto a um experimento científico tradicional, os experimentos que utilizam modelos computacionais como composições de serviços podem ser facilmente reproduzidos, analisados, repetidos, revisados e adaptados.

De acordo com GOBLE e ROURE [2009] *workflows* fornecem: 1) um meio sistemático e automatizado de conduzir análises entre diferentes aplicações e dados; 2) um meio de capturar processos onde os resultados podem ser reproduzidos e os métodos podem ser revisados, validados e adaptados; 3) uma *interface* onde os cientistas podem criar seus experimentos computacionais sem que haja a necessidade deles conhecerem as especificações das tecnologias; 4) uma integração de plataformas, independentemente dos recursos utilizados por cada cientista.

Nesse sentido, alguns esforços são realizados para melhorar a eficácia da geração ou automatização do processo de criação de experimentos, visando as composições. Dentre esses métodos, pode-se destacar a *web semântica* a partir do uso de ontologias para descrever áreas de pesquisa, conectar termos e serviços web [NOVAIS, 2012].

Contudo, SYCARA et al. [2011] aponta que existem três problemas fundamentais para as composições de serviços web: Planejamento, descrevendo como os serviços podem ser integrados e como as funcionalidades se ligam para solucionar um problema; Descoberta, onde exige o desvelamento dos serviços em tarefas requisitadas no planejamento; e Integração, que consiste no gerenciamento entre as interações dos serviços.

Sendo assim, composição de serviços pode ser enunciada como a habilidade de integrar serviços em aplicações de alto nível [BLAKE, TAN e ROSENBERG, 2010], os quais podem ser modelados com ferramentas como Taverna [TAVERNA, 2014], BPEL [ORACLE, 2014], VisTrails [VISTRAILS, 2014], Kepler [KEPLER, 2014] e outras. No contexto deste trabalho, é proposto

o uso de recursos da web semântica conjuntamente com redes ponto a ponto e redes de petri para a especificação de composições de serviços.

2.5.1 REDES DE PETRI

Existem diferentes abordagens para a especificação e modelagem de fluxos de processos, dentre essas, algumas podem ser citadas como a BPMN (*Business Process Modeling Notation*), a BPEL (*Business Process Execution Language*) e as redes de Petri, por exemplo. A abordagem BPMN fornece uma notação compreensível pelo usuário para a modelagem de fluxos de processos, essa abordagem também permite a criação de diagramas compostos por fluxos de objetos, suas conexões, eventos e atividades [WHITE, 2004]. Outra abordagem bastante usada é a BPEL, cuja modelagem de workflows é baseada em XML por meio de padrões de conexão, variáveis, e atividades, entretanto, existem ferramentas que permitem as construções gráficas dos processos [STREULE, 2009].

No contexto desta dissertação, foi utilizada uma terceira abordagem, a chamada redes de Petri. Essa notação permite expressar as composições entre os serviços por meio de conexões entre tarefas de maneira gráfica, o que a faz compreensível para o usuário [PÁDUA et al., 2002]. Este formalismo foi selecionado para este trabalho devido ao seu uso prévio em pesquisas do grupo [NENC, 2014] para a modelagem de fluxos de serviço, podendo ser citados textos como o de Da Silva, Braga e Campos [2012] e Costa et al. [2003].

A teoria inicial das redes de Petri foi apresentada em 1962 por Carl Adam Petri [PETRI, 1962], na qual permite a representação matemática de sistemas, fornecendo informações importantes sobre a estrutura e o comportamento de modelos. Tais redes podem ser utilizadas para modelagem de sistemas e estudos em diferentes áreas de conhecimento como por exemplo economia, administração, indústria, engenharia de software e desenvolvimento de *workflows* [PÁDUA et al., 2002].

De acordo com MURATA [1989] as redes de Petri são constituídas por dois componentes principais denominados transição e lugar, ou posição, sendo

estes interconectados por arcos orientados. Os arcos podem ser únicos ou múltiplos, conectando as posições para transições. A realização de ações no sistema está relacionada a pré-condições, ou seja, existe uma relação entre a posição e a transição que possibilita a execução de uma ação. Assim, após a realização de uma ação, algumas posições terão suas informações alteradas. Graficamente, neste tipo de representação, as posições são representadas por círculos e as transições por barras ou traços, como mostrado na Figura 2.2.

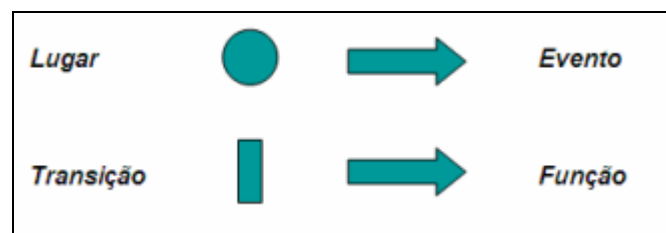


Figura 2.2: Elementos de uma rede de Petri [DA SILVA, BRAGA e CAMPOS, 2012]

No contexto de modelagem de composição de serviços, o uso da representação gráfica fornecida pelas redes de Petri se mostra útil pois permite a visualização do processo como um todo e a comunicação entre cada serviço, detalhando, por exemplo, a representação lógica entre os serviços, os quais podem ser representados por transições e o próprio fluxo de trabalho representado pelas conexões dos arcos [PÁDUA et al., 2002]. HOLT [2000] considera que um modelo de *workflow* apoiado por redes de Petri pode oferecer a fundamentação lógica necessária ao gerenciamento de um processo, devido a flexibilidade de modelagem fornecida por elas, ao passo que se o modelo é considerado como um sistema com operações complexas interligadas, o uso das redes em seu planejamento parece uma ideia natural.

De acordo com VAN DER AALST [1998] existem algumas razões para se utilizar redes de Petri na definição de *workflows*, sendo elas:

- **Semântica Formal:** um processo de *workflow* especificado com termos de redes Petri apresenta uma definição precisa e clara, devido as definições formais presentes nas redes de Petri e suas extensões;

- **Natureza Gráfica:** a rede de Petri é uma linguagem gráfica, e por isso, elas são intuitivas e de fácil aprendizado. Além disso, a natureza gráfica também fornece comunicação com o usuário final;
- **Expressividade:** as redes de Petri suportam todas as primitivas necessárias para a construção de processos de *workflow*. Todas as construções de rotas presentes em sistemas de gerenciamento de *workflows* podem ser modeladas, até mesmo permitindo a modelagem de escolhas;
- **Propriedades:** a base matemática permite que conhecimento seja obtido pelas propriedades das redes de Petri;
- **Análise:** as redes de Petri oferecem muitas técnicas de análises, fato que possibilita uma grande vantagem no uso para a modelagem de *workflows*. Essas técnicas podem fornecer dados, por exemplo, sobre o desempenho de tarefas e de execução;
- **Independência de Fornecedor:** as redes de Petri fornecem moldes para modelagem e análise de processos independente da ferramenta utilizada;

A principal força no uso de redes de Petri na modelagem de *workflows* é resultado entre a combinação da forte fundamentação matemática, representação gráfica facilmente compreensiva, a possibilidade de executar simulações e verificações, diferentes níveis possíveis de formalidade, flexibilidade e integração de aspectos comportamentais [PÁDUA et al., 2002].

De acordo com WfMC [1995 apud DA SILVA, BRAGA e CAMPOS, 2012], um modelo de *workflow* deve ser construído determinando como as tarefas são roteadas seguindo as seguintes construções: sequencial, paralela, condicional e interação. Segundo [VAN DER AALST [1998] o detalhamento desses processos são:

- **Roteamento Sequencial (*SEQUENCE*):** é utilizado para conexão casual entre duas tarefas, por exemplo: sejam A e B duas tarefas, se B é executada após o término da execução de A, então o relacionamento

entre elas é sequencial. É possível ver na Figura 2.3 que a posição c_2 modela o relacionamento entre A e B, ou seja, c_2 apresenta uma pós-condição para a tarefa A e uma pré-condição para a tarefa B;

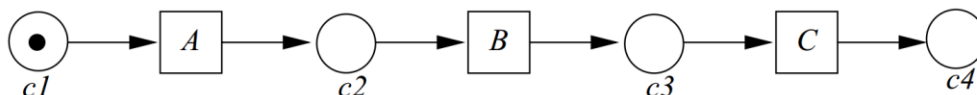


Figura 2.3: Roteamento sequencial [VAN DER AALST, 1998]

- **Roteamento Paralelo (*AND-JOIN* ou *AND-SPLIT*):** este tipo de roteamento é utilizado onde a ordem de execução é menos restrita. Considerando duas tarefas B e C as quais precisam ser executadas não importando a ordem. Para a formalidade das redes de Petri, são inclusos dois blocos: *AND-SPLIT* e *AND-JOIN*. Observando a Figura 2.4, a tarefa A é dividida habilitando tanto B quanto C, enquanto a tarefa D, indica uma sincronização entre B e C, ou seja, A é uma pré-condição para a execução de B e C, e D só está habilitado a ser executado quando B e C forem executadas. Como resultado, as tarefas B e C são executadas em paralelo;

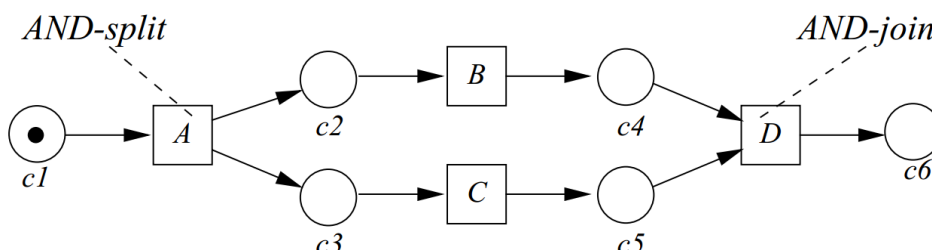


Figura 2.4: Roteamento paralelo [VAN DER AALST, 1998]

- **Roteamento Condicional (*OR-SPLIT* e *OR-JOIN*):** este roteamento é utilizado quando as possibilidades podem variar entre os casos. Nas redes de Petri para se modelar um roteamento condicional é necessário utilizar dois blocos de construção: *OR-SPLIT* e *OR-JOIN*. Olhando a Figura 2. 5 é possível ver que após a execução da tarefa A, a tarefa B ou a tarefa C é executada, ou seja, uma é executada e a outra não dependendo de uma condição qualquer. Nesse sentido, a execução dessas tarefas é seguida pela execução de D;

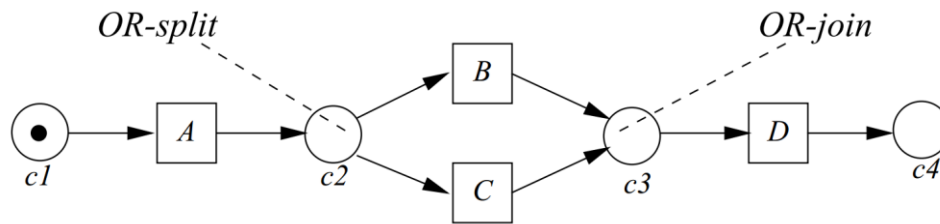


Figura 2. 5: Roteamento condicional [VAN DER AALST, 1998]

- **Roteamento Iterativo:** esse tipo de roteamento é utilizado quando existe a necessidade de utilização de uma tarefa várias vezes. Veja a Figura 2.6, ela mostra que após a execução da tarefa B, pode ser que exista a necessidade de utilizar o mesmo serviço B outra vez.

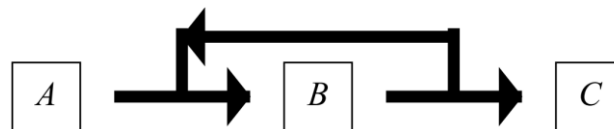


Figura 2.6: Roteamento iterativo [VAN DER AALST, 1998]

2.6 BIOINFORMÁTICA: SEQUENCIAMENTO E ALINHAMENTO GENÉTICO

Considerando o contexto de *e-Science*, este trabalho tem como um dos seus focos o uso da infraestrutura e-ScienceNet em domínios científicos. Desta maneira, para que haja compatibilidade entre as provas de conceito apresentadas no capítulo 4 e o entendimento do leitor, é detalhado a seguir o subdomínio de sequenciamento e alinhamento genético, no intuito de facilitar o entendimento acerca dos termos utilizados.

A partir da década de 90, com o aparecimento de sequenciadores automáticos de DNA, ocorreu um significativo aumento na quantidade de seqüências a serem armazenadas, exigindo computadores cada vez mais eficientes. Com este aumento súbito também surge a necessidade de análise desses dados, exigindo plataformas computacionais eficazes para processar estes recursos. Nesse contexto temos o surgimento da bioinformática, uma nova ciência que envolve a união de diversas áreas do conhecimento como engenharia de software, matemática, estatística e biologia molecular [PROSDOCIMI et al., 2002].

De acordo com a Biologia, DNA (Ácido Desoxirribonucleico) e RNA (Ácido Ribonucleico) são cadeias de polímeros denominadas ácidos nucleicos compostos por substâncias químicas similares e individuais chamadas nucleotídeos. O DNA é composto por ligações entre nucleotídeos do tipo A (Adenina), T (Timina), C (Citosina) e G (Guanina), enquanto o RNA não apresenta a timina e em seu lugar é encontrada o nucleotídeo U (Uracila) [KLIPP et al., 2008]. Nesse sentido, a pesquisa em Bioinformática pode compreender desde abstrações de um sistema biológico em modelos físicos ou matemáticos, até a implementação de novos algoritmos ou plataformas de processamentos para análise de dados [DA SILVA, BRAGA e CAMPOS, 2012].

Dado o volume de dados gerados, esses dados biológicos se encontram espalhados em diversas bases de dados que podem ser acessadas por meio de serviços web ou de *interfaces* de acesso próprias. Esses bancos de dados são resultados de investimentos contínuos nessa área e uma das grandes motivações para isso é o sucesso de projetos genoma no mundo. Dentre essas bases de dados, podem ser citadas como mais utilizadas [PROSDOCIMI et al., 2002]:

- **GenBank**⁶: é o banco de dados americano para sequências de DNA e proteínas, fazendo parte do NCBI (*National Center for Biotechnology Information*);
- **EBI**⁷ (*European Bioinformatics Institute*): é o banco de dados europeu de sequências de DNA;
- **DDBJ**⁸ (*DNA Data Bank of Japan*): é o banco de dados japonês para sequências de DNA;
- **PDB**⁹ (*Protein Data Bank*): base de dados onde são armazenadas estruturas tridimensionais de proteínas resolvidas;
- **TIGBDatabases**¹⁰: banco de dados com informações genômicas de vários organismos;

⁶ <http://www.ncbi.nlm.nih.gov/>

⁷ <http://www.ebi.ac.uk/>

⁸ <http://www.ddbj.nig.ac.jp/>

⁹ <http://www.rcsb.org/pdb>

- **PIR**¹¹ (*Protein Information Resource*): banco de dados de proteínas anotadas;
- **SWISS-PROT**¹²: armazena as sequências de proteínas e suas características moleculares anotadas por especialistas;
- **KEGG**¹³ (*Kyoto Encyclopedia of Genes and Genomes*): banco de dados com informações genômicas de diversos organismos e informações sobre suas vias metabólicas.

Ao isolar novas sequências em laboratórios, os pesquisadores desejam saber todas as informações possíveis sobre ela. Assim, uma das maneiras mais simples é verificar se outros cientistas já estudaram suas propriedades [DA SILVA, BRAGA e CAMPOS, 2012]. De acordo com PROSDOCIMI et al., [2002] existem algoritmos de alinhamento, os quais são ótimas maneiras de realizar comparações de similaridades entre duas ou mais sequências.

O estudo do material genético é possível através do processo de sequenciamento, que determina a sequência de nucleotídeos deste material. Existem diversas plataformas de análise de sequências genéticas sendo que uma das mais utilizadas é o método Sanger [SANGER, NICKLEN e COULSON, 1977]. Ainda existem plataformas de sequenciamento mais modernas chamadas NGS (*Next Generation Sequencing*) que, em alguns casos, podem trazer resultados bem melhores que a plataforma anterior [CARVALHO e SILVA, 2010], podendo ser destacadas: Solexa, 454 FLX, tSMS e SOLiDSystem.

Baseando-se nos conceitos de sequenciamento/alinhamento genético, COSTA et al. [2013], em seu trabalho desenvolveu uma sequência de tarefas que estão envolvidas no processo de sequenciamento e alinhamento de DNA. De acordo com PROSDOCIMI et al. [2002] as atividades envolvidas neste processo são:

¹⁰ <http://www.tigr.org/tdb/>

¹¹ <http://www-nbrf.georgetown.edu/>

¹² <http://www.expasy.ch/spro/>

¹³ <http://www.genome.ad.jp/kegg/>

- **Base Calling:** essa atividade está relacionada com o processo de leitura genética através de um aparelho denominado sequenciador genético. Tal aparelho atribui valores e posições a cada sequência de nucleotídeos encontrada. Geralmente, cada um desses sequenciadores realiza esse processamento de maneira diferente. Por exemplo, sequenciadores que utilizam a plataforma Sanger normalmente utilizam o programa *PHRED*. Já os que utilizam alguma plataforma *NGS* são sensíveis às tecnologias de sequenciamento e precisam de muitos recursos computacionais. Estes softwares geralmente reconhecem a sequência de nucleotídeos através de um arquivo cromatograma gerando ao final, arquivos *FASTA* ou *PHD* contendo notas com pontuações de qualidade do sequenciamento;
- **Vector Masking:** com a conclusão da *base calling*, com o arquivo contendo a pontuação é possível realizar uma varredura na sequência buscando regiões que devem ser substituídas ou mascaradas. Essas regiões são denominadas regiões contaminantes, pois não apresentam corretamente o DNA a ser analisado. Como exemplo de programa para mascaramento de vetores é possível citar o *Cross Match*;
- **Sequence Grouping:** após a remoção das regiões contaminadas toda a informação obtida passa para um programa como o *PHRAP*, ou *CAP3*, os quais agrupam os fragmentos de DNA em sequências maiores denominadas *contigs*. Nesse sentido, espera-se ao final do processamento obter o que se chama de contíguo genômico.
- **View and Editing:** essa é uma etapa opcional, pois se trata de um processo manual em que o pesquisador verifica a montagem e o agrupamento de sequências. Entre os programas que possibilitam estas operações, encontram-se o *Phrapview* e o *Consed*;
- **Aligning:** essa tarefa consiste basicamente em comparar sequências, colocando-as uma sobre a outra a fim de especificar as partes comuns entre elas. Existem dois tipos de algoritmos de alinhamento os quais dependem do tipo que deseja ser realizado. São eles: alinhamento local

(*local aligning*), o programa realiza o alinhamento apenas de fragmentos das sequências; e alinhamento global (*global aligning*), onde é realizado o alinhamento completo das sequências envolvidas. Além disso, o grau de alinhamento entre as sequências pode ser feito para duas ou mais sequências o que acarreta, também na divisão entre: alinhamento duplo (*pairwise_aligning*), o que corresponde a alinhamentos entre pares de sequências; e alinhamento múltiplo (*multiple_aligning*), o que corresponde ao alinhamento de múltiplas sequências ao mesmo tempo. Programas como *Blast*, *EMBOSS* e *Clustal* podem ser citados como exemplos dessa tarefa;

- **Assembly:** essa tarefa ocorre quando existem sequências que sirvam como referências após o processo de alinhamento, ou seja, quando não existem sequências produzidas de um genoma da mesma espécie da amostra ou de alguma espécie relacionada;

A partir destes conceitos, COSTA et al. [2013] desenvolveu a ontologia *SequenceAligningOntology*¹⁴, para o domínio de sequenciamento e alinhamento genético contendo os termos relacionados a ele. Essa ontologia é capaz de inferir equivalência entre os programas e as atividades discutidas acima e será utilizada no contexto deste trabalho.

2.7 TRABALHOS RELACIONADOS

Considerando o tema de redes ponto a ponto semânticas, *e-Science* e composição de serviços alguns trabalhos foram apresentados ao longo dos anos, explorando a capacidade tecnológica fornecida por essas abordagens [STUCKENSCHMIDT et al., 2006], e com isso, surgiram trabalhos importantes como Edutella [NEJDL et al., 2002], KEx [BONIFACIO et al. 2002], XAROP [LLADÓ E SALAMANCA, 2006], CABO-P2P [FURTADO E FARIAS, 2005] dentre outros.

¹⁴ http://plscience.superdignus.com/e_ScienceNet/SequenceAligningOntology.owl

Um dos primeiros trabalhos de destaque na área foi a ferramenta Bibster [HAASE et al., 2004]. Essa ferramenta consiste em um sistema web semântico baseado na tecnologia de redes ponto a ponto, tendo como objetivo principal o compartilhamento de informação bibliográfica contidas em arquivos *BibTeX* [BIBTEX, 2014]. Estes arquivos contêm diversas informações de referencial bibliográfico como autores, publicação, ano, título e etc. A partir de informações bibliográficas, o sistema Bibster realiza o mapeamento delas em ontologias as quais fornecem informações que podem ser pesquisadas pelos *peers* conectados à rede.

Outro trabalho que merece destaque, é o desenvolvido por BIANCHINI et al. [2010], que cria uma abordagem usando redes ponto a ponto semânticas baseadas em modelos dinâmicos de *peers* para fornecer recursos com conteúdos similares. Nesse enfoque, também merece destaque o uso de uma estrutura local de conhecimento, a qual inclui registros de publicações de serviços web, além de ontologias baseadas em conteúdos distintos.

A arquitetura CABO-P2P de [FURTADO E FARIAS, 2005] propõe o uso de uma rede ponto a ponto semântica para o compartilhamento de arquivos, onde os recursos são mapeados em ontologias, possibilitando a melhoria da descrição dos mesmos. Esse mapeamento abrange informações como nome, data e formato do arquivo, que podem ser pesquisadas através de linguagens de consultas ontológicas.

A abordagem de redes ponto a ponto semânticas na descoberta de serviços são mostradas nos trabalhos de GHARZOULI E BOUFAIDA [2011] e MANDLER et al. [2013]. Nestes trabalhos, os pesquisados criam composições de *workflows* realizando buscas em serviços web semânticos, além disso, essas composições se baseiam em três tipos de operações básicas: *AND*, *OR* e *SEQUENCE*. Ademais, GHARZOULI E BOUFAIDA [2011] e MANDLER et al. [2013] destacam o uso de algoritmos para a descoberta, recuperação e cruzamento de informações de serviços web, a fim de realizar o processo de composição do *workflow*. O trabalho de MANDLER et al. [2013] propõe a utilização de DHT (*Distributed Hash Table*) no processo de descoberta de serviços, enquanto GHARZOULI E BOUFAIDA [2011] fazem o uso de um

algoritmo de comparação de parâmetros de entrada e saída que é disseminado aos *peers* existentes na rede.

DI MODICA et al. [2011] propôs uma arquitetura que combina as tecnologias de rede ponto a ponto com web semântica para realizar o agrupamento dos *peers* de acordo com suas áreas de interesse, permitindo que o tráfego da rede diminua, pois a comunicação seria restrita aos grupos e não mais a toda a rede. Este autor explica que por maior que seja a comunidade, sua comunicação não se compara a realizada entre todos os *peers* de uma rede, sendo o pior caso, quanto há somente a existência de apenas um grupo.

O trabalho de WILKINSON [2013], se baseia em um método de composição de experimentos através do uso de uma ontologia que serve de modelo para o desenvolvimento do *workflow*, realizando a associação de serviços web existente com o modelo criado.

Além dos trabalhos assinalados acima, é importante destacar os trabalhos de DA SILVA, BRAGA e CAMPOS [2012] e COSTA et al. [2013], que influenciaram diretamente neste trabalho. O trabalho de DA SILVA, BRAGA e CAMPOS [2012] propôs um *framework* para a composição de serviços web semântica, por meio da inclusão de serviços web anotados semanticamente e a composições utilizando BPEL [ORACLE, 2014]. Já COSTA et al. [2013], desenvolveu uma arquitetura de Linhas de Produto de Software onde é possível desenvolver um *workflow* através de conceitos e regras existentes em uma ontologia de domínio. Em suma, estes dois trabalhos tem como objetivo apoiar experimentos em bioinformática, mais precisamente, no sequenciamento e alinhamento de genes.

2.7.1 ANÁLISE DOS TRABALHOS RELACIONADOS

A partir dos trabalhos relacionados, algumas ideias e características foram aproveitadas e adaptadas contribuindo no desenvolvimento das principais funcionalidades desenvolvidas na e-ScienceNet, como, por exemplo, o uso de ontologias para o mapeamento de recursos e anotação semântica de serviços [HAASE et al., 2004], [FURTADO E FARIAS, 2005], pesquisa e descoberta de

serviços web semânticos [MANDLER et al., 2013], [GHARZOULI E BOUFAIDA, 2011], [BIANCHINI et al., 2010], composição de serviços científicos [DA SILVA, BRAGA e CAMPOS, 2012], [WILKINSON, 2013], [MANDLER, 2013] e utilização de grupos semânticos [DI MODICA et al., 2011]. Entretanto, a abordagem apresentada neste trabalho mostra características distintas conforme serão detalhadas a seguir.

A ferramenta Bibster [HAASE et al., 2004] assim como a e-ScienceNet é formada por um modelo arquitetural baseado em módulos, os quais usam ontologias e motores de inferência como características web semânticas das aplicações. Como já dito, essa ferramenta se baseia no compartilhamento de informações bibliográficas por meio de arquivo *BibTeX* [BIBTEX, 2014]. Essas informações são mapeadas por meio de ontologias desenvolvidas em RDF, as quais permitem as consultas de informações por meio de *queries* criadas com a linguagem de consulta SeRQL [BROEKSTRA E KAMPMAN, 2003]. Pode-se dizer que a e-ScienceNet também faz o compartilhamento de seus recursos através de consultas em ontologias, porém, nesta abordagem são usadas ontologias OWL, por se tratar de uma tecnologia mais moderna e que oferece mais funcionalidades que a RDF, e sua consulta é feita através de *queries* na linguagem SPARQL [DUCHAME, 2011] por fornecer a padronização recomendada pelo W3C para recuperação de dados web semânticos. Outra diferença é que a e-ScienceNet compartilha qualquer tipo de documento ou serviço web, não somente informações mapeadas por arquivos *BibTeX*.

A abordagem CABO-P2P de FURTADO e FARIAS [2005] propõe a utilização de um modelo arquitetural baseado em rede ponto a ponto semântica com o propósito de facilitar o compartilhamento de documentos, utilizando para tal fim, ontologias. Entretanto, este trabalho é apresentado somente como uma arquitetura, enquanto a e-ScienceNet apresenta também um protótipo implementado. Os autores do trabalho CABO-P2P citam futuras melhorias, e que seriam utilizadas consultas baseada na indexação de documentos com termos de possíveis ontologias, ou seja, existiria em cada *peer* um repositório contendo um índice de documentos onde haveria informações sobre este mesmo e uma relação com o termo de alguma ontologia. Eles ainda mencionam uma possível implementação de consultas às informações por

meio de SeRQL. A e-ScienceNet considera o mapeamento de seus recursos em ontologias existentes em cada um de seus nós, onde suas informações são obtidas através de uma *interface* de inserção de documentos. Essas informações são mapeadas na ontologia, dispensando o uso de um arquivo de indexação. Utilizando consultas em SPARQL nessas ontologias locais, é possível realizar o compartilhamento dos documentos e também obter informações inferidas por mecanismos aptos à isso.

DI MODICA [2011] apresenta em seu trabalho o conceito de que *peers* com conteúdos semelhantes sejam agrupados em comunidades semanticamente relacionadas. Em sua proposta, ele tira proveito da característica de criação de grupos existentes na tecnologia JXTA para fazer a separação destes *peers*. A este conceito DI MODICA [2011] deu o nome de comunidades semânticas (ou grupos de interesse), onde os *peers* são classificados em grupos relacionados com seus interesses e áreas de atuação. DI MODICA [2011] utilizando-se da funcionalidade de pontos de encontro (*rendezvous*, pontos responsáveis por grupos de *peers*) da tecnologia JXTA, realiza o conceito de que estes *peers* seriam, automaticamente, os responsáveis por administrarem os recursos de uma comunidade semântica. Neste trabalho também é utilizada a tecnologia JXTA para implementar o conceito de grupos de interesse, porém, qualquer *peer* existente na e-ScienceNet é capaz de gerenciar uma comunidade. Ou seja, aqui a gerência de uma comunidade primeiramente é associada a sua construção, quando um *peer* cria um grupo este passa a ser o seu administrador (Super Nó). Caso este *peer* saia da comunidade, a gerência é passada a qualquer um nó que pertença a este grupo. Outra diferença da e-ScienceNet é que, normalmente, uma comunidade semântica é associada a uma ou mais ontologias de domínio, as quais caracterizam as regras e comportamentos do grupo.

Em termos de redes ponto a ponto semântica e composição de *workflows*, BIANCHINI et al. [2011] abordam o uso de descrição semântica de serviços web a partir do processamento de arquivos WSDL. Essa associação segue um padrão chamado taxonomia de categorização de serviços, o que compreende informações de UDDI (*Universal Description, Discovery and Integration*), parâmetros de entrada e saída do serviço e uma conexão

semântica feita por ontologias de domínio. Na e-ScienceNet os serviços web são processados através da informação contida em um arquivo WSDL e essas informações são anotadas por termos existentes em ontologias de domínio disponíveis em cada grupo semântico. Entretanto, nossa abordagem faz o uso do padrão OWL-S para anotação semântica do serviço web. Ao final do processamento, são armazenados em repositórios tanto a ontologia OWL-S gerada e o arquivo WSDL do serviço. A vantagem deste armazenamento é a de que caso haja necessidade de consultar informações sobre algum serviço novamente, não é preciso que o mesmo seja acessado através da internet, apenas consultar suas informações localmente.

GHARZOULI E BOUFAIDA [2011] também abordam a composição de serviços através de redes ponto a ponto. Em seu trabalho eles realizam a descoberta dos serviços cruzando informações obtidas em arquivos OWL-S e fazendo o retorno dos serviços web compatíveis. Em sua abordagem existe o uso de tabelas distribuídas entre os nós da rede, as quais armazenam informações sobre o *peer* responsável por um determinado serviço e suas possíveis composições. No contexto da e-ScienceNet, a descoberta de serviços existentes nos *peers* também é realizada através de cruzamento das informações existentes em OWL-S, contudo, cada ontologia de serviço é anotada com termos de ontologias de domínio, o que faz com que somente os serviços web relacionados com uma determinada tarefa sejam retornados e posteriormente, é realizado o cruzamento de informações sobre a compatibilidade dos parâmetros. As composições geradas na e-ScienceNet também são armazenadas, porém, são disponibilizadas como recursos da e-ScienceNet, podendo ser reutilizadas em outros *peers* da rede, sem que haja o uso das tabelas distribuídas.

O trabalho de WILKINSON [2013] não se baseia em redes ponto a ponto, porém apresenta um conceito interessante para a composição de serviços. O referido autor se baseia na construção de uma ontologia biológica, a qual serve de modelo para a pesquisa de serviços, estas associadas à ontologia. Essa busca de conceitos é baseada em *queries* SPARQL. Na e-ScienceNet existe o conceito de *workflow* abstrato, que é um modelo de composição construído com base em regras e conceitos existentes em

ontologias de domínios. Diferente do trabalho de WILKINSON [2013], a e-ScienceNet realiza a busca dos serviços web baseada neste modelo abstrato, não desenvolvendo uma ontologia própria para a composição.

Baseando-se na área de Bioinformática DA SILVA, BRAGA e CAMPOS [2012] propôs um *framework* arquitetural para a composição de serviços web utilizando ontologias de domínio. Em seu trabalho foi desenvolvido mecanismos de anotação semântica de serviços web por meio de caixas de seleção, as quais continham termos das ontologias de domínio. O processo de composição de DA SILVA, BRAGA e CAMPOS [2012] também se baseava na descoberta de serviços por meio da utilização do cruzamento de parâmetros existentes em ontologias de serviços OWL-S. Os serviços retornados pelas pesquisas eram interconectados por caixas de seleção contendo dados baseados em conexões lógicas descritas em redes de Petri. As composições geradas, então, eram tratadas pela tecnologia BPEL. Comparada com o trabalho de DA SILVA, BRAGA e CAMPOS [2012], a e-ScienceNet apresenta *interfaces* gráficas onde é possível verificar em tempo de execução a criação dos modelos de composição, bem como realizar anotações semânticas de serviços web. Estes modelos também são baseados em redes de Petri. A e-ScienceNet não faz o uso de BPEL, tendo um próprio padrão para o armazenamento e execução de composições.

Considerando ainda os conceitos de Bioinformática, COSTA et al. [2013] propôs uma abordagem para a criação de linhas de produtos de software a partir da utilização e adaptação da ontologia de domínio *myGrid* [MYGRID, 2014], em uma nova ontologia chamada *SequenceAligningOntology*. Em seu trabalho é possível verificar a possibilidade do desenvolvimento de um *workflow* para sequenciamento e alinhamento genético baseado no uso de regras existentes na ontologia de domínio. Porém, o trabalho de COSTA et al. [2013] não se baseia no uso de redes ponto a ponto e seu foco principal não é na execução do modelo de *workflow* criado. A e-ScienceNet permite que o *workflow* desenvolvido pelo usuário seja executado, gerando os resultados do experimento.

Para melhor compreensão, a Tabela 1 apresenta a visão comparada entre os diversos trabalhos relacionados discutidos nesta seção.

Tabela 1: Comparação de Trabalhos Relacionados

	Web Semântica	Ontologia	Consulta	Serv. Web Semânticos	Anotação Semântica	Mapeamento	P2P	Composição de Serviços
CLASSE et al. [2014]	Sim	OWL	SPARQL	Sim	OWL-SE	Documentos e Serviços	Sim	Sim
HAASE et al. [2004]	Sim	RDF	SeRQL	Não	-	BibTeX	Sim	Não
FURTADO e FARIAS [2005]	Sim	RDF	SeRQL	Não	-	Documentos	Sim	Não
DI MODICA [2011]	Sim	OWL	-	Não	-	-	Sim	Não
BIANCHINI et al. [2011]	Sim	OWL	Taxonomia De Serviços	Sim	OWL-S	Serviços	Sim	Sim
GCHARZOULI E BOUFAIDA [2011]	Sim	OWL	-	Sim	OWL-S	Serviços	Sim	Sim
WILKINSON [2013]	Sim	OWL	SPARQL	Sim	OWL-S	Serviços	Não	Sim
DA SILVA, BRAGA e CAMPOS [2012]	Sim	OWL	SPARQL	Sim	OWL-S	Não	Não	Sim
COSTA et al. [2013]	Sim	OWL	SPARQL	Não	-	Não	Não	Sim

2.8 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou as principais tecnologias relacionadas ao contexto deste trabalho como a criação de uma rede ponto a ponto semântica, que oferece suporte ao desenvolvimento de *workflows* por parte de cientistas, além de apresentar uma visão geral sobre serviços web semântico e o domínio de bioinformática que envolve o seqüenciamento e alinhamento de sequências genéticas.

Verifica-se, portanto, que a combinação das tecnologias web semântica com a abordagem proposta pela arquitetura de rede ponto a ponto, pode trazer benefícios. Em uma rede ponto a ponto semântica, o compartilhamento de recursos pode ser realizado com base em inferências realizadas em ontologias muitas vezes através de linguagens de consulta como SPARQL [DUCHAME, 2011], SeRQL ou outra linguagem com essa finalidade [FURTADO e FARIAS, 2005].

Assim, considera-se que com o uso de tecnologias como ontologias, serviços web semânticos e redes ponto a ponto, é possível especificar uma

infraestrutura de apoio a pesquisa científica. Desta forma, neste trabalho é abordada a junção dessas tecnologias a fim de criar uma rede ponto a ponto semântica para o compartilhamento de informações relacionadas a experimentos científicos.

Foram apresentadas ainda, as principais pesquisas relacionadas à abordagem e-ScienceNet, detalhando cada uma das pesquisas, ressaltando as diferenças em relação a e-ScienceNet.

No próximo capítulo, serão abordadas as principais características arquiteturais deste trabalho, junto com seus conceitos e funcionalidades.

3 E-SCIENCENET

Neste capítulo será descrita a abordagem e-ScienceNet, sendo detalhadas sua arquitetura, modelos utilizados em sua construção, tecnologias utilizadas e sua metodologia de utilização. Desta forma, aqui serão destacados pontos importantes sobre o conceito da e-ScienceNet como: gerentes, ontologias, buscas, uso de serviços web na execução dos serviços, interesses de pesquisa, entre outros. Nesse sentido, este capítulo se organiza da seguinte maneira: seção 3.1, introdução; seção 3.2, tecnologias utilizadas; seção 3.3, arquitetura da e-ScienceNet e seus gerentes; e finalmente na seção 3.4 serão apresentadas as considerações finais para este capítulo.

3.1 INTRODUÇÃO

No contexto do domínio científico em geral, uma das atividades importantes é o desenvolvimento de experimentos, que atualmente envolvem a troca de resultados através da colaboração entre pesquisadores. Na maioria das vezes, pesquisadores de diferentes áreas precisam utilizar recursos e aplicações computacionais reutilizando resultados já existentes e alterando partes do experimento para atender a sua necessidade. Porém, nem sempre é trivial encontrar tais recursos ou modificá-los à sua demanda, devido à falta de compartilhamento de tais recursos ou o conhecimento pouco aprofundado de técnicas de computação. Muitas vezes o desenvolvimento de experimentos exige operações complexas e a combinação de vários serviços, que, em geral, são executados de maneira distribuída. Assim, a criação de fluxos de trabalho, envolvendo diversas aplicações científicas, se torna ainda mais difícil, o que muitas vezes faz com que o cientista desista de compor aplicações científicas e não realize o experimento.

De tal modo, neste capítulo é apresentada a arquitetura e-ScienceNet [CLASSE et al., 2013b], que utiliza uma abordagem distribuída através do uso de conceitos de redes ponto a ponto e web semântica a fim de criar um ponto de acesso a dados e aplicações científicas distribuídas e heterogêneas para

fornecer suporte a pesquisadores dispersos geograficamente no processo de obtenção, criação, execução e compartilhamento de experimentos. As buscas por recursos nos *peers* (nós ou pontos em uma rede ponto a ponto) da e-ScienceNet baseiam-se no uso de ontologias. Além das buscas, tais ontologias facilitam no processo de criação de composições científicas através de inferências de novos conteúdos que ajudam o cientista neste processo.

Neste contexto, uma das principais funcionalidades dessa arquitetura é a criação e execução de *workflows* por meio do uso de ontologias de domínio para a montagem do fluxo de serviços e busca de serviços web.

3.2 A ARQUITETURA DA E-SCIENCENET

Como dito na seção 3.1, a e-ScienceNet é uma rede ponto a ponto semântica com o intuito de fornecer suporte a *e-Science* através do uso de uma arquitetura distribuída para a utilização e criação de recursos dispersos geograficamente, permitindo que vários cientistas compartilhem seus trabalhos. Isso faz com que o pesquisador possa gerenciar, analisar, criar, discutir e compartilhar experimentos de forma mais eficiente.

É válido assinalar, que a arquitetura da e-ScienceNet é composta por gerentes de funcionalidades específicas, sendo eles: Gerente de Ferramentas, Gerente de Semântica, Gerente de Recursos, Gerente de Dados, Gerente de Serviços, Gerente de Interesse, Gerente de Pesquisa e Gerente de Composição.

No desenvolvimento da rede ponto a ponto foram analisadas tecnologias como *Sockets* [NUMBERG, 2003] para o seu desenvolvimento. Porém devido à necessidade de adequação do tempo para andamento do trabalho, optamos pelo uso do JXTA [WILSON, 2002], o qual já implementa em suas camadas os protocolos de comunicação necessários para que a rede P2P funcione, além de mecanismos de segurança como autenticação e encriptação.

A Figura 3.1 mostra a arquitetura da e-ScienceNet com seus principais componentes (gerentes). A arquitetura P2P da e-ScienceNet é baseada no *framework* JXTA [WILSON, 2002].

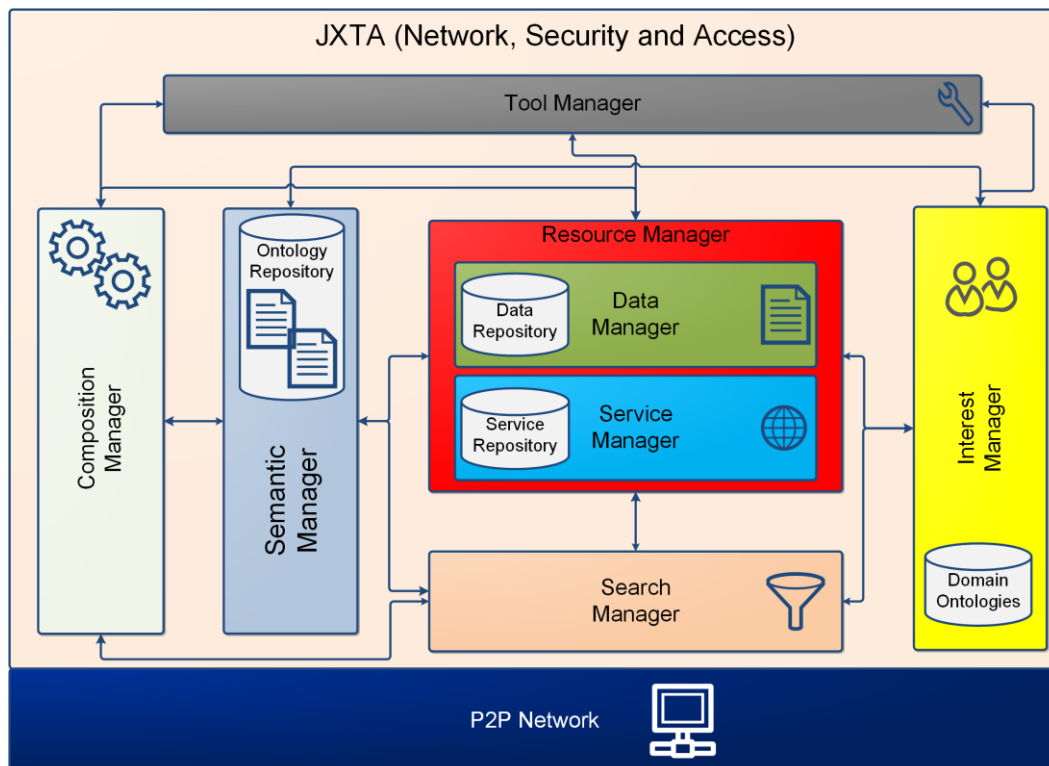


Figura 3.1: Arquitetura da e-ScienceNet

O foco principal da e-ScienceNet é a busca semântica de recursos entre os *peers*, os quais são divididos em grupos de interesses (grupos semânticos (seção 3.2.3.1)). Estas comunidades semânticas são criadas baseando-se no interesse comum de pesquisa em áreas distintas. Dentro de cada uma dessas comunidades semânticas, os cientistas podem realizar buscas a fim de obter recursos que ajudem em sua pesquisa. Neste contexto, documentos e serviços são retornados como recursos utilizáveis. Estes recursos por sua vez são pesquisados considerando termos existentes em ontologias de domínio, utilizando motores de inferência para aumentar o conhecimento utilizado nestas buscas [CLASSE et al., 2013b].

A Figura 3.2 mostra um esboço da arquitetura funcional da e-ScienceNet, que é detalhada abaixo:

- **Descoberta e Atualização de Grupos Semânticos:** quando um *peer* se conecta ao sistema pela primeira vez, é necessário que o mesmo seja incluído em alguma comunidade semântica. Para isso existe a necessidade de procurar na rede, caso existam, os grupos disponíveis. O responsável por este processo é o Gerente de Interesse, o qual

pesquisa as comunidades disponíveis e informa aos novos *peers* a sua existência. Depois de selecionado o grupo, o novo *peer* passa a ser considerado membro da comunidade, o que permite a comunicação e compartilhamento de informações com os demais *peers*. Tal processo de descoberta e conexão será explicado mais detalhadamente na seção 3.2.3;

- **Registro de Ontologias de Domínio:** as ontologias de domínio são responsáveis por descrever as comunidades semânticas. Essas ontologias englobam termos e restrições que são úteis para a pesquisa semântica de informação e composições de serviços. Portanto, para que um *peer* consiga realizar buscas por recursos dentro de um grupo é necessária a seleção ou registro de ontologias de domínio. As ontologias de domínio, que serão descritas com mais detalhes na seção 3.2.3.2, são compartilhadas pelos “super nós”¹⁵ da e-ScienceNet e podem ser utilizadas em qualquer um dos *peers* conectados na comunidade;
- **Inclusão de Serviços:** a fim de compartilhar serviços entre os *peers* em uma comunidade semântica, os mesmos devem estar disponíveis na e-ScienceNet. Este trabalho é feito através do Gerente de Serviços, o qual é responsável pela criação de serviços anotados semanticamente em ontologias OWL-S. Essas ontologias são persistidas no Repositório de Ontologias do Gerente de Semântica e mapeadas na *PeerOntology*¹⁶ para futuras buscas. Esse processo será detalhado na seção 3.2.6;
- **Pesquisa de Serviços:** a pesquisa de serviços é realizada pelo Gerente de Pesquisa em conjunto com o Gerente de Semântica consistindo, primeiramente, na busca de termos inferidos através de motores de inferência em ontologias de domínio. E por último, buscando informações nas ontologias de serviços (arquivos OWL-S) através de consultas SPARQL para que sejam retornados serviços compatíveis. Este processo será delineado na seção 3.2.7;

¹⁵ Este conceito é detalhado na seção 2.3.

¹⁶ http://plscience.superdignus.com/e_ScienceNet//PeerOntology.owl

- Composição de Serviços:** o Gerente de Composição utiliza o serviço de busca a fim de desenvolver composição de aplicações científicas e executar um experimento. A primeira etapa deste processo é gerar um modelo de composição de aplicações científicas, baseando-se nas tarefas e regras descritas pela ontologia de domínio do grupo e depois, buscar serviços que sejam compatíveis um com o outro através de comparações semânticas e sintáticas entre os seus parâmetros de entrada e saída. Na seção 3.2.8, serão descritos os processos utilizados para a criação de tais composições.

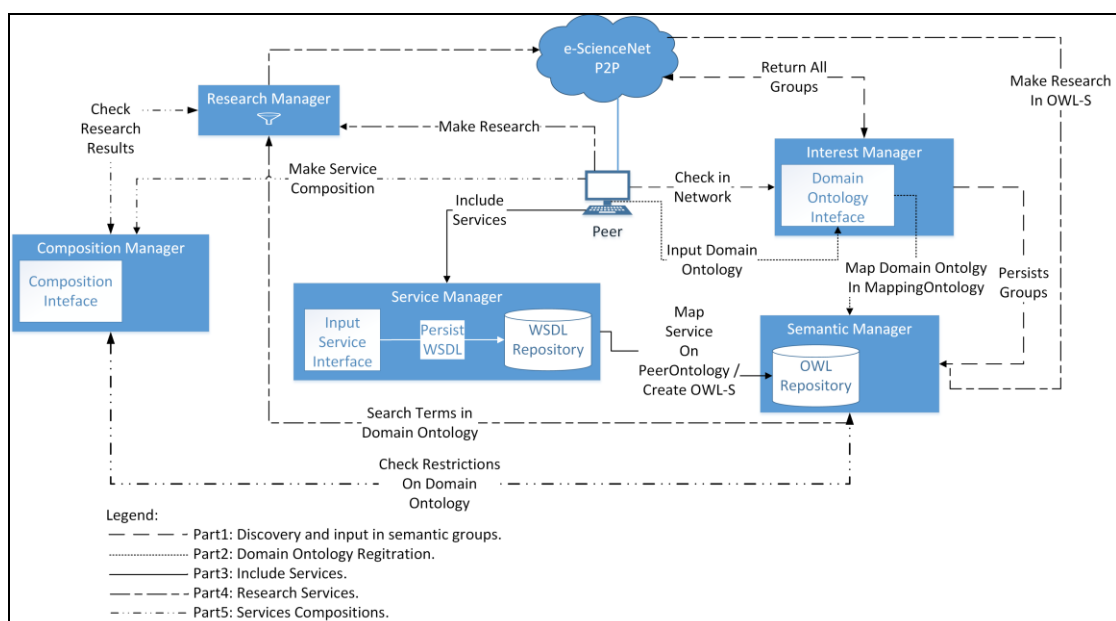


Figura 3.2: Arquitetura Funcional da e-ScienceNet

Nas seções abaixo são detalhados os gerentes da e-ScienceNet e suas principais funcionalidades.

3.2.1 GERENTE DE FERRAMENTAS (TOOL MANAGER)

O Gerente de Ferramentas é responsável pelas ferramentas administrativas da e-ScienceNet como conexão entre os *peers*, configuração da plataforma JXTA, informações sobre conexões e conteúdo geral da e-ScienceNet (Figura 3.3).

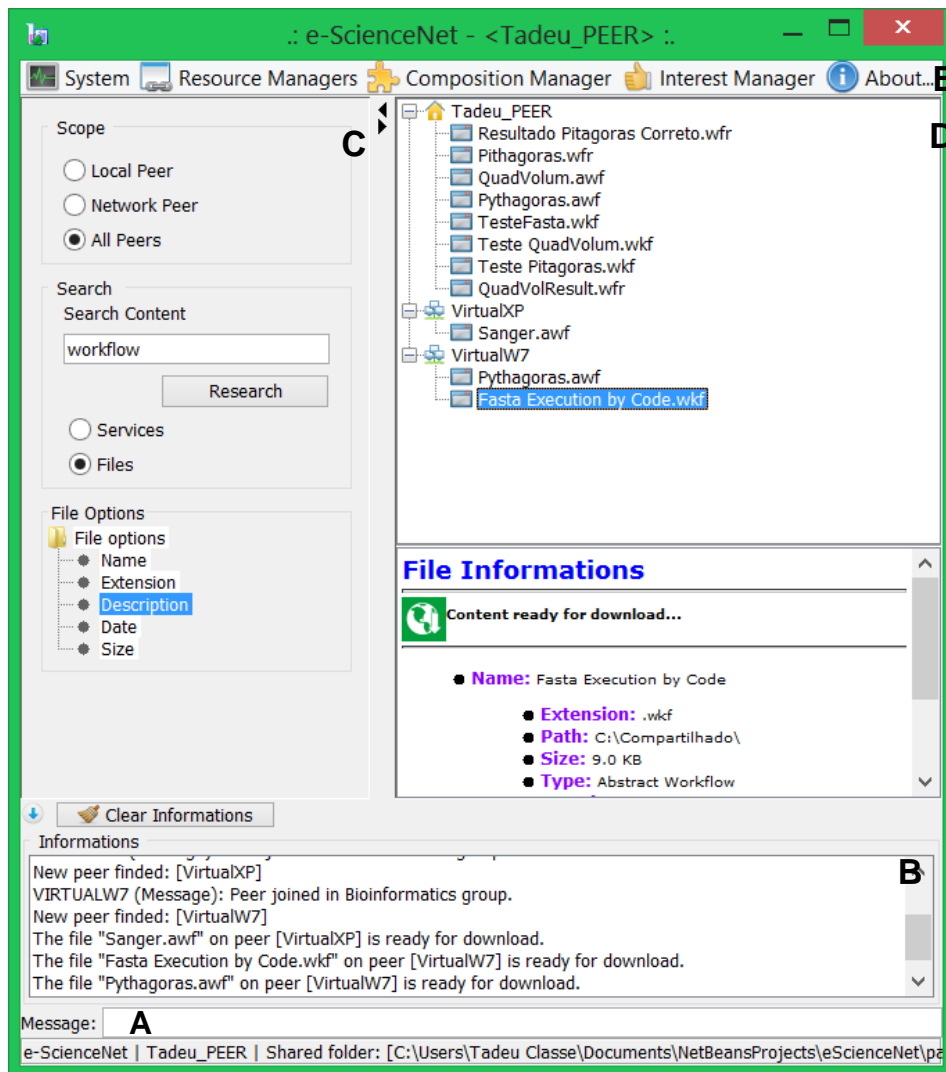


Figura 3.3: Tela Principal da e-ScienceNet

A Figura 3.3 detalha as principais funcionalidades do Gerente de Ferramentas. Na Figura 3.3 (A) é apresentado o sistema de *chat* da rede ponto a ponto, o qual é responsável pela comunicação em tempo real entre os *peers* conectados em cada um dos grupos. Logo acima do *chat* (Figura 3.3 (B)) são apresentadas informações como a descoberta de novos *peers*, a troca de mensagens feitas através do *chat* e a saída de um nó. Na Figura 3.3 (C) é possível visualizar funcionalidades relacionadas ao Gerente de Pesquisa, mais precisamente os campos para a pesquisa de arquivos na e-ScienceNet. Os resultados retornados por uma pesquisa são apresentados logo a frente (Figura 3.3 (D)), onde também é possível obter informações sobre cada um dos recursos retornados, selecionando um dado arquivo. Tais informações são obtidas através de consultas utilizando as ontologias de domínio relacionadas

aos *peers* da e-ScienceNet. A parte superior da figura apresenta os menus do sistema (Figura 3.3 (E)), relacionados aos gerentes da e-ScienceNet. A Figura 3.5 (A) mostra o menu *System*, o qual contém ferramentas administrativas como a exibição dos *peers* que estão conectados no momento no grupo de interesse (Figura 3.4), o menu de reconfiguração do JXTA e a opção de saída. O menu relacionado ao Gerente de Recursos é apresentado na Figura 3.5 (B), englobando o Gerente de Dados com funcionalidades como: inclusão e visualização de arquivos no sistema e a possibilidade de visualização de resultados gerados a partir da execução das composições de aplicações e o Gerente de Serviço com as seguintes funcionalidades: inclusão e visualização de serviços web como serviços e conectores e a execução de serviços sem que haja a composição ou conexão dos mesmos entre si.

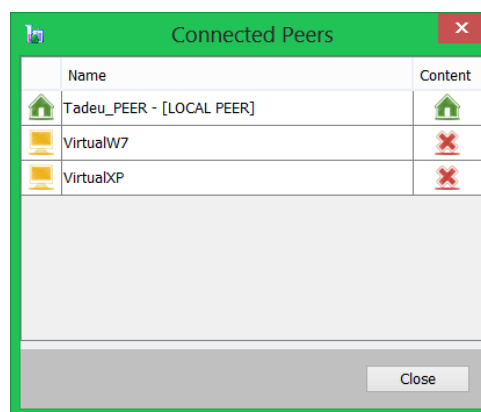


Figura 3.4: Tela de Visualização dos Peers Conectados no Grupo

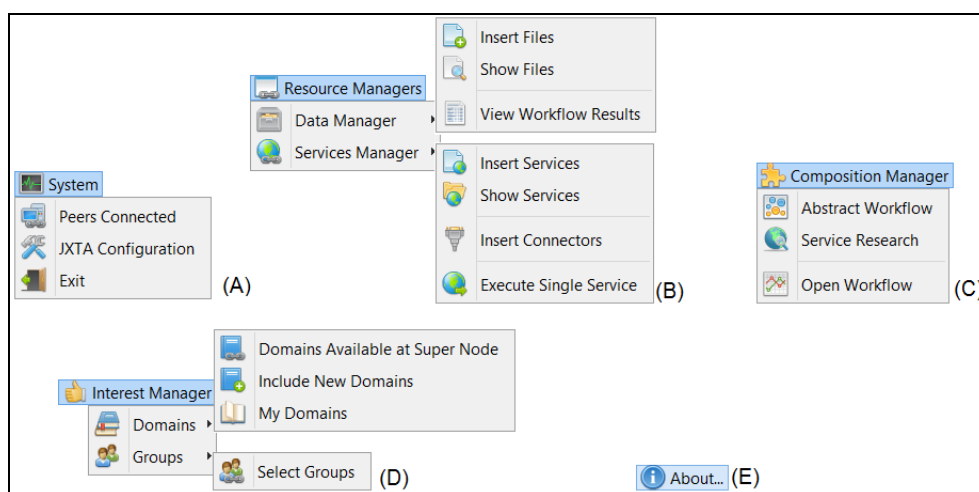


Figura 3.5: Menu Principal da e-ScienceNet

O menu relacionado ao Gerente de Composição pode ser visto na Figura 3.5 (C), com funcionalidades como: criação de *workflows* abstratos, busca de serviços web (aplicações científicas) para a criação de composições e carregamento de composições já existentes para a execução de um dado experimento científico. Na Figura 3.5 (D) é visto o menu relacionado ao Gerente de Interesse englobando funções relacionadas a busca de ontologias de domínios disponibilizadas pelos “Super Nós”, inclusão e visualização de novos domínios e grupos, permitindo ainda a seleção de outras comunidades semânticas.

Pode-se dizer que o Gerente de Ferramentas é a parte inicial da execução do sistema. Este processo de inicialização é formado por etapas (Figura 3.6) as quais são divididas em:

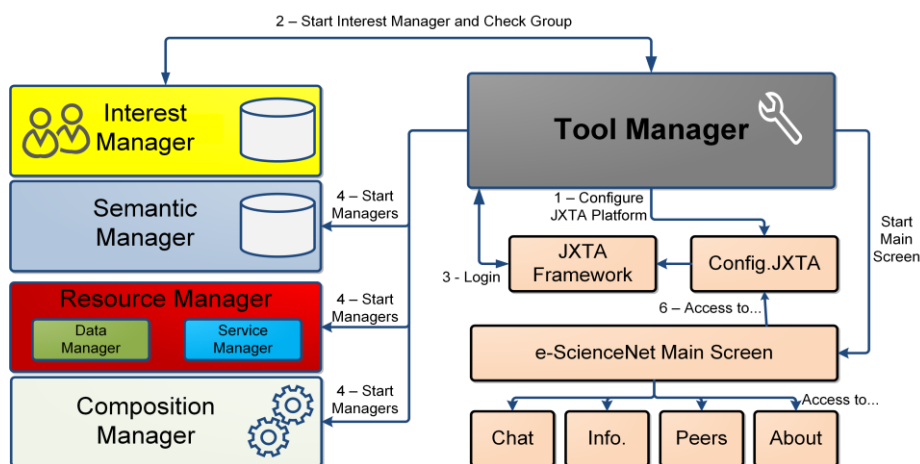


Figura 3.6: Arquitetura Funcional do Gerente de Ferramentas

- 1. Configuração da Plataforma JXTA:** conforme mencionado anteriormente, a e-ScienceNet é baseada no *framework* JXTA, sendo necessária a configuração dos parâmetros iniciais (IPs, nome do *peer*, protocolos e senhas). Tal configuração ocorre no momento do primeiro acesso ao sistema. Neste primeiro acesso, a tela de configuração da plataforma (Figura 3.7) é apresentada, sendo necessário informar dados como IP (*Internet Protocol*), endereços em protocolos HTTP (*Hyper Text Transfer Protocol*) e TCP (*Transmission Control Protocol*), login, senha, nome do *peer*, além do tipo de atuação do nó podendo ser: *Relay* (nós repetidores de informações responsáveis pelo roteamento na rede

P2P) e *Rendezvous* (nós que descobrem recursos e outros *peers* na rede P2P). Após essas configurações, o Gerente de Ferramentas é disponibilizado;

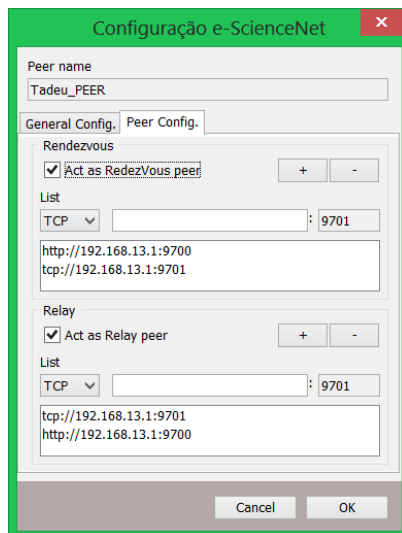


Figura 3.7: Tela de Configuração do JXTA na e-ScienceNet

2. Inicialização do Gerente de Interesse e busca pelos Grupos

Semânticos: A identificação do grupo é uma etapa importante para a e-ScienceNet. É neste momento que é definida qual será a localização do *peer* na rede, ou seja, a que grupos o *peer* pertence, e quais serão as comunicações e restrições aplicadas a ele. O processo de seleção de grupos semânticos é uma tarefa realizada pelo Gerente de Interesse e será abordada nas próximas seções. O importante a ser ressaltado neste momento, é que o Gerente de Interesse tem responsabilidade de descobrir se o nó está localizado em algum grupo ou se existe a necessidade de inclusão do mesmo em um grupo já existente. Essa seleção só é realizada na primeira conexão com a rede P2P. Uma vez que o nó esteja incluso em um grupo, a inicialização apenas verifica se as informações sobre o(s) grupo(s) estão corretas e avança para a próxima etapa;

3. Autenticação na e-ScienceNet:

A autenticação de usuários na e-ScienceNet é feita através do *framework* JXTA;

4. Inicialização dos demais Gerentes: a inicialização dos gerentes da e-ScienceNet ocorre após a validação dos grupos semânticos e a autenticação no sistema. Na primeira execução, são carregados os repositórios utilizados pelos gerentes como: repositórios de dados, serviços e ontologias. O carregamento de informações sobre o *peer* também ocorre nesta etapa, obtendo informações como seu nome, identificador de conexão e IP.

3.2.2 GERENTE DE SEMÂNTICA (SEMANTIC MANAGER)

O Gerente de Semântica (Figura 3.8) é o módulo da e-ScienceNet responsável por fornecer as informações semânticas ao sistema, como: informações de grupos semânticos, informações sobre recursos disponíveis, que são importantes para os Gerentes de Pesquisa, Dados e Serviço, bem como informações sobre a compatibilidade de serviços.

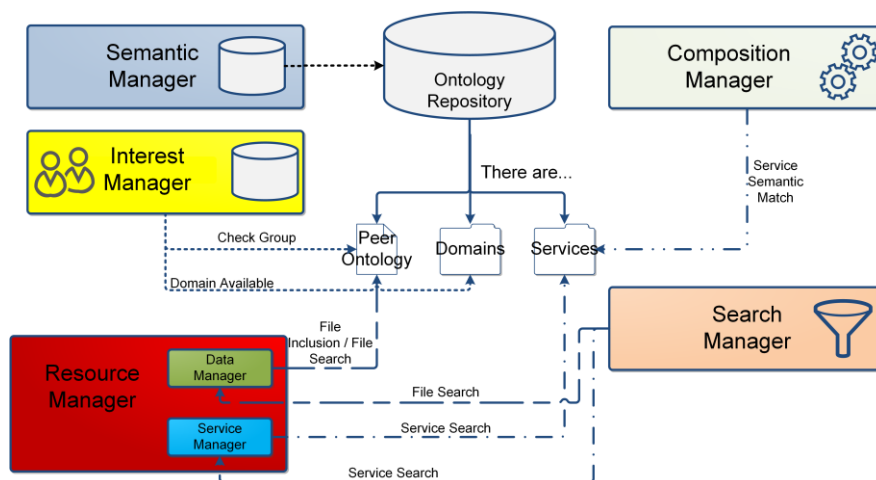


Figura 3.8: Gerente de Semântica

Internamente o Gerente de Semântica apresenta um repositório local chamado *Ontology Repository*, o qual é dividido em três módulos sendo eles:

- **PeerOntology:** também conhecida como ontologia de *peers*. Devido à característica de dispersão geográfica fornecida pela abordagem de redes P2P, a e-ScienceNet especifica pequenas ontologias em cada um dos seus nós. Estas pequenas ontologias são denominadas

PeerOntology (Figura 3. 9) e desempenham o papel de fornecer informações sobre o seu *peer* (Figura 3. 10) como IP, identificação, grupos semânticos conhecidos pelo peer; recursos como serviços e documentos incluídos na e-ScienceNet; ontologias de domínio selecionadas pelo nó; tecnologias de *web services* e outras. A partir da implementação de restrições e regras nestas ontologias e do uso de máquinas de inferência¹⁷ é possível obter informações que não estejam explícitas, (por exemplo, recursos existentes entre os peers e suas relações com as comunidades semânticas) (Figura 3. 9 (B)) e assim, realizar buscas por recursos, que não seriam possíveis caso não fosse utilizado uma ontologia. Ou seja, a *PeerOntology* é um modelo semântico importante no contexto da e-ScienceNet. Caso fosse utilizado apenas um modelo de dados simples, a descoberta por novas informações seria comprometida. A Figura 3. 11, apresenta um exemplo de uma *PeerOntology* para um dado nó, mostrando o modelo original Figura 3. 11 (A) e um exemplo de informações que a ontologia pode inferir sobre este peer Figura 3. 11 (B);

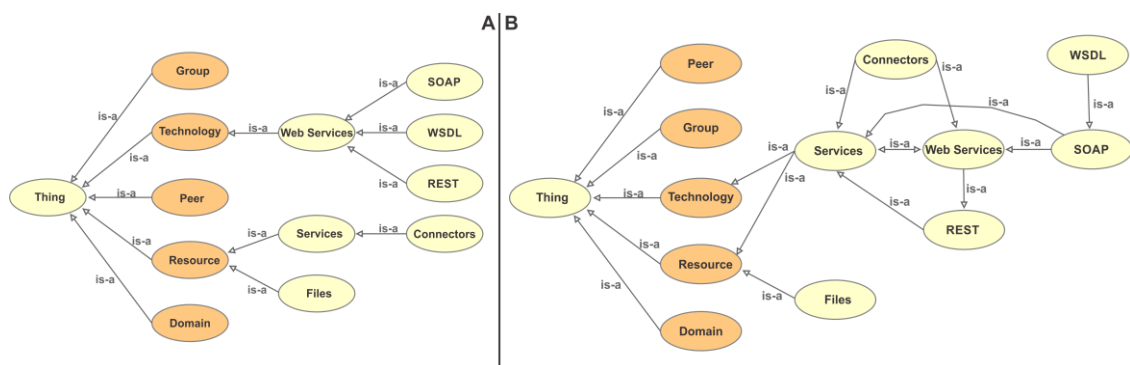


Figura 3. 9: A) PeerOntology modelo criado. B) PeerOntology modelo inferido

¹⁷ Mecanismos capazes de processar e inferir informações de ontologias, sendo que estas não estão implícitas em um modelo ontológico.

```

<owl:NamedIndividual rdf:about="&PeerOntology;Tadeu_PEER">
  <rdf:type rdf:resource="&PeerOntology;Peer"/>
  <peerIP rdf:datatype="&xsd:string">192.168.13.1</peerIP>
  <peerName rdf:datatype="&xsd:string">Tadeu_PEER</peerName>
  <peerIdJXTA rdf:datatype="&xsd:string">
    urn:jxta:uuid-59616261646162614A78746150325033943E152216574DEB91458999915C0C4603
  </peerIdJXTA>
  <hasGroup rdf:resource="&PeerOntology;Bioinformatics"/>
</owl:NamedIndividual>

```

Figura 3. 10: Informações sobre o peer contida na PeerOntology

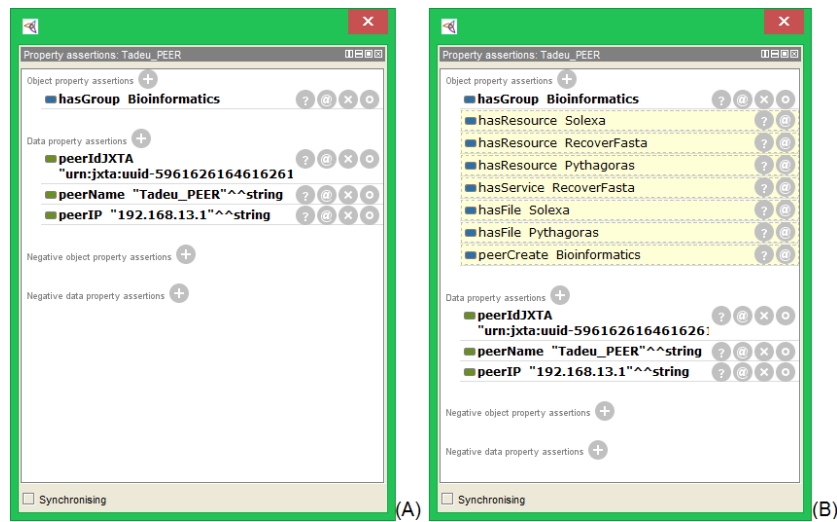


Figura 3. 11: Exemplo da PeerOntology de um nó

- Repositório de Domínios:** o Gerente de Semântica também armazena as ontologias de domínio que foram selecionados pelo *peer* para utilização. Quando um *peer* cadastra ou requisita uma ontologia de domínio, o sistema armazena o arquivo em um repositório local chamado *Domain Ontology Repository* e persiste informações como nome do grupo, localização e URI na classe *Domain* da *PeerOntology*. Isso ocorre no intuito de armazenar as informações sobre quais ontologias de domínio um *peer* tem conhecimento e associar estas ontologias a recursos existentes;
- Repositório de serviços semanticamente anotados:** neste repositório são armazenados todos os arquivos OWL-S gerados a partir das anotações semânticas de um serviço. Essas ontologias são utilizadas pelo Gerente de Semântica a fim de pesquisar informações sobre os

serviços disponibilizados na e-ScienceNet, além de serem utilizadas para a compatibilização semântica e sintática de serviços para a composição de um experimento científico.

A maioria das consultas e pesquisas realizadas por este gerente é feita através do uso de SPARQL (Figura 3. 12).

```
private boolean peerExists(String nome){
    try{
        OntModel model = this.getPeerOntModel();

        String sql = "PREFIX t: <"+ URI_PEERGROUPS_ONTOLOGY +">\n" +
            "SELECT ?name ?id ?ip\n" +
            "WHERE {\n" +
            "\t?x t:peerName \""+ nome +"\".\n" +
            "\t?x t:peerName ?name\n" +
            " }";

        Query query = QueryFactory.create(sql);
        QueryExecution qe = QueryExecutionFactory.create(query, model);
        ResultSet results = qe.execSelect();

        ResultSetFormatter.out(results);

        return true;
    }catch(Exception e){
        System.out.println(e.getMessage());
        return false;
    }
}
```

Figura 3. 12: Exemplo de consulta SPARQL para verificar peers.

No decorrer da apresentação dos demais gerentes, várias operações do Gerente de Semântica também serão apresentadas como o processo de consulta de documento e serviços, descoberta de grupos semânticos, comparação de serviços para as composições e outras mais.

3.2.3 GERENTE DE INTERESSE (INTEREST MANAGER)

Este módulo do sistema é o responsável pela configuração dos interesses de cada um dos *peers*. É o Gerente de Interesse (Figura 3.13) quem controla o conceito de grupos semânticos.

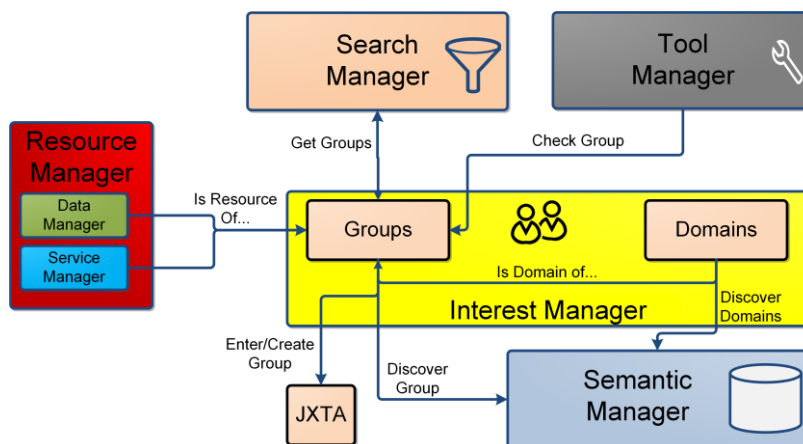


Figura 3.13: Fluxo de Controle do Gerente de Interesse

Um grupo semântico pode ser caracterizado pelos interesses de pesquisa de um grupo de cientistas, agrupando-os, portanto em comunidades com a mesma afinidade científica. A adoção deste conceito ajuda no controle de tráfego dos dados na rede da e-ScienceNet, pois, assim, a comunicação dos *peers* é feita apenas dentro de uma comunidade, diminuindo o tráfego de informação em toda a rede. Um *peer* pode se localizar dentro de uma ou mais comunidades semânticas, baseando-se em seus interesses de pesquisa.

Neste contexto, outro conceito importante é o de super nó (*super peer*). Um super nó geralmente é o *peer* de uma rede P2P com um grande poder de processamento, respondendo pelo roteamento do tráfego da rede e manutenção de tabelas de índices [CHIRITA et al., 2006]. No contexto da e-ScienceNet, um super nó pode ser caracterizado como o *peer* que criou uma determinada comunidade semântica, ou seja, este *peer* é o responsável por gerenciar tal grupo de interesse. Outra função importante de um super nó, é o gerenciamento de ontologias de domínio. Cada grupo de interesse possui uma ou mais ontologias de domínio e o super nó é responsável por manter tais ontologias, além de controlar versões diferentes das mesmas.

3.2.3.1 GRUPOS SEMÂNTICOS

Como dito na seção 3.2 e seção 3.2.1, ao se conectar pela primeira vez na e-ScienceNet, um usuário deve selecionar um grupo semântico de acordo com as suas características de pesquisa. Porém, de imediato, o novo nó não

têm nenhum conhecimento de quais são as comunidades que existem na e-ScienceNet. O processo de pesquisa de grupos de interesse pode ser contemplado através da Figura 3. 14 e conta com as etapas abaixo:

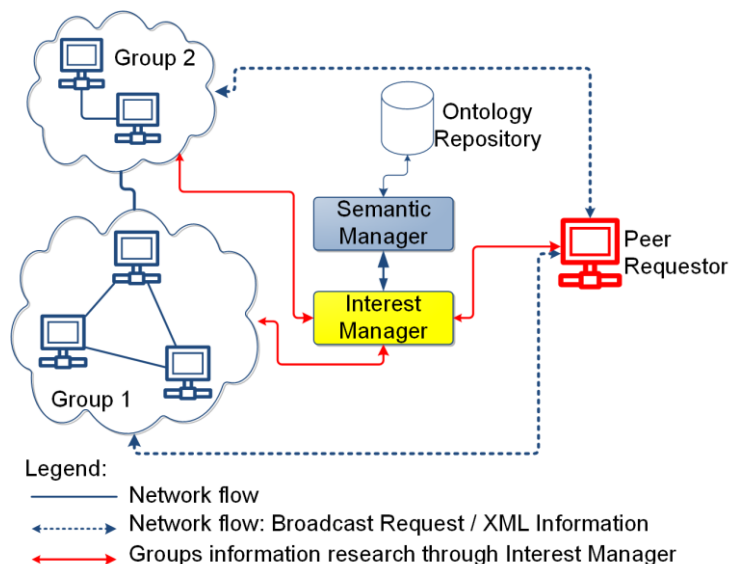


Figura 3. 14: Pesquisa semântica de grupos de interesse

1. **Requisição de Informação:** o novo *peer*, não tendo conhecimento da existência de quais grupos estão disponíveis na e-ScienceNet envia uma mensagem de *broadcast* a todos os nós, fazendo a verificação de quais são os grupos que eles têm conhecimento. Este é o único momento no sistema em que um *peer* realiza a comunicação com todos os nós existentes sem que haja controle por grupos semânticos;
2. **Busca de Grupos Semânticos:** ao receber a mensagem de *broadcast*, os demais *peers* por meio de seu Gerente de Semântica realiza uma pesquisa em sua *PeerOntology* (Figura 3. 15) a fim de encontrar os grupos de que possuem conhecimento e suas informações principais como: nome, identificação do JXTA, área, descrição, data de criação e dono (super nó).

```

public String obterXMLGruposOnOWL(){
    String retorno = "";

    OntModel model = this.getPeerOntModel();

    OntClass group = model.getOntClass(URI_PEERGROUPS_ONTOLOGY + "Group");
    List<PeerGroupModel> lstPeerGroupModels = new ArrayList<>();

    Iterator i = null;
    for(i = group.listInstances(); i.hasNext();){
        Individual individuo = (Individual) i.next();

        PeerGroupModel peerGroup = new PeerGroupModel();

        Property groupName = model.getProperty(URI_PEERGROUPS_ONTOLOGY + "groupName");
        peerGroup.setGroupName(individuo.getPropertyValue(groupName).asLiteral().getString());

        lstPeerGroupModels.add(peerGroup);
    }

    XStream xstream = new XStream(new DomDriver());
    retorno = xstream.toXML(lstPeerGroupModels);

    return retorno;
}
}

```

Figura 3. 15: Parte do código de busca de grupos semânticos

3. **Geração de XML resposta:** após um *peer* encontrar as informações sobre os grupos, o mesmo gera um arquivo XML (veja apêndice A) contendo todas as informações relevantes para o conhecimento do novo *peer*. Esse recurso é utilizado na e-ScienceNet para facilitar a troca de informações entre os *peers*, fazendo uma comunicação direta entre seus canais de mensagens;
4. **Recebimento de Informações:** os *peers* da e-ScienceNet retornam uma nova mensagem para o novo *peer*, contendo todas as informações que conhecem. O novo *peer* então, ao receber essas informações realiza um pré-processamento, verificando o recebimento de informações redundantes, por exemplo, dois *peers* podem enviar informações sobre um mesmo grupo, então o novo nó deve filtrar essas informações obtendo somente os dados inéditos. A partir dessa verificação, o novo *peer* grava as informações como URI e nome do grupo em sua ontologia para que ela sirva de conhecimento para futuras conexões de novos nós ou até mesmo para a sua conexão em outras comunidades semânticas;

5. **Seleção de Grupo:** com as informações sobre os grupos disponíveis na e-ScienceNet salvas na *PeerOntology*, o novo nó deve selecionar os grupos de interesse e se conectar a estes. A Figura 3.16 mostra a *interface* de seleção de grupos semânticos no Gerente de Interesse.

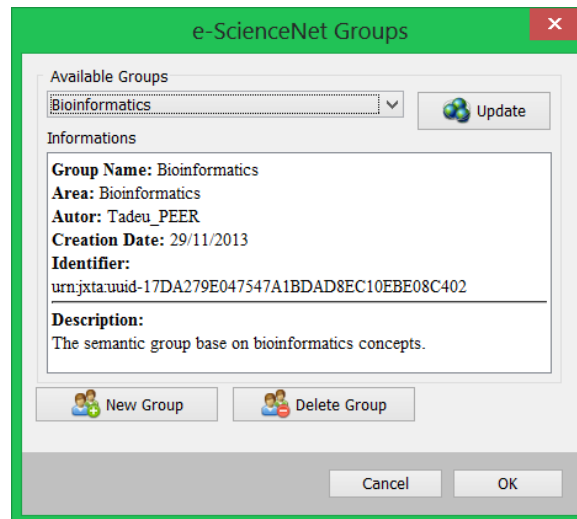


Figura 3.16: Interface de seleção de grupo semântico

Outra opção à seleção de comunidades semânticas é a criação de novos grupos. Caso um cientista não esteja satisfeito com os grupos existentes, ele pode criar a sua própria comunidade semântica (Figura 3. 17), tornando-se, assim, o super nó deste novo grupo semântico. Com a criação de um novo grupo de interesse, o mesmo é salvo na *PeerOntology* do nó e se torna disponível para os demais *peers* da e-ScienceNet.

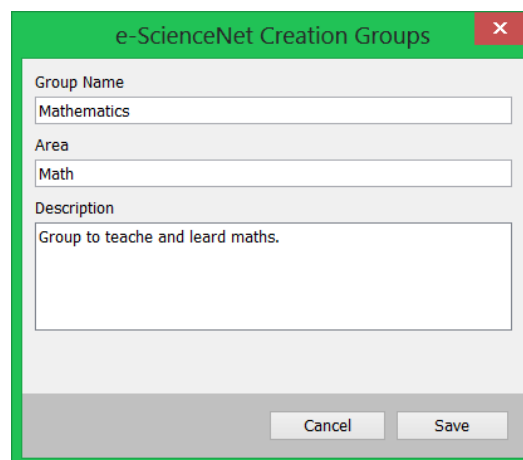


Figura 3. 17: Criação de novos grupos semânticos

3.2.3.2 ONTOLOGIAS DE DOMÍNIO

Conforme apontado na seção anterior os grupos semânticos são responsáveis por agrupar os *peers* da e-ScienceNet baseando-se nos interesses de pesquisas dos cientistas. Porém, nessas comunidades, o conhecimento é compartilhado por meio das ontologias de domínio. Uma ontologia de domínio contém termos e regras úteis que podem ser utilizadas para a criação de experimentos e como guias de pesquisa. Os grupos de interesse podem ter diversas ontologias de domínios, o que faz com que o poder de conhecimento da comunidade se amplie.

Essas ontologias são gerenciadas pelos super nós das comunidades semânticas e são persistidas no repositório *Domain Ontology Repository*. Assim, os super nós, além das funcionalidades de um nó normal da e-ScienceNet, acumulam a funcionalidade de gerência dessas ontologias, como: versionamento; download e upload por outros *peers* (Figura 3.18), além do cadastramento de novas ontologias (Figura 3.19).

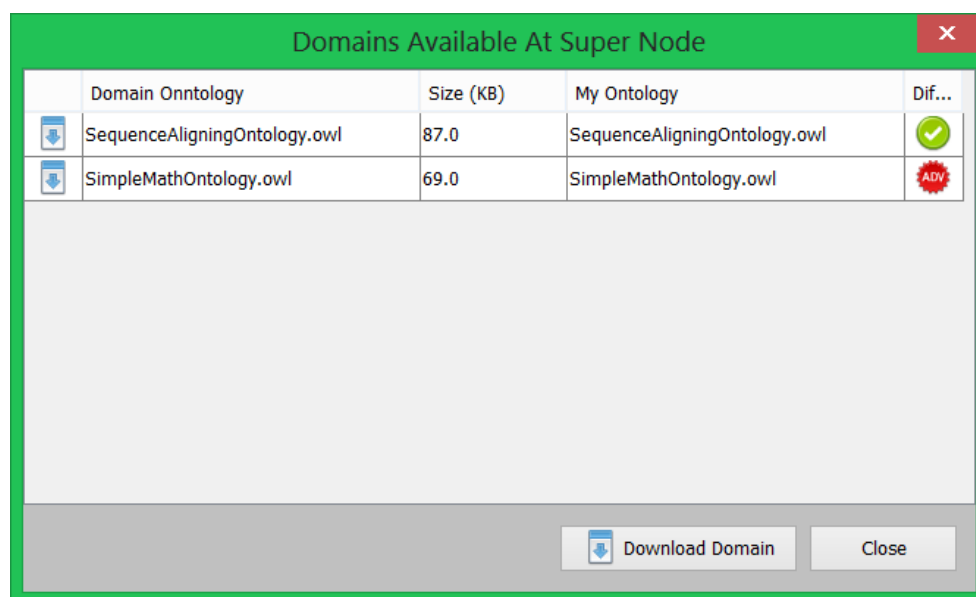


Figura 3.18: Ontologias de domínio disponíveis para download no Super nó da comunidade semântica

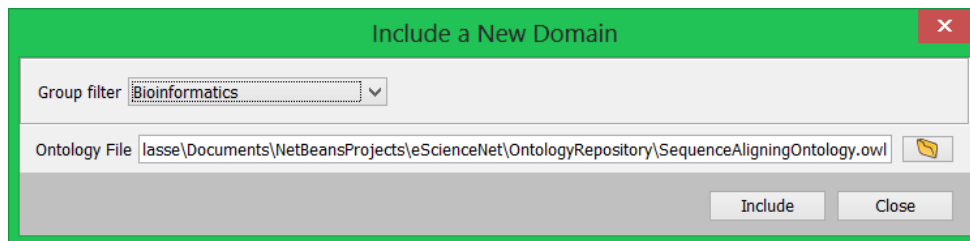


Figura 3.19: Cadastro de nova ontologia de domínio e upload para o super nó

A Figura 3. 20, mostra o esquema de requisição de ontologias de domínio disponíveis no super nó e o processo de disponibilização de uma nova ontologia de domínio para um dado grupo semântico, feito por um *peer* comum. O primeiro processo se dá através de uma requisição feita ao Gerente de Interesse para verificar quais são as ontologias de domínio disponíveis no super nó da comunidade semântica. O super nó ao receber a requisição, retorna uma mensagem com os domínios disponíveis. Assim, o cientista pode fazer o *download* da mesma em seu próprio *peer* considerando o *Domain Ontology Repository*¹⁸. Ao fazer o *download* da ontologia, a e-ScienceNet, por meio do seu gerente de semântica, realiza a verificação de consistência a fim de determinar se não existem informações inconsistentes (conexões entre termos e relações semânticas). Se tal validação ocorrer sem problemas, informações da ontologia de domínio, tais como sua URI e nome, são persistidas na *PeerOntology* para que os recursos dos nós possam ser conectados através dos termos dos domínios e sua utilização na pesquisa semântica de serviços web.

A Figura 3. 20 detalha ainda, como é o processo de alteração de uma ontologia do domínio ou o cadastramento de uma nova. Com a localização do arquivo, a e-ScienceNet realiza o processo de envio para o nó responsável pelo grupo. O super nó, então, faz o mapeamento de tal ontologia e verifica se

¹⁸ Devido às limitações tecnológicas da plataforma JXTA e para não sobrecarregar a rede, foi decidido que seria realizado o download de ontologias de domínio para um repositório local dentro do *peer*, pois, toda e qualquer consulta realizada na e-ScienceNet requer um grande número de retorno de informações inferidas nas ontologias de domínio.

a mesma já existe em seu repositório. Caso exista, o *peer* faz o versionamento numérico da mesma, incluindo números de versão no arquivo.

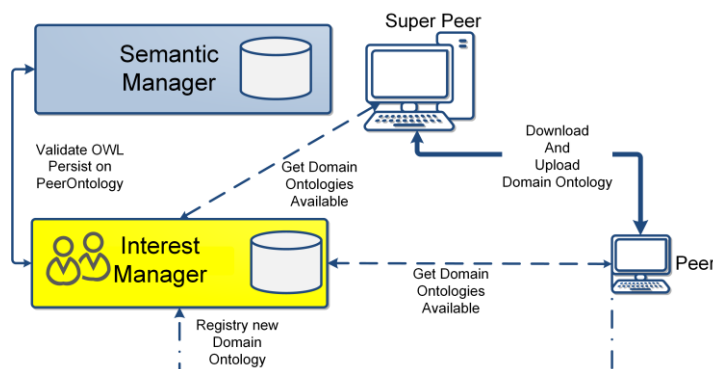


Figura 3. 20: Download e Upload de Ontologias de Domínio

O usuário da e-ScienceNet, também conta com uma *interface* para verificação e exclusão de ontologias de domínio, permitindo que o mesmo possa gerenciar os domínios que deseja adicionar a sua pesquisa.

3.2.4 GERENTE DE RECURSOS (RESOURCE MANAGER)

O Gerente de Recursos é o módulo da e-ScienceNet responsável por controlar os recursos disponibilizados pela rede P2P como: arquivos, documentos, programas, modelos de *workflows* e serviços web. Este gerente engloba dois outros módulos da e-ScienceNet: o Gerente de Dados, responsável pelo gerenciamento de arquivos; e o Gerente de Serviços, responsável pelos serviços web. Estes dois módulos serão detalhados a seguir.

3.2.5 GERENTE DE DADOS (DATA MANAGER)

O Gerente de Dados (Figura 3.21) desempenha o papel de gerenciamento e controle de arquivos englobando inclusão, visualização e compartilhamento.

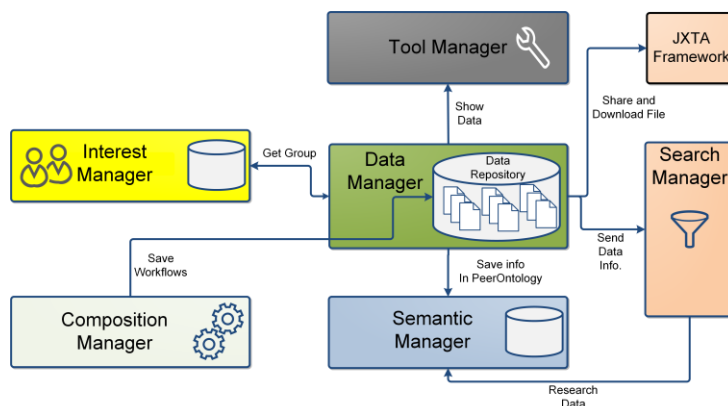


Figura 3.21: Comunicação do Gerente de Dados

Através de uma *interface* gráfica, o usuário pode incluir arquivos (Figura 3.22) para compartilhamento pelos demais nós do grupo de interesse. Informações como localização física do arquivo no *peer*, tamanho, data de criação, extensão, tipo, nome e descrição podem ser recuperados automaticamente no momento da inclusão de um novo documento.

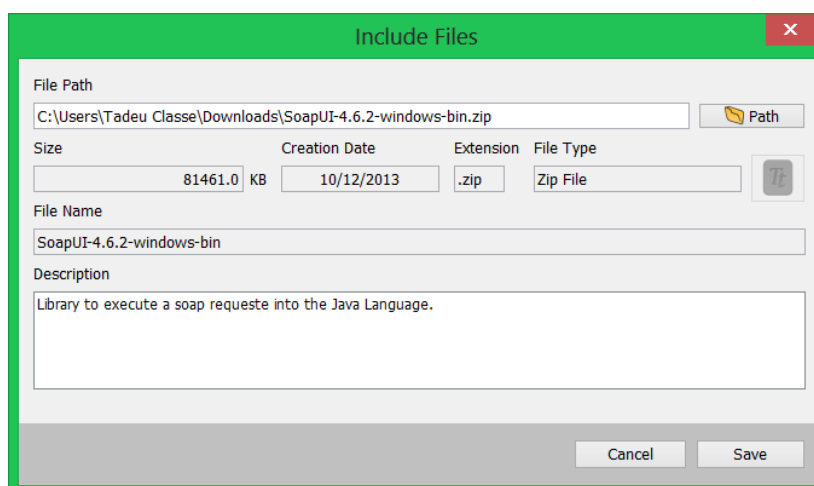


Figura 3.22: Interface de inclusão de documentos

Ao incluir um documento, o Gerente de Dados salva seu dados na *PeerOntology* (Figura 3.23) do *peer* a fim gerar informações para pesquisa e gerenciamento do conteúdo existente (Figura 3.24), podendo acessar o arquivo, filtrar por grupos de interesse e excluir o arquivo compartilhado. Além disso, envia uma notificação para a plataforma JXTA informando que existe um novo conteúdo disponível para compartilhamento na e-ScienceNet.

```

<!-- http://www.esciencenet.com.br/tadeuclasse/esciencenet/PeerOntology.owl#TeamViewer_Setup_pt-dix -->
<owl:NamedIndividual rdf:about="#PeerOntology:TeamViewer_Setup_pt-dix">
  <rdf:type rdf:resource="#PeerOntology:Files"/>
  <fileExtension rdf:datatype="#xsd:string">.exe</fileExtension>
  <fileCreationDate rdf:datatype="#xsd:string">26/04/2013</fileCreationDate>
  <fileSize rdf:datatype="#xsd:decimal">4791.0</fileSize>
  <filePath rdf:datatype="#xsd:string">C:\Users\Tadeu\Downloads\</filePath>
  <fileType rdf:datatype="#xsd:string">Generic Binary Stream</fileType>
  <fileDescription rdf:datatype="#xsd:string">No Description</fileDescription>
  <fileName rdf:datatype="#xsd:string">TeamViewer_Setup_pt-dix</fileName>
  <pertenceOneGroup rdf:resource="#PeerOntology:Bioinformatica"/>
  <isFileOf rdf:resource="#PeerOntology:Tadeu_PEER"/>
</owl:NamedIndividual>

```

Figura 3.23: Informações sobre arquivo persistido na PeerOntology

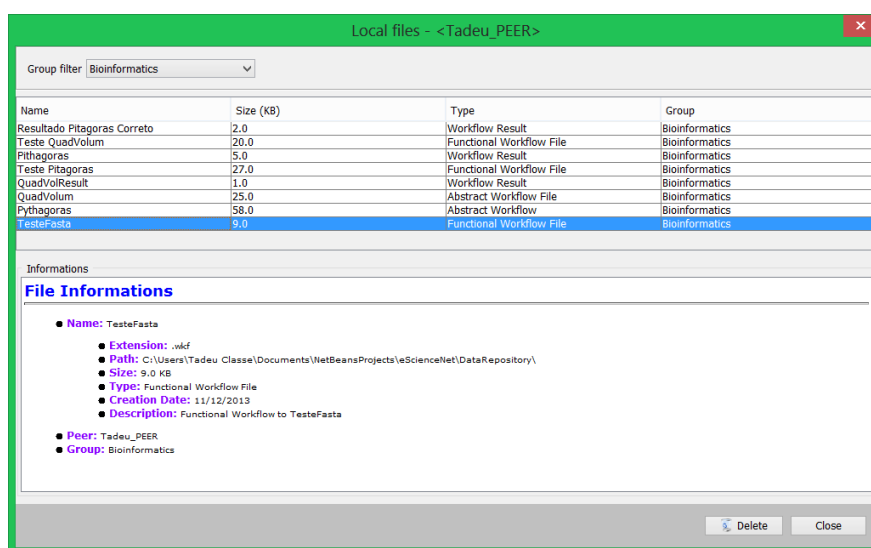


Figura 3.24: Interface de gerenciamento de arquivos

Neste contexto, o Gerente de Dados atua como provedor de informações relacionadas aos arquivos disponíveis nos *peers*, cadastrando novos conteúdos e provendo informações para as pesquisas realizadas pelo Gerente de Pesquisa.

3.2.6 GERENTE DE SERVIÇOS (SERVICE MANAGER)

Assim como o Gerente de Dados, o Gerente de Serviço (Figura 3.25) é um módulo pertencente ao Gerente de Recursos.

O Gerente de Serviços é responsável por controlar os serviços web no contexto da e-ScienceNet, permitindo sua inclusão, execução, pesquisa de recursos, além de ser responsável por prover informações dos serviços para os Gerentes de Pesquisa e de Ferramentas.

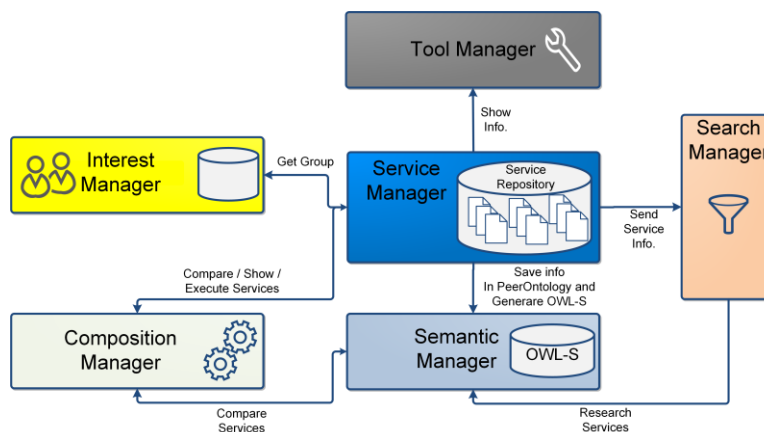


Figura 3.25: Modelo de Comunicação do Gerente de Serviço

No contexto da e-ScienceNet, os serviços web são divididos em duas categorias principais: os serviços científicos, sendo estes os serviços web que podem ser pesquisados para criar composições de serviços e especificar um experimento na e-ScienceNet; e os conectores: que são serviços web criados para possibilitar a conexão de serviços científicos incompatíveis, ou seja, cujos parâmetros de entrada e saída e/ou restrições não são compatíveis. Estes conectores têm como papel possibilitar a conexão destes serviços. Os conectores serão detalhados na seção 3.3.6.2.

Nesse sentido, a inclusão de serviços na e-ScienceNet se dá através da inserção de uma URL (*Uniform Resource Location*) de um documento WSDL na *interface* de criação, a qual foi desenvolvida para identificar quais são as operações que este serviço comporta e cada um de seus parâmetros de entrada e saída, permitindo a anotação semântica destes com termos das ontologias gerando ao final um serviço anotado semanticamente por meio de ontologias OWL-S.

Assim, a inserção de novos serviços na e-ScienceNet passa por algumas etapas sendo elas:

1. **Inclusão de Serviços Web (WSDL):** como primeira e importante etapa da disponibilização de novos serviços na e-ScienceNet, tem-se a inclusão de um serviço web descrito através da linguagem WSDL. O sistema é capaz de identificar as operações disponíveis, nome das operações, descrições, URI (*Uniform Resource Identifier*) dos *namespaces* associados ao serviço e seus parâmetros de entrada e saída. Na Figura

3.26, é possível ver o processo de inclusão. Nesta imagem, uma URL foi informada e o sistema processa seu conteúdo gerando as operações e suas informações, como: descrição, operações, parâmetros, nome dos serviços e seus *name spaces*;

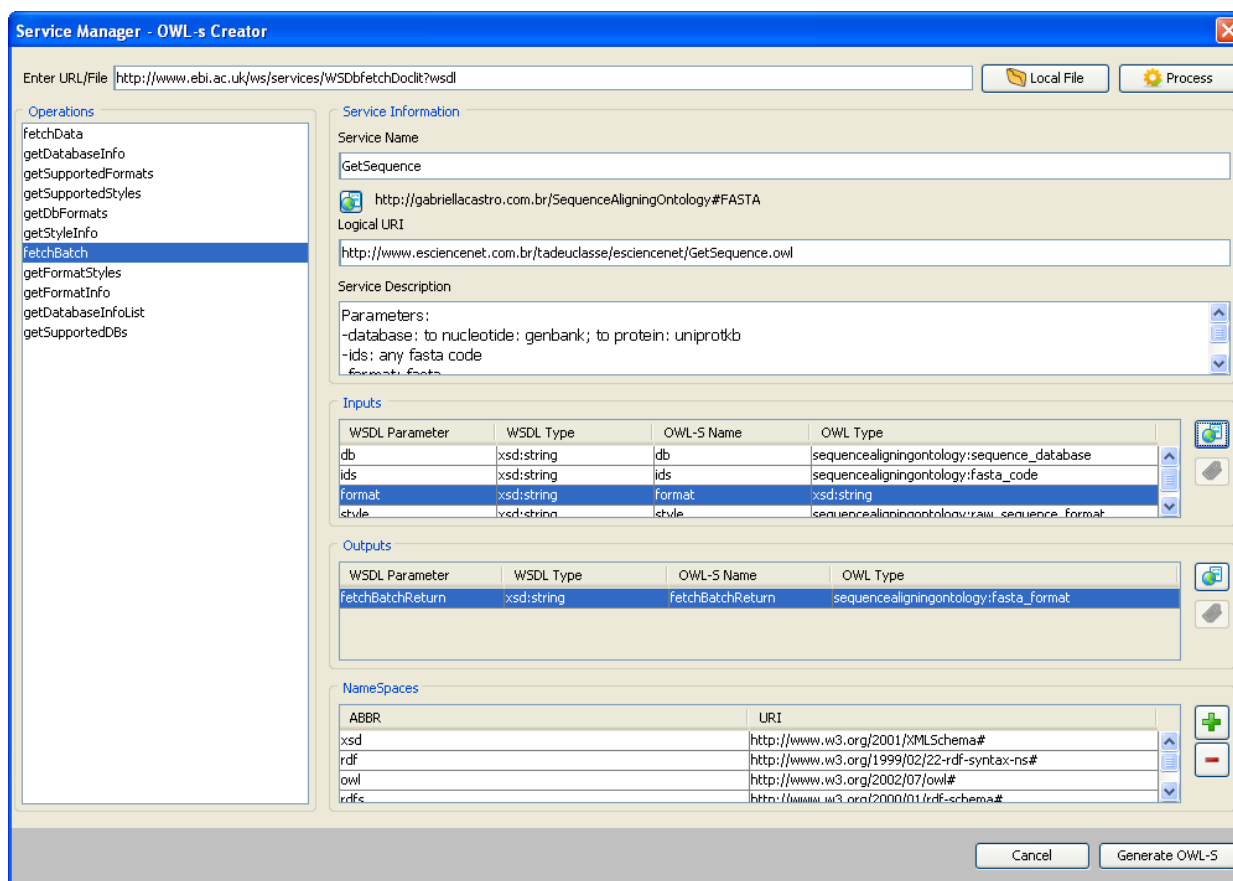


Figura 3.26: Interface de inclusão de serviços web

2. **Descrição semântica para o serviço:** a anotação semântica do serviço constitui a associação do serviço com termos de ontologias de domínio. A *interface* apresentada na Figura 3.26, possui mecanismos para realizar a anotação do serviço como pode ser observado na Figura 3.27. O Gerente de Semântica é capaz de exibir os termos semânticos das ontologias de domínio que estejam relacionados com nome do serviço web (Figura 3.28 (A)), utilizando, caso necessário, máquinas de inferência no processo. Além disso, caso a *interface* não seja capaz de sugerir termos, o cientista pode selecionar manualmente algum dos termos através de um processo de busca manual (Figura 3.28 (B)) nas ontologias de domínio. Tal função se fez necessária na e-ScienceNet

para facilitar o processo de pesquisas por serviços, pois, como poderá ser visto na seção 3.2.7, o Gerente de Pesquisa ao buscar por algum serviço na rede, consegue trazer resultados mais precisos caso haja algum termo semântico associado ao serviço web. Se isso não fosse realizado, a pesquisa poderia retornar quaisquer resultados somente considerando as suas conexões, apenas cruzando informações de entrada e saída;



Figura 3.27: Anotações

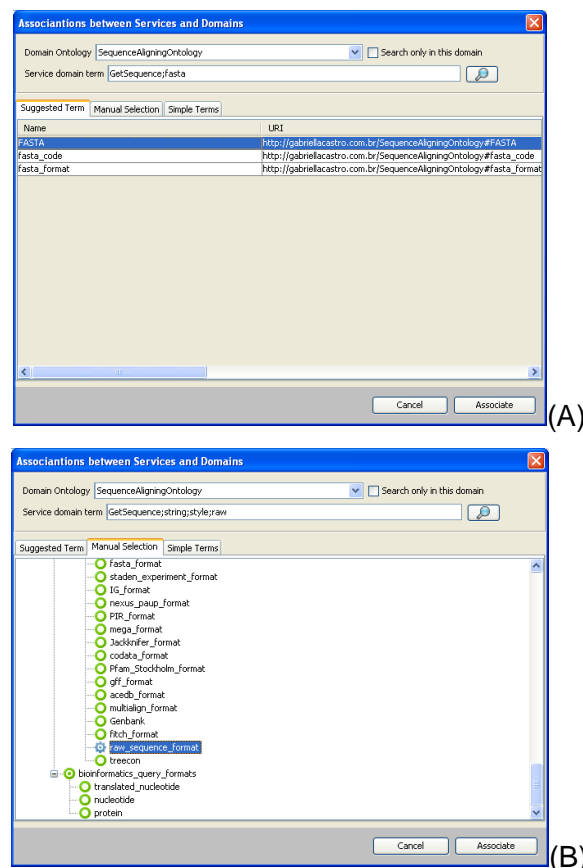


Figura 3.28: A) Anotação semântica por termos sugeridos. B) Anotação semântica por seleção manual

3. **Anotação semântica de parâmetros:** consiste em, assim como na etapa anterior, utilizar termos das ontologias de domínio para anotar os parâmetros de entrada e saída de um serviço, dando a eles significado semântico (Figura 3.29). Da mesma maneira que a anotação semântica de serviços, a e-ScienceNet mostra a *interface* de sugestão de termos disponíveis nas ontologias de domínio e a opção de seleção manual. Porém, neste caso, o sistema também exhibe a opção de realizar a anotação do parâmetro utilizando termos simples disponíveis no XSD (*XML Schema Definition*), como textos, inteiros, lógicos e etc., o que pode ser visualizado na Figura 3.30;

WSDL Parameter	WSDL Type	OWL-S Name	OWL Type
db	xsd:string	db	sequencealigningontology:sequence_database
ids	xsd:string	ids	sequencealigningontology:fasta_code
format	xsd:string	format	xsd:string
style	xsd:string	style	sequencealigningontology:raw_sequence_format

WSDL Parameter	WSDL Type	OWL-S Name	OWL Type
fetchBatchReturn	xsd:string	fetchBatchReturn	sequencealigningontology:fasta_format

Figura 3.29: Anotação semântica de parâmetros

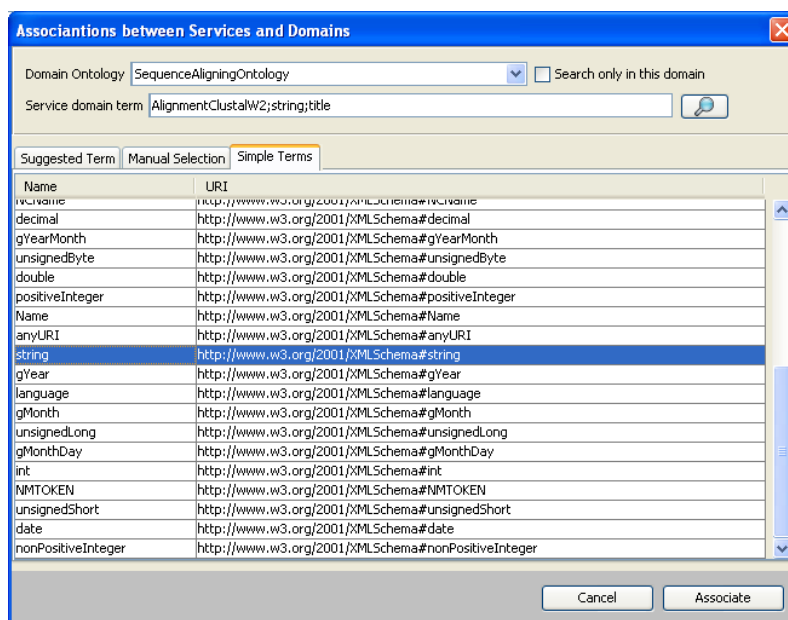


Figura 3.30: Anotação de parâmetros por XSD

4. **Criação de OWL-S:** após as etapas de anotação semântica, a e-ScienceNet disponibiliza a possibilidade de criação da ontologia do

serviço no padrão OWL-S. Este processo consiste na criação da anotação semântica do serviço por meio do Gerente de Semântica, o qual varre todas as informações geradas na *interface* de inclusão do serviço e baseado na OWL-S API [OWL-S API, 2013], cria a ontologia do serviço. O Gerente de Semântica também precisa verificar se a ontologia criada é consistente (Figura 3.31) para depois poder armazená-la em seu repositório de serviços semânticos e, também, persistir suas informações na *PeerOntology*.

```
private boolean owlsValidation(File owlsFile){
    try{
        boolean valid = false;

        OWLKnowledgeBase kb = OWLFactory.createKB();
        kb.setReasoner("Pellet");
        kb.read(owlsFile.toURI());
        valid = kb.isConsistent();

        if(!valid)
            throw new Exception("The semantic service description (OWL-s) wasn't generated with success!");
        else {
            JOptionPane.showConfirmDialog(null, "Semantic service description (OWL-s) was generated with success!",
                ".: e-ScienceNet .:", JOptionPane.DEFAULT_OPTION,
                JOptionPane.INFORMATION_MESSAGE,
                (new javax.swing.ImageIcon(getClass().getResource("/images/information.png"))));
        }

        return true;
    }catch(Exception e){
        JOptionPane.showMessageDialog(null, e.getMessage(), ".: e-ScienceNet .:", JOptionPane.ERROR_MESSAGE,
            (new javax.swing.ImageIcon(getClass().getResource("/images/error.png"))));

        return false;
    }
}
```

Figura 3.31: Código de verificação de consistência de OWL-S

Conforme descrito na seção 2.2.3.1, a OWL-S é baseada na seguinte estrutura: *ServiceProfile*, fornece informações sobre o que o serviço faz; *ServiceModel*, fornece informações sobre como o serviço funciona; e o *ServiceGrounding*, fornece informações sobre os protocolos de comunicação do serviço. Como pode ser visto na Figura 3.32, a e-ScienceNet é capaz de gerar tal estrutura para a descrição semântica de um serviço web. Entretanto, foram encontradas algumas deficiências do OWL-S em relação ao seu uso pela e-ScienceNet, como a anotação semântica de serviços web e seus parâmetros, o armazenamento do tipo sintático entre entradas e saídas e a informação sobre o nome real do serviço web para a sua execução. Para corrigir tal problema, uma extensão da ontologia OWL-S foi feita no contexto deste

trabalho, que foi nomeada como OWL-SE (OWL-SExtended), e será detalhada na seção 3.2.6.1.

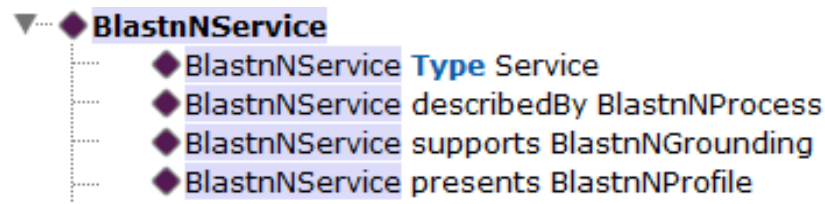


Figura 3.32: Estrutura OWL-S de um serviço incluído na e-ScienceNet

Assim como o Gerente de Dados, este módulo da e-ScienceNet conta com uma *interface* gráfica para gerenciar serviços e conectores disponíveis no sistema, podendo filtrá-los de acordo com os grupos semânticos do *peer*, excluí-los ou obter informações sobre os termos semânticos e ontologias conectadas a eles por meio de um gráfico. Tal *interface* é apresentada na Figura 3.33.

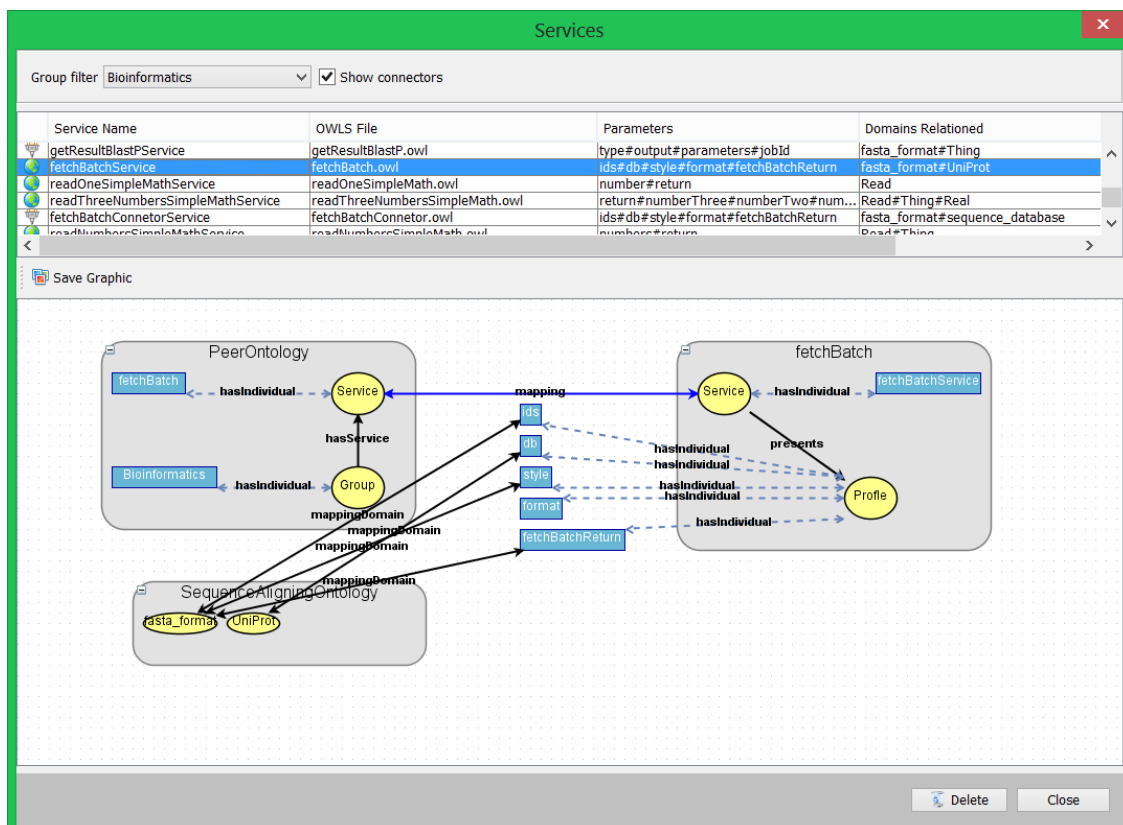


Figura 3.33: Interface de gerenciamento de serviços disponíveis no peer.

3.2.6.1 OWL-SEXTENDED

Como mencionado na seção anterior, para a anotação semântica de serviços, a e-ScienceNet se baseia no padrão OWL-S, a fim de obter informações sobre os serviços. Entretanto, foram detectados problemas tanto na associação do serviço a termos de uma ontologia de domínio, quanto na inclusão de parâmetros. Na versão padrão da OWL-S somente os parâmetros explícitos no WSDL são anotados, não sendo possível anotar parâmetros complexos, ou seja, não foram encontradas formas de mapear no OWL-S a relação de sub-parâmetros em tipos complexos existentes em serviços web.

Para resolver tais problemas, foi feita uma extensão no padrão OWL-S mantendo as suas características originais e implementando propriedades úteis no contexto da e-ScienceNet, sendo elas:

- **Data Property domainTerm:** essa propriedade foi definida para as instâncias de *Service* em um OWL-S devido a necessidade de anotação semântica do serviço descrito na seção anterior, ou seja, essa propriedade armazena o termo semântico da ontologia de domínio associado com a operação do *web service* (Figura 3.34);

```
<owl:NamedIndividual rdf:about="&j.0;fetchBatchService">
  <rdf:type rdf:resource="&service;Service"/>
  <domainTerm rdf:datatype="&xsd:anyURI">
    http://gabriellacastro.com.br/SequenceAligningOntology#fasta_format
  </domainTerm>
  <service:supports rdf:resource="&j.0;fetchBatchGrounding"/>
  <service:describedBy rdf:resource="&j.0;fetchBatchProcess"/>
  <service:presents rdf:resource="&j.0;fetchBatchProfile"/>
</owl:NamedIndividual>
```

Figura 3.34: Data Property *domainTerm* no OWL-SE

- **Data Property serviceRealName:** outra *dataProperty* que foi incluída no modelo OWL-S foi a *serviceRealName*, que é responsável por armazenar o nome real da operação do serviço. O nome do serviço pode ser alterado de acordo com a necessidade do usuário como pode ser visualizado na Figura 3.27. Entretanto, para que o serviço possa ser executado, existe a necessidade do armazenamento do nome real da operação. Sendo assim, o mesmo foi incluído dentro das instâncias de

Profile, por fornecer informações de execução do *web service* (Figura 3.35);

```
<owl:NamedIndividual rdf:about="&j.0;fetchBatchProfile">
  <rdf:type rdf:resource="&profile;Profile"/>
  <serviceRealName rdf:datatype="&xsd:string">fetchBatch</serviceRealName>
  <profile:textDescription> No Description </profile:textDescription>
  <profile:serviceName>fetchBatch</profile:serviceName>
  <profile:hasInput rdf:resource="&j.0;db"/>
  <profile:hasOutput rdf:resource="&j.0;fetchBatchReturn"/>
  <service:presentedBy rdf:resource="&j.0;fetchBatchService"/>
  <profile:hasInput rdf:resource="&j.0;format"/>
  <profile:hasInput rdf:resource="&j.0;ids"/>
  <profile:hasInput rdf:resource="&j.0;style"/>
</owl:NamedIndividual>
```

Figura 3.35: Data Property *serviceRealName* no OWL-SE

- **Data Property *propertyTypeXSD*:** a anotação semântica dos parâmetros constitui-se da associação de um parâmetro do serviço com um termo de alguma ontologia de domínio, dando significado semântico a ela. Porém, em termos de composição, a não existência do termo sintático pode ocasionar problemas, por exemplo, na comparação de serviços, se houver um parâmetro de mesmo termo semântico não importa se sua sintaxe seja textual e inteiro, tal serviço será considerado compatível. Sendo assim, foi determinado que uma propriedade contendo o conteúdo sintático fosse colocada no OWL-SE permitindo não só comparações semânticas entre serviços, mas também sintáticas. A essa propriedade deu-se o nome de *parameterTypeXSD* (Figura 3.36);

```
<owl:NamedIndividual rdf:about="&j.0;fetchBatchReturn">
  <rdf:type rdf:resource="&process;Output"/>
  <rdfs:label>fetchBatchReturn</rdfs:label>
  <process:parameterType rdf:datatype="&xsd:anyURI">
    http://gabriellacastro.com.br/SequenceAligningOntology#fasta_format
  </process:parameterType>
  <parameterTypeXSD rdf:datatype="&xsd:anyURI">
    http://www.w3.org/2001/XMLSchema#string
  </parameterTypeXSD>
</owl:NamedIndividual>
```

Figura 3.36: Data Property *parameterTypeXSD* no OWL-SE

- **Criação da Classe *SubParameter* e *Object Properties isSubParameterOf* e *hasSubParameter*:** Conforme explicitado

anteriormente, o OWL-S anota semanticamente apenas os parâmetros explícitos dos serviços web, não levando em conta os parâmetros complexos existentes. Estes parâmetros complexos são formados por outros parâmetros, que por sua vez podem ser formados por outros sub-parâmetros, e assim por diante. Sendo assim, estes parâmetros precisam ser descobertos e anotados semanticamente para que as comparações entre serviços no processo de criação de composições sejam mais precisos. Assim, foi implementado no OWL-SE uma nova classe denominada *SubParameter* (Figura 3.37), a qual herda as características da classe *Parameter* já presente no padrão OWL-S;



Figura 3.37: Classe SubParameter no OWL-SE

Porém, essa classe necessitava de relações com os parâmetros para realizar descobertas de quais dependem um do outro. Assim, as *Object Properties* *isSubParameter* e *hasParameter* (Figura 3.38) foram criadas no intuito de adicionar restrições aos parâmetros.

```

<owl:ObjectProperty rdf:about="&j.0;hasSubParameter">
  <rdf:type rdf:resource="&owl;TransitiveProperty"/>
  <rdfs:domain rdf:resource="&process;Parameter"/>
  <rdfs:range rdf:resource="&j.0;SubParameter"/>
</owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:about="&j.0;isSubParameterOf">
  <rdfs:range rdf:resource="&process;Parameter"/>
  <rdfs:domain rdf:resource="&j.0;SubParameter"/>
  <owl:inverseOf rdf:resource="&j.0;hasSubParameter"/>
</owl:ObjectProperty>

```

Figura 3.38: hasSubParameter e isSubParameterOf da OWL-SE

Mesmo com essas modificações, houve a necessidade de desenvolvimento de uma API (*Application Programming Interface*) para a obtenção de informações sobre os sub-parâmetros existentes em cada *web service*. Assim, com auxílio do *framework* Predict8 [PREDICT8, 2014] para SOA (*Service-Oriented Architecture*) foram desenvolvidas bibliotecas para e-ScienceNet responsáveis pela obtenção de dados não explícitos nos serviços WSDL, como mensagens a serem trocadas, criação de mensagens de comunicação SOA, bem com o rastreamento e mapeamento de parâmetros complexos. Dentro da *interface* de inserção de serviços (Figura 3.26), foram criados mecanismos de identificação de quando existem parâmetros

complexos em um parâmetro principal. Dentre esses mecanismos, existe a criação de uma *interface* gráfica para a anotação semântica dos sub-parâmetros (Figura 3.39) que, da mesma forma que a tela de inclusão de serviço faz, verifica se estes sub-parâmetros também são complexos e assim por diante. Essa *interface* em conjunto com a API SOA desenvolvida, também faz a verificação de obrigatoriedade de parâmetros. A anotação semântica dos parâmetros ocorre da mesma maneira que a anotação dos parâmetros principais, utilizando as *interfaces* das Figura 3.29 e Figura 3.30.

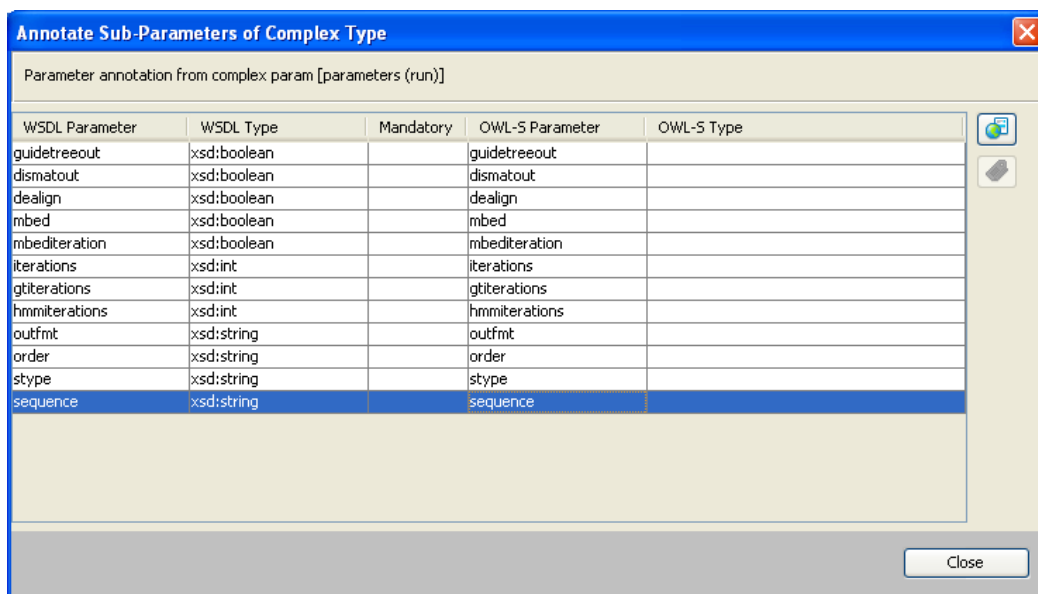


Figura 3.39: Interface de anotação semântica de sub-parâmetros

Ao final da inclusão de todos os parâmetros, complexos ou não, é realizada a geração do OWL-SE. Os sub-parâmetros são inclusos com indivíduos pertencentes à classe *SubParameter* na ontologia de serviços, de maneira semelhante aos parâmetros principais já inclusos na OWL-S, na classe *Parameter*. Porém, aqui esses parâmetros e sub-parâmetros são relacionados uns com os outros através de relacionamentos descritos pelas propriedades *hasSubParameter* e *isSubParameterOf*. Como exemplo, a Figura 3.40 (A) detalha os sub-parâmetros: *sequence*, *database*, *program*, *string* e *stype* relacionados com o parâmetro principal *parameters*, e conseqüentemente o sub-parâmetro *database* possui também um sub-parametro chamado *string*, mostrado na Figura 3.40 (B).

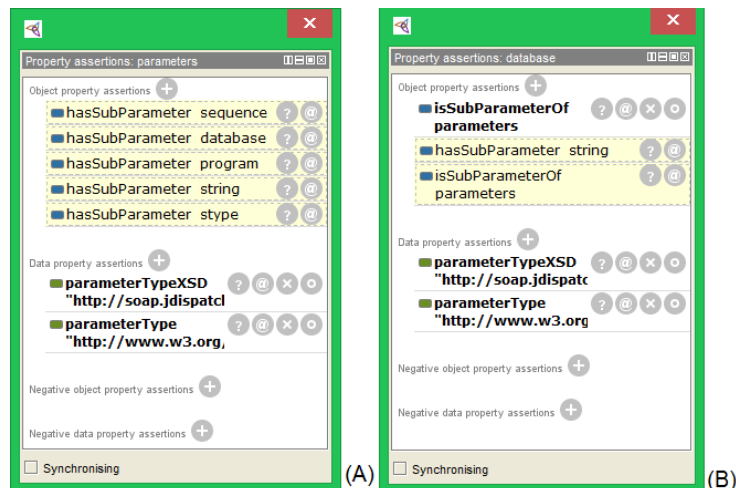


Figura 3.40: A) Sub-Parâmetros do parâmetro parameters. B) Sub-Parâmetro do sub-parâmetro database

Gerado o OWL-SE ao final da inclusão dos serviços, o mesmo passa pelo processo de validação (Figura 3.31), sendo disponibilizado no *Ontology Repository* do Gerente de Semântica e para a consulta de serviços na e-ScienceNet.

3.2.6.2 CONECTORES

Os conectores são serviços web responsáveis por converterem entradas e saídas de um serviço. Em outras palavras, podemos inferir que os conectores são serviços que modificam algum parâmetro, convertendo valores ou fornecendo novas opções de conexão. Por exemplo, considerando a composição entre dois serviços web, o primeiro tem parâmetro de saída numeral e o segundo um parâmetro de entrada textual, sendo assim, não seria possível realizar a sua conexão devido à diferença entre eles. Entretanto, cabe assinalar a existência de um serviço web que realiza a conversão entre conteúdo numeral para conteúdo textual. Outro exemplo que pode ser destacado também no contexto de modificador dos parâmetros, seria no momento de pesquisa de serviços. Nesse sentido, considerando os mesmos serviços do exemplo anterior, sua pesquisa não traria qualquer conexão se usado somente o seu parâmetro de saída com o uso do conector o serviço seguinte seria encontrado possibilitando a composição.

Assim, conectores no contexto do Gerente de Serviço são considerados serviços web especiais que servem para o auxílio de outros serviços, sendo também armazenados na e-ScienceNet. Para isso, da mesma forma que os serviços, os conectores contam com uma *interface* de inclusão (Figura 3.41), podendo os mesmos ser inclusos com disponibilização somente para um tipo de serviço previamente cadastrado ou para qualquer tipo de serviço (Figura 3.42). Desse modo, a anotação semântica dos conectores é realizada da mesma maneira que um serviço, o qual faz a busca por sub-parâmetros dos tipos complexos e gerando ao final um OWL-SE. As informações sobre os conectores também são persistidas na *PeerOntology* do *peer*, porém, diferente de um serviço que é armazenado na classe *WebService*, estes são armazenados na classe *Connector*.

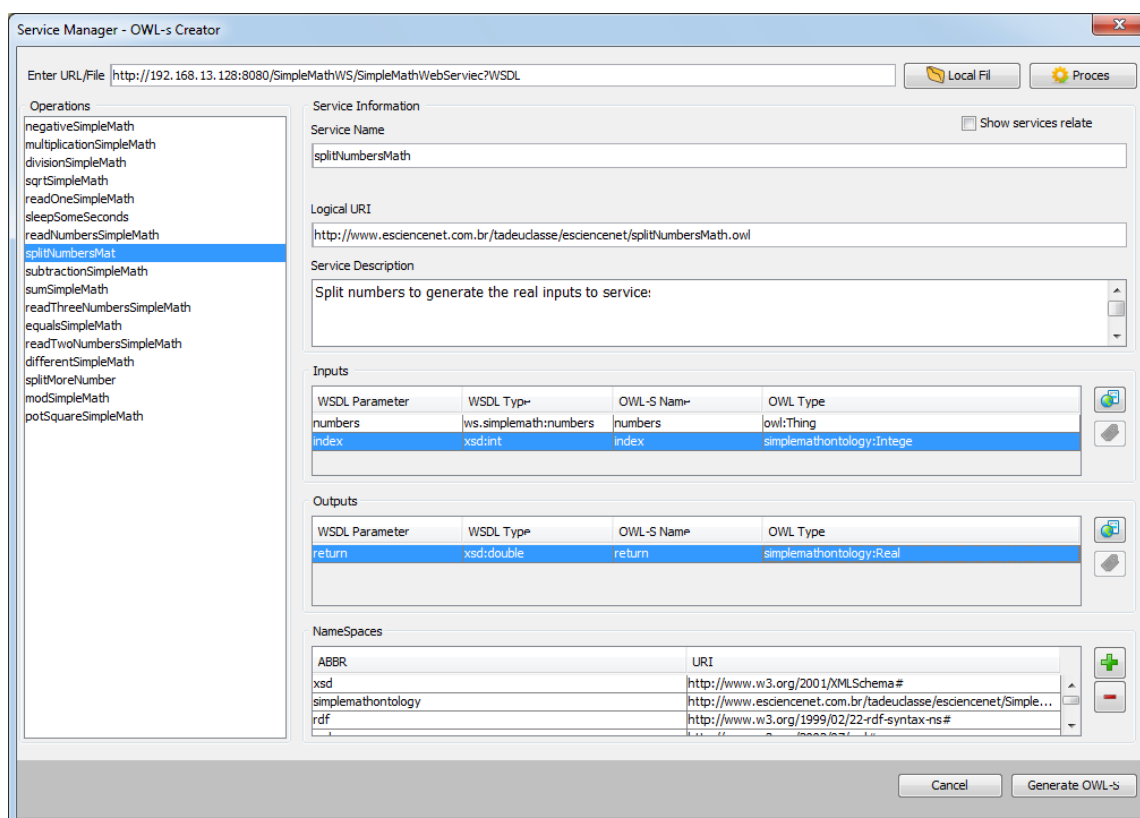


Figura 3.41: Inclusão de conector na e-ScienceNet

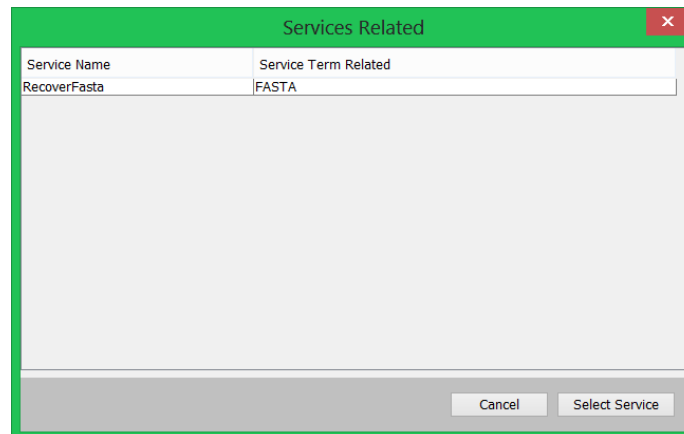


Figura 3.42: Associação de conector com um serviço web

3.2.7 GERENTE DE PESQUISA (SEARCH MANAGER)

O Gerente de Pesquisa (Figura 3.43) é o responsável pela pesquisa por recursos na e-ScienceNet, baseando-se nos grupos de interesses existentes.

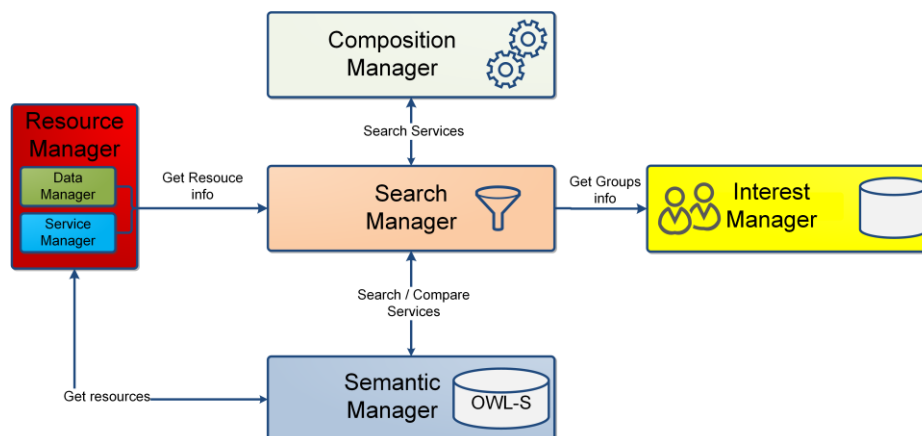


Figura 3.43: Modelo de comunicação do Gerente de Pesquisa

Devido à diversidade de recursos que podem ser partilhados na rede, este gerente apresenta dois tipos de métodos de pesquisa, sendo eles: pesquisa de arquivos e pesquisa de serviços.

- **Pesquisa de Arquivos:** os arquivos são compartilhados na e-ScienceNet pelo Gerente de Dados e o mesmo armazena informações sobre estes arquivos na *PeerOntology*. Com este armazenamento, há a possibilidade de busca semântica pelos dados através da pesquisa

SPARQL em cada *peer* de um grupo semântico. A e-ScienceNet por meio de sua tela principal implementa meios de pesquisa de arquivos fornecendo opções para buscas locais, buscas remotas ou ambas, além de ser possível a seleção dos campos de pesquisa como: nome, extensão, descrição, data e tamanho do arquivo. Tal mecanismo é de simples utilização bastando o cientista informar no campo textual alguma palavra que deseja buscar dentro do seu grupo de interesse e selecionar as opções de busca. Após isso, o Gerente de Pesquisa envia um *broadcast* de pesquisa para todos os *peers* conectados à comunidade semântica. Os *peers* então recebem os termos e fazem as buscas em sua própria *PeerOntology*, acionando o método de pesquisa de arquivos no Gerente de Semântica (Figura 3.44). Com os resultados de cada busca, os *peers* retornam os resultados para quem requisitou e o Gerente de Pesquisa deste *peer*, exibe os resultados em uma árvore de dados na tela principal (Figura 3.45), permitindo que os usuários possam requisitar o *download* dos recursos por meio da própria e-ScienceNet.

```

PREFIX t: <http://www.esciencenet.com.br/tadeuclasse/esciencenet/PeerOntology.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?name ?extension ?dataCriacao ?tamanho ?description ?tipo ?caminho
where {?group t:pertenceOneGroup t:Bioinformatics.
      ?group rdf:type t:Files.
      ?group t:fileName ?name.
      ?group t:fileExtension ?extension.
      ?group t:fileCreationDate ?dataCriacao.
      ?group t:fileDescription ?description.
      ?group t:fileSize ?tamanho.
      ?group t:fileType ?tipo.
      ?group t:filePath ?caminho.
      ?group t:isFileOf t:Tadeu_PEER.
      FILTER(
        regex(str(?name), "workflow", "i") ||
        regex(str(?description), "workflow", "i")
      )
}

```

Figura 3.44: Query SPARQL para consultas de arquivos na PeerOntology

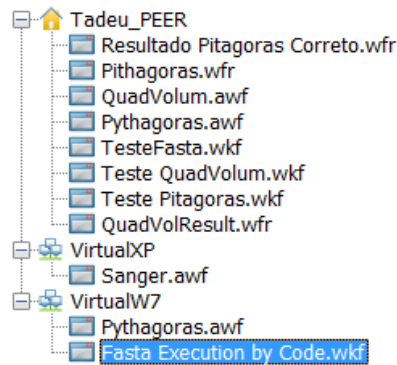


Figura 3.45: Resultado de uma busca de arquivos pelo Gerente de Pesquisa

- **Pesquisa de Serviços:** a pesquisa de serviços é considerada parte importante da composição de serviços realizada pela e-ScienceNet, trabalhando em conjunto com os Gerentes de Serviço, Semântica e Composição. Este tipo de pesquisa se baseia nos termos relacionados com algum tipo de serviço e suas variações obtidas através de inferências em ontologias de domínio. Essa pesquisa conta com alguns passos importantes os quais dependem do Gerente de Composição para a realização da compatibilização entre parâmetros dos serviços web, para que seja possível a composição. Cabe ressaltar que esses passos serão detalhados na seção 3.2.8. No entanto, é apresentado aqui o detalhamento de parte do processo, que é pesquisa de serviços.

Ao determinar uma tarefa a ser pesquisada, o Gerente de Semântica precisa calcular e inferir as possíveis variações existentes para um termo. Como exemplo, é apresentada a Figura 3.46. Originalmente, o serviço a ser pesquisado seria algum serviço web que fosse anotado semanticamente pelo termo *Blast* na ontologia de domínio. Porém, foi verificado utilizando máquinas de inferência, que existiam equivalências relacionadas com tal termo e, portanto, o critério de pesquisa deve ser expandido, incluindo não só o termo *Blast*, mas todos os termos retornados pela inferência na ontologia (veja o algoritmo no apêndice B).



Figura 3.46: Variações dos termos de pesquisa através da inferência na ontologia de domínio

Da mesma maneira que na pesquisa de arquivos, os termos gerados são enviados por meio de *broadcast* aos *peers* conectados aquela comunidade semântica a fim de realizar a busca por serviços. Os *peers* ao receberem os termos de pesquisa buscam os serviços que são compatíveis em sua *PeerOntology* e os coloca em uma lista. Essa lista, então, passa para a etapa de busca de informações sobre os serviços nas ontologias OWL-SE (veja o algoritmo no apêndice C), as quais são carregadas pelo Gerente de Semântica no intuito de recuperar as suas informações como: parâmetros de entrada e saída, descrições, nome real do serviço, dentre outras.

Com as informações sobre os serviços encontrados, o Gerente de Pesquisa retorna os resultados localizados para o *peer*, o qual os exibe em uma hierarquia (Figura 3.47) para que possam ser utilizados na criação da composição.

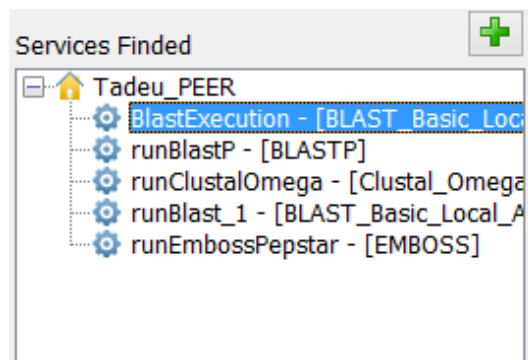


Figura 3.47: Serviços retornados pela busca

O processo de pesquisa de conectores para um serviço também se baseia no mesmo princípio de pesquisa de serviços científicos. A única diferença é que não existe uma lista de termos a ser pesquisado, somente o serviço para o qual se deseja utilizar o conector, empregando seus parâmetros como comparativos para encontrar resultados. De tal modo, a comparação

entre os parâmetros tanto de serviços, quanto de conectores, é um processo realizado em complemento à pesquisa e será detalhado na próxima seção.

3.2.8 GERENTE DE COMPOSIÇÃO (COMPOSITION MANAGER)

O desenvolvimento de uma arquitetura capaz de realizar a criação e compartilhamento de experimentos científicos é um dos focos da especificação da e-ScienceNet e para isso, foi desenvolvido o Gerente de Composição. O Gerente de Composição utiliza as funcionalidades do Gerente de Pesquisa para a busca dos serviços; do Gerente de Semântica para o processamento de inferências, anotações semânticas, geração de composições abstratas e compatibilização de parâmetros entre conectores e/ou serviços web; do Gerente de Dados para o armazenamento das várias possíveis etapas do processo; e do Gerente de Ferramentas na apresentação dos resultados obtidos com a execução.

Neste contexto, o processo de criação de uma composição de serviços na e-ScienceNet passa por algumas fases distintas, sendo elas: desenvolvimento do modelo abstrato de composição; busca de serviços web e conectores, conexão entre serviços; geração da composição; e execução. A Figura 3.48 mostra o esquema do Gerente de Composição e suas etapas.

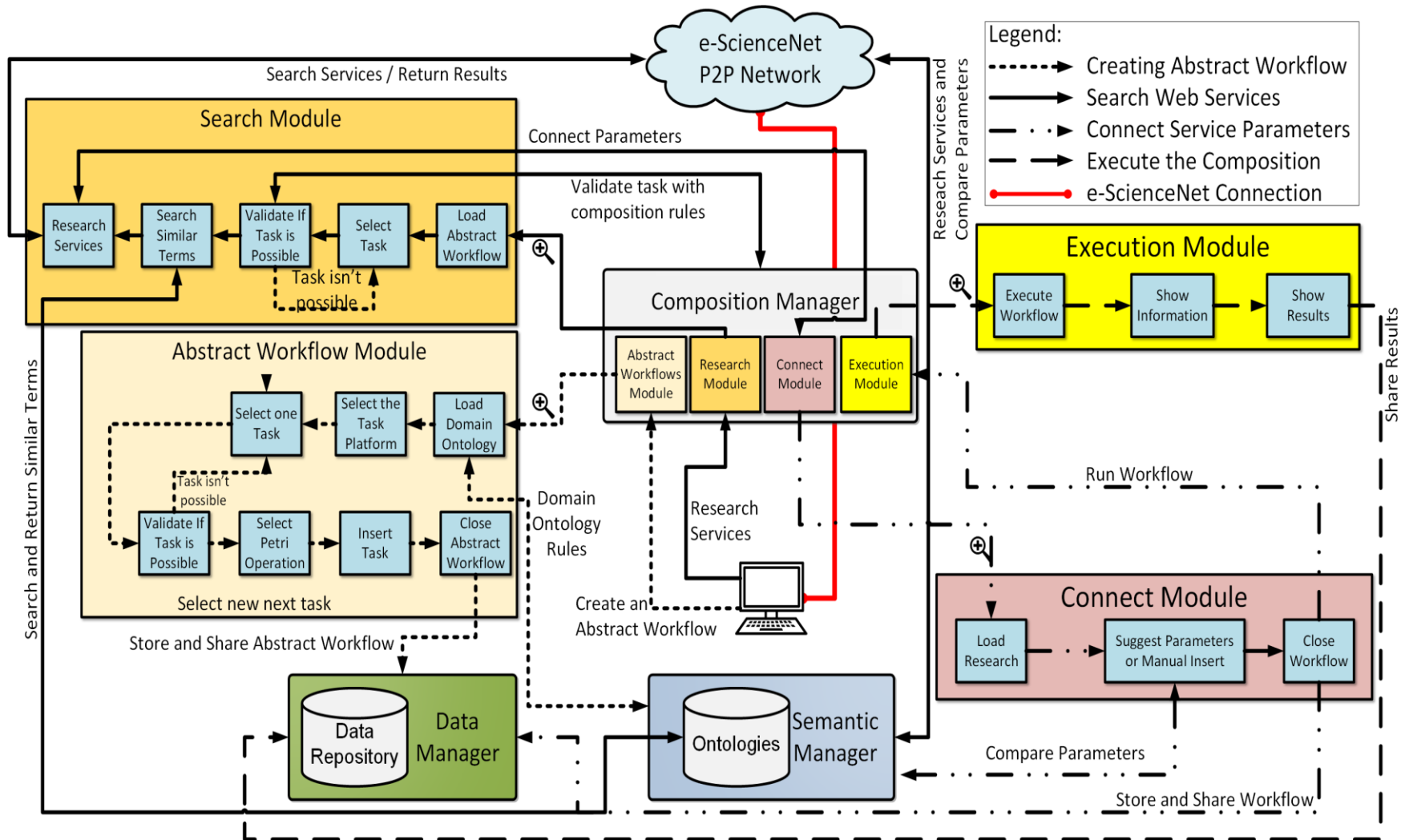


Figura 3.48: Fases de execução do Gerente de Composição

1. **Criação do *workflow* abstrato:** este tipo de representação foi baseada na proposta apresentada no trabalho de COSTA et al. [2013], que especifica um modelo lógico de tarefas baseadas em um domínio exclusivo de conhecimento. Tal modelo não compreende informações de quais são os serviços/algoritmos/aplicações científicas a serem conectadas, mas sim quais os tipos de tarefas devem ser compostas para se especificar um dado experimento. No contexto da e-ScienceNet, estas tarefas podem ser descritas através de termos de uma ontologia de domínio. Assim, para que sejam criados tais modelos, o Gerente de Composição fornece uma *interface* (Figura 3. 49) específica. Os termos utilizados na descrição de tarefas são obtidos por meio das ontologias de domínio e obedecem a regras e restrições existentes. Diante de tais configurações, a sequência das tarefas é obtida através das próprias restrições existentes na ontologia de domínio. E mais, cada inserção na sequência de tarefas é acompanhada por uma conexão lógica entre elas (*SEQUENCE, OR-SPLIT, AND-SPLIT, OR-JOIN, AND-JOIN*) as quais são responsáveis por determinar o fluxo de execução da composição. Ao concluir tal tarefa, o usuário deve salvar o modelo abstrato em um arquivo (em formato *.awf). Automaticamente o arquivo do modelo da composição é disponibilizado como recurso compartilhado na e-ScienceNet, podendo ser pesquisado por outros nós da comunidade semântica. No capítulo 4, serão apresentados alguns modelos reais de composição.

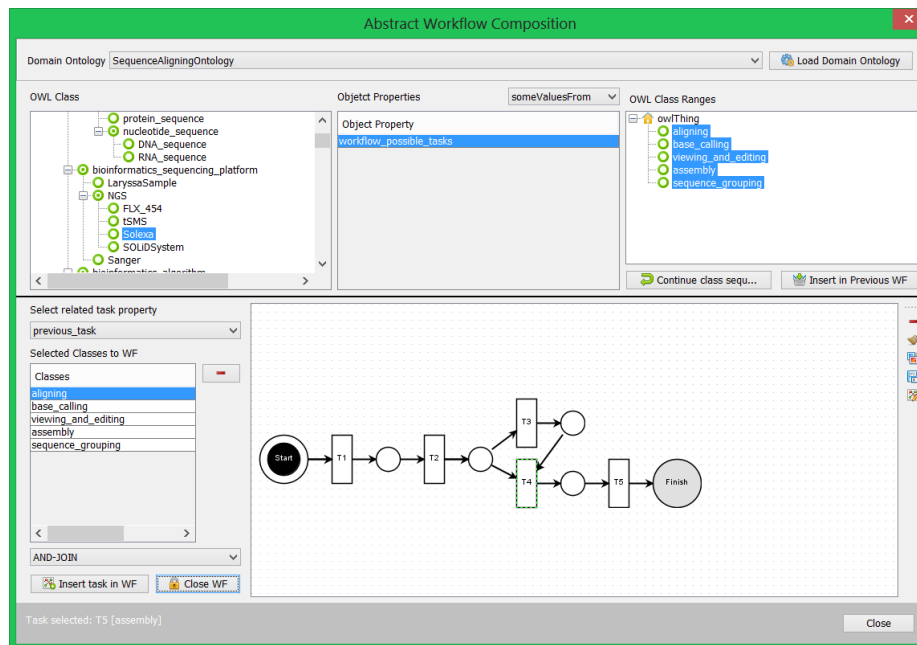


Figura 3. 49: Criação do modelo de workflow abstrato

2. **Busca de Serviços Web:** existem dois métodos de busca de serviços na e-ScienceNet para a composição de serviços, a saber, pesquisa por composição abstrata e pesquisa livre. Considerando o contexto da composição, o foco é na primeira opção. A partir do modelo abstrato criado na fase 1, o usuário pode carregá-lo na *interface* de pesquisa de serviços, e este servirá de guia para as buscas. Neste contexto, o cientista deve selecionar sequencialmente as tarefas no modelo. Fazendo isso, o sistema irá pesquisar os serviços relacionados a cada tarefa. Como visto na seção 3.3.7 o Gerente de Semântica, por meio do Gerente de Pesquisa, deve gerar variações dos termos relacionados e acionar requisições de pesquisas a todos os *peers* da comunidade semântica. Caso seja um serviço inicial - ou seja, o primeiro serviço do modelo - não serão realizadas comparações para a compatibilidade dos parâmetros do serviço. Caso o serviço não seja o inicial, após a inclusão deste serviço na composição, é necessário que sejam realizadas verificações de compatibilidade entre os parâmetros do serviço existente, com os encontrados na pesquisa para que os mesmos sejam incluídos como resultados válidos. Para ampliar a qualidade dos serviços web retornados por uma pesquisa, as comparações não são feitas somente baseadas na compatibilidade semântica de um

parâmetro, mas também na sua compatibilidade sintática. Com as informações obtidas, i) na obtenção dos serviços web compatíveis com os termos inferidos e também ii) na busca pelas informações de parâmetros dentro das ontologias OWL-SE, é possível verificar, em primeiro lugar a compatibilidade semântica dos parâmetros e logo em seguida, se a sua sintaxe é também compatível. Desse modo, um serviço só é considerado um resultado válido caso o mesmo tenha compatibilidade semântica e sintática em pelo menos um de seus parâmetros. Com o retorno dos resultados encontrados pela pesquisa, o serviço pode ser incluso na composição, permitindo ao usuário escolher a próxima tarefa a ser pesquisada.

2.1. Serviço com conexão lógica JOIN: como conexão lógica nas composições muitas vezes existe a tarefa *join*, ou seja, a junção de mais de um serviço em outro. Porém, na e-ScienceNet existe a necessidade de informar quais são os serviços que se conectam com outro por meio deste tipo de conexão. Assim sendo, uma *interface* (Figura 3. 50) é disponibilizada exibindo as opções dos serviços que aceitam esse tipo de conexão lógica, bastando somente o usuário associá-las.

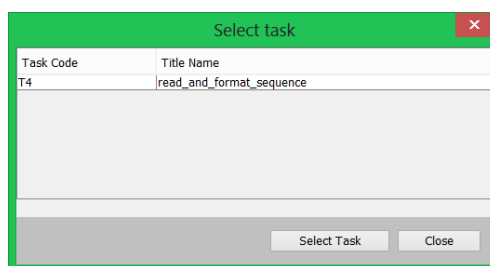


Figura 3. 50: Interface de associação da conexão lógica JOIN

2.2. Busca por conectores: como dito na seção 3.2.6.2, conectores são serviços web que podem alterar a configuração dos parâmetros de um serviço dentro de uma composição. O processo de pesquisa de conectores é parecido com a busca de serviços para a composição, sendo, portanto similar à busca de serviços, considerando também a compatibilidade semântica e sintática de seus parâmetros. Na e-

ScienceNet, um serviço web pode usar um conector de duas formas diferentes: conectores de entrada, que são conectores que alteram o comportamento de parâmetros de entrada em um serviço web; ou conectores de saída, que alteram parâmetros de saída de um serviço. Por exemplo, é suposto que um serviço sem conectores possua três parâmetros de entrada, e um conector apresente dois parâmetros de entrada, ao realizar a conexão de algum dos três parâmetros do serviço com a saída deste conector, o serviço passa a contar com quatro parâmetros de entrada, pois são os três parâmetros de entrada do serviço, acrescido dos dois do conector, subtraindo uma entrada do serviço, pois a mesma se conecta ao parâmetro de saída do conector. A Figura 3.51, apresenta essa configuração. Como ocorrido na busca de serviços, para a busca de conectores, uma requisição é enviada para os demais *peers* e estes retornam os conectores compatíveis, os exibindo para a seleção do cientista (Figura 3.52). Após a seleção do conector, o cientista imediatamente deve selecionar qual dos parâmetros do serviço o conector deve se associar conforme é exposto na Figura 3.53.

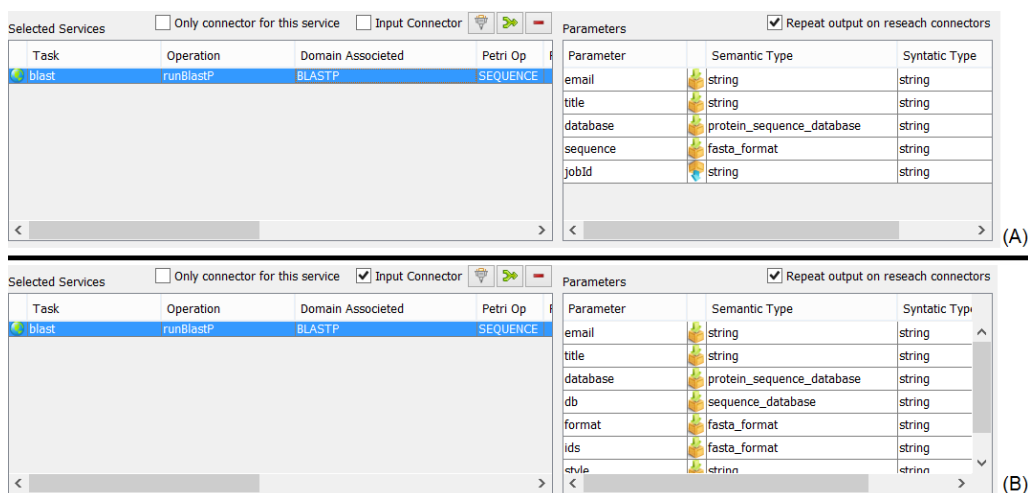


Figura 3.51: Diferenças entre os parâmetros de um serviço antes e depois de se unir a um conector

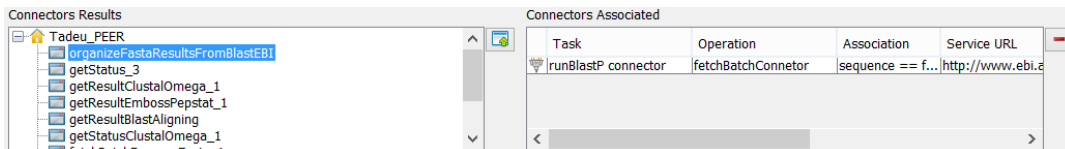


Figura 3.52: Conectores na e-ScienceNet

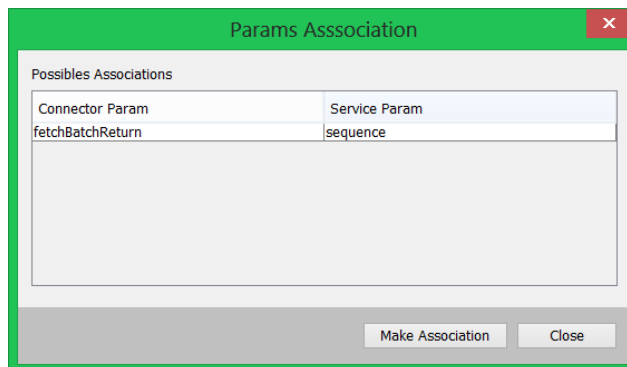


Figura 3.53: Associação de parâmetro do serviço com o conector

3. **Conexão entre serviços:** após a fase de pesquisa dos serviços, a e-ScienceNet fornece uma *interface* para que o usuário consiga fazer os ajustes entre as conexões dos parâmetros dos serviços, como pode ser visto na Figura 3.54. Essa *interface* é capaz de mostrar tanto os parâmetros de entrada como os de saída dos serviços, e também, permite que os usuários alterem parâmetros de conectores. Esse mecanismo de associação permite que o usuário preencha os parâmetros da composição com dois tipos de conexão, chamada de conexão manual, que consiste na inserção manual de valores nos parâmetros do serviço web; e a conexão do parâmetro, que consiste em ligar a saída de um serviço com a entrada de outro e vice-versa. A Figura 3.55 mostra estes dois tipos de inserção de parâmetros, sendo que a opção de conexão é mostrada de acordo com a compatibilidade do parâmetro com o serviço seguinte. Ao final deste processo, o usuário pode salvar o que é chamado de composição funcional (arquivos *.wkf), o que consiste em um arquivo com as configurações realizadas para futuras alterações ou execuções. Este arquivo é automaticamente compartilhado com os outros *peers* da comunidade semântica através do Gerente de Dados. Além da criação desta composição funcional, neste momento o cientista já possui uma composição que pode ser executada pela e-ScienceNet.

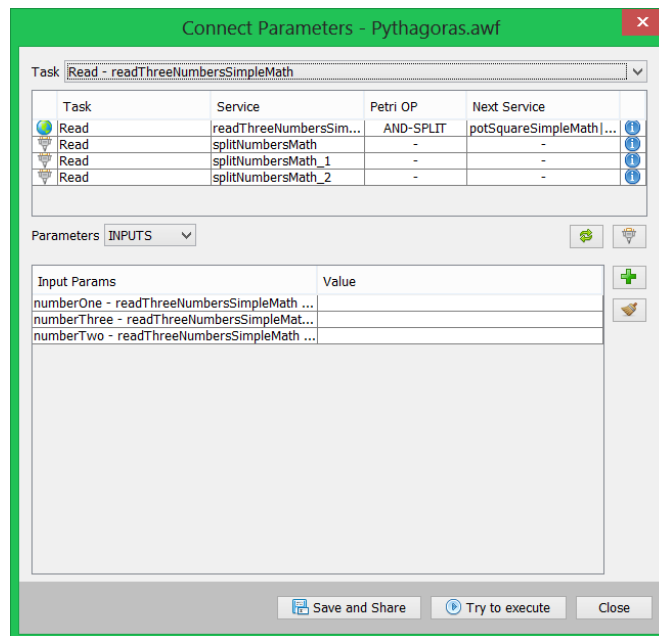


Figura 3.54: Conexão de parâmetros entre serviços e inclusão de valores

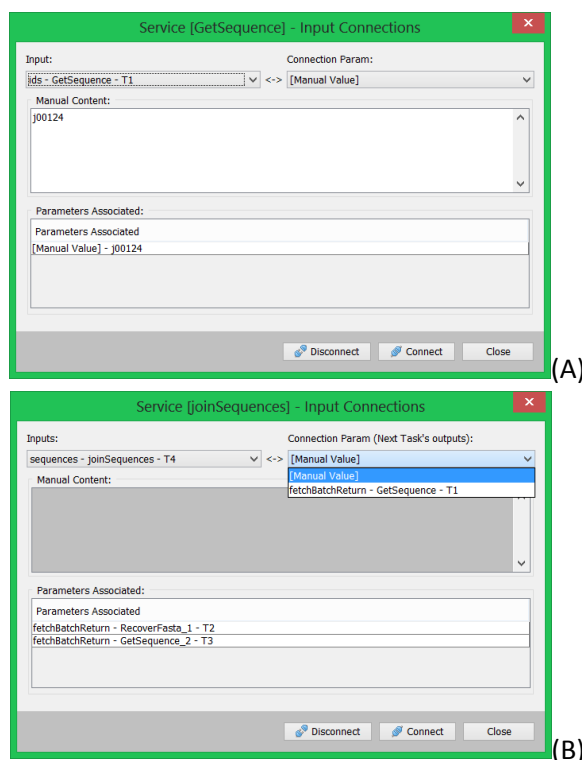


Figura 3.55: (A) Inclusão de valores manuais em um parâmetro. (B) Conexão de múltiplas saídas em uma entrada

3.1. Restrições em serviços: Muitas vezes na composição de aplicações científicas é necessário que sejam adicionadas restrições como

validações lógicas e estruturas de repetição. A e-ScienceNet permite a criação de tais estruturas no momento de conexão entre os parâmetros, bastando ao usuário somente selecionar qual o serviço que terá a restrição e adicionar tal funcionalidade. A título de exemplificação, a Figura 3.56 mostra a tela de configuração de uma restrição para um serviço web. Nesse sentido, o usuário pode configurar tempo para a próxima chamada ao serviço e a condicional para o término de sua execução junto com o parâmetro a ser verificado.

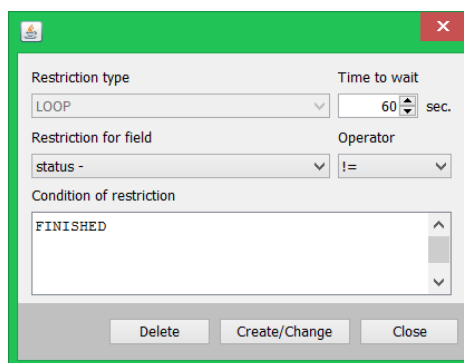


Figura 3.56: Interface de criação de restrições para parâmetros

4. **Execução de *Workflows*:** essa é a fase que compreende a execução final da composição de serviços. A partir do modelo de composição funcional, a e-ScienceNet fornece uma *interface* visual (mostrada na Figura 3.57) onde o cientista pode executar seu novo experimento. Essa funcionalidade conta com o auxílio do Gerente de Serviço por meio da API SOA para criação de mensagens de SOAP (veja nos apêndices D e E exemplos das mensagens de requisição e resposta dos serviços web) e execução do serviço web (Figura 3.58), onde todos os serviços são inspecionados e executados. Levando em conta as configurações de conectores e restrições, um serviço só é considerado com execução completa após ser executado por completo, ou seja, após todos os seus conectores terminarem sua execução. Além disso, a etapa de execução conta com mecanismos úteis como campos onde o usuário pode acompanhar a execução passo a passo, além de um campo onde ele também pode verificar os valores de cada um dos parâmetros de um serviço selecionado. Caso a composição tenha

problemas em sua execução, mensagens são exibidas para o usuário. O resultado final de tal execução é exibido para o usuário.

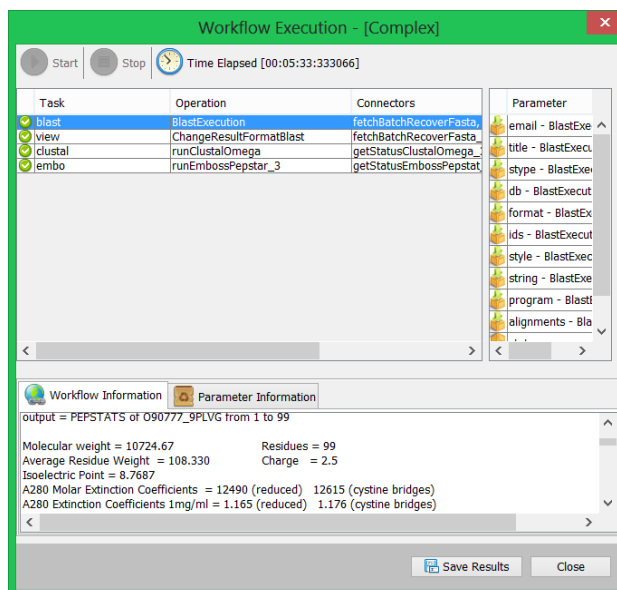


Figura 3.57: Execução do Workflow Científico

```
private boolean processOperation(OWLSOperation operation){
    try{
        ProcessWSDLInformation processWSDLInformation = new ProcessWSDLInformation();

        WSDLRecover wsdlRecover = processWSDLInformation.getWSDLInfo(operation.getServiceURL(),
                                                                    operation.getOperationRealName(),
                                                                    null, null);

        if(wsdlRecover != null){
            String envelop = processWSDLInformation.generateEnvelopeSOAP(wsdlRecover,
                                                                        operation.getServiceURL(),
                                                                        operation.getOperationRealName());

            if(!envelop.equals("")){
                envelop = this.replaceInputParameters(envelop, operation.getInputs());

                SOAPMessage soapMessage = processWSDLInformation.createSOAPMessage(envelop);

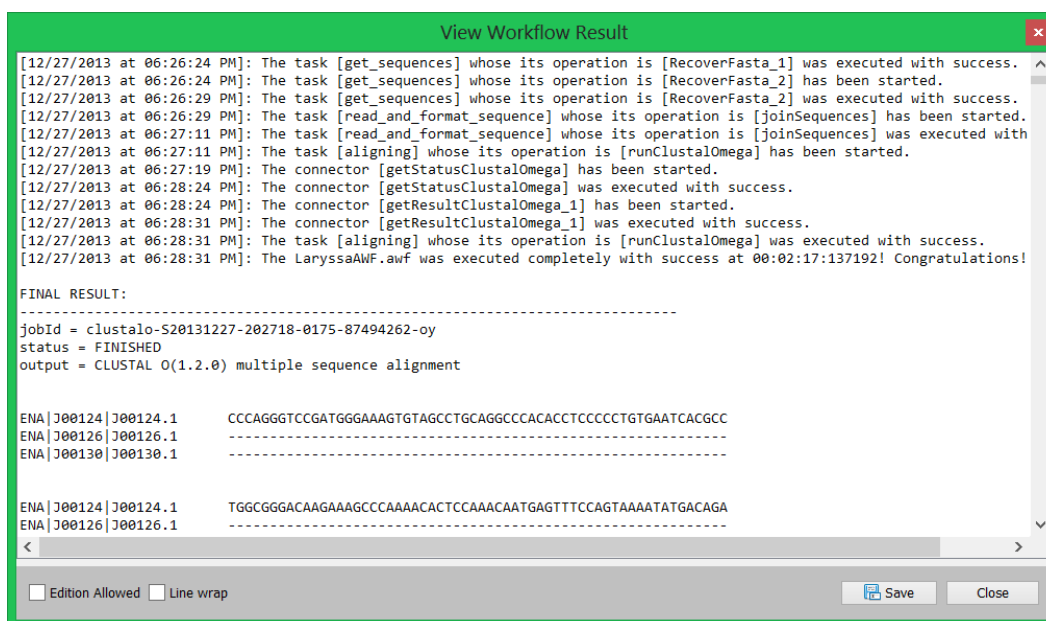
                WSDLRecoverService wsdlRecoverService = wsdlRecover.getServices().get(wsdlRecover.getServices().size() - 1);
                if(wsdlRecoverService != null){
                    WSDLRecoverPort wsdlRecoverPort = wsdlRecoverService.getPorts().get(wsdlRecoverService.getPorts().size() - 1);

                    if(wsdlRecoverPort != null){
                        if(!processWSDLInformation.soapRequest(wsdlRecoverPort.getAddress(), soapMessage, operation)){
                            this.errors = processWSDLInformation.getError();
                            return false;
                        }
                    }
                }
            }else{
                this.errors = processWSDLInformation.getError();
                return false;
            }
        }else{
            throw new Exception("Errors are happening in the moment to execute the service " + operation.getOperationName());
        }

        return true;
    }catch(Exception e){
        this.errors = e.getMessage();
        return false;
    }
}
```

Figura 3.58: Código para a execução de um web service

5. **Exibição dos Resultados:** após a execução correta de uma composição, os resultados são exibidos pelo Gerente de Composição, que possibilita também que os resultados sejam armazenados e compartilhados por meio do Gerente de Dados em um arquivos de resultados com a extensão *.wfr. Estes arquivos podem ser visualizados por meio de uma *interface* que permite também a edição desta composição. A Figura 3.59 implementa essa *interface*.



```

View Workflow Result
[12/27/2013 at 06:26:24 PM]: The task [get_sequences] whose its operation is [RecoverFasta_1] was executed with success.
[12/27/2013 at 06:26:24 PM]: The task [get_sequences] whose its operation is [RecoverFasta_2] has been started.
[12/27/2013 at 06:26:29 PM]: The task [get_sequences] whose its operation is [RecoverFasta_2] was executed with success.
[12/27/2013 at 06:26:29 PM]: The task [read_and_format_sequence] whose its operation is [joinSequences] has been started.
[12/27/2013 at 06:27:11 PM]: The task [aligning] whose its operation is [runClustalOmega] was executed with success.
[12/27/2013 at 06:27:19 PM]: The connector [getStatusClustalOmega] has been started.
[12/27/2013 at 06:28:24 PM]: The connector [getStatusClustalOmega] was executed with success.
[12/27/2013 at 06:28:24 PM]: The connector [getResultClustalOmega_1] has been started.
[12/27/2013 at 06:28:31 PM]: The connector [getResultClustalOmega_1] was executed with success.
[12/27/2013 at 06:28:31 PM]: The task [aligning] whose its operation is [runClustalOmega] was executed with success.
[12/27/2013 at 06:28:31 PM]: The LaryssaAWF.awf was executed completely with success at 00:02:17:137192! Congratulations!

FINAL RESULT:
-----
jobId = clustalo-S20131227-202718-0175-87494262-oy
status = FINISHED
output = CLUSTAL O(1.2.0) multiple sequence alignment

ENA|J00124|J00124.1      CCCAGGGTCCGATGGGAAAGTGTAGCCTGCAGGCCACACACCTCCCCCTGTGAATCACGCC
ENA|J00126|J00126.1      -----
ENA|J00130|J00130.1      -----

ENA|J00124|J00124.1      TGGCGGGACAAGAAAGCCCAAAACACTCCAACAATGAGTTCCAGTAAAAATATGACAGA
ENA|J00126|J00126.1      -----

```

Edition Allowed Line wrap

Figura 3.59: Visualização de resultados obtidos na execução do workflow

6. **Execução de Web Service:** essa não é uma fase em si da composição de serviços pela e-ScienceNet, porém, por se tratar de uma função conjunta entre os Gerentes de Dados, são detalhados aqui: Serviços e Composição. Essa funcionalidade permite ao cientista realizar a execução de um serviço web isolado. Para usá-lo basta o usuário acessar a tela de execução (Figura 3.60), selecionar o serviço na lista, preencher seus parâmetros e executá-lo. Dessa forma, os resultados podem ser exibidos da mesma maneira que descrito na Fase 5 da composição de serviços. Tal funcionalidade pode ser útil caso um pesquisador decida obter apenas o resultado de um serviço web sem que necessite criar uma nova composição para isso.

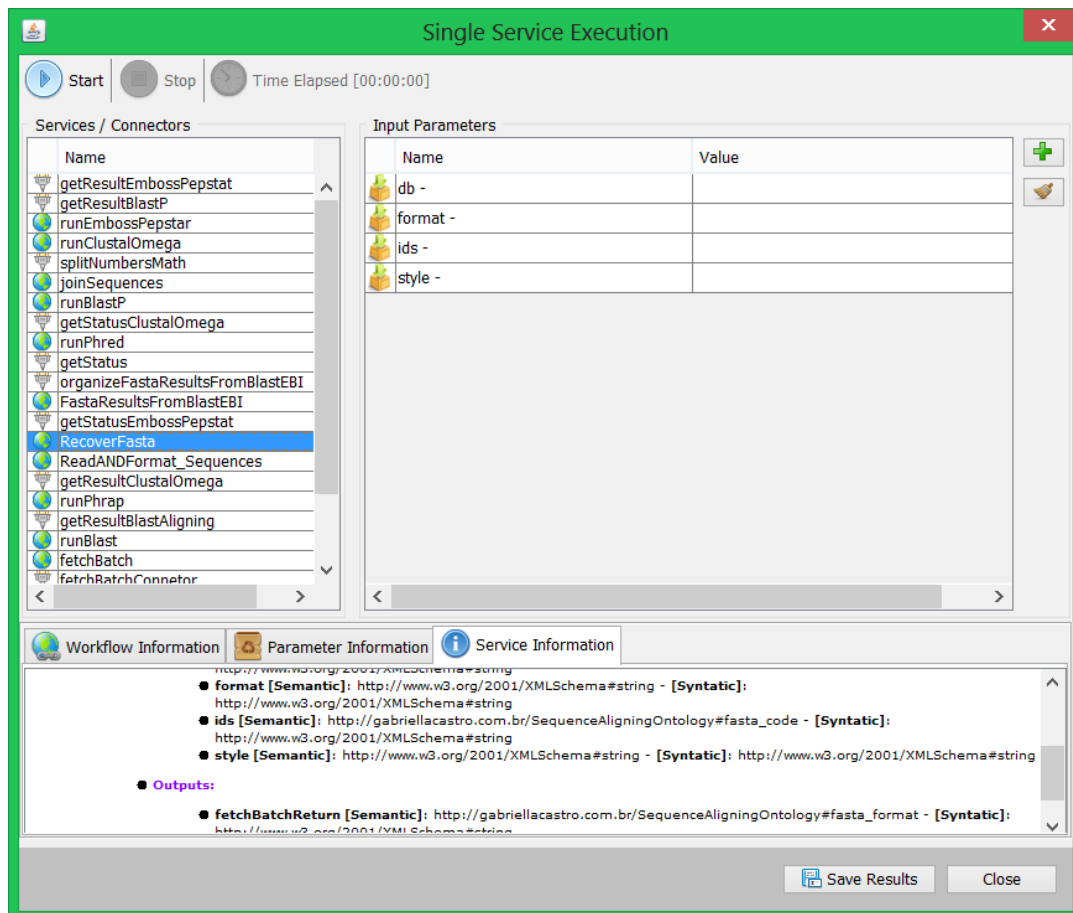


Figura 3.60: Execução de serviço único

3.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

A e-ScienceNet é um protótipo de rede ponto a ponto semântica com o intuito de realizar composições de serviços entre comunidades semânticas e pesquisadores que estejam dispersos geograficamente.

Este capítulo mostrou uma visão funcional da arquitetura, baseada na comunicação entre os gerentes do sistema. Dentre as principais funcionalidades podem ser destacadas i) a separação dos *peers* entre grupos semânticos de interesse, onde essas comunidades possuem ontologias responsáveis por fornecer suporte a diferentes subsistemas, como o auxílio na geração de *workflows* abstratos; ii) a realização de anotações semânticas de serviços web e a geração da extensão do padrão OWL-S; iii) a inferência de variações dos termos nas ontologias de domínios para determinada tarefa; iv) a

busca de serviços web semânticos para a composição dos *workflows*; v) o uso de conectores para ampliar as possibilidades de pesquisa; e vi) a execução das composições de serviços.

O próximo capítulo irá abordar as funcionalidades da e-ScienceNet no processo de composição de experimentos científicos, sendo essas funcionalidades demonstradas por meio de provas de conceito. Essas provas de conceito visam demonstrar a viabilidade da arquitetura proposta no processo de criação de experimentos científicos em ambientes distribuídos.

4 PROVA DE CONCEITO: USO DA E-SCIENCENET EM CENÁRIOS CIENTÍFICOS

Com o intuito de verificar a viabilidade de utilização da e-ScienceNet, três provas de conceito (*Proof of Concept - PoC*) foram executadas. O objetivo desta prova de conceito foi definido de acordo com o “G”oal da abordagem *Goal/Question/Metric* (GQM) [BASILI et al., 1994]. Os objetivos, segundo a abordagem GQM, devem ser formulados conforme o *template* a seguir:

“Analisar o <objeto de estudo> com a finalidade de <objetivo> com respeito à <foco da qualidade> do ponto de vista de <perspectiva> no contexto de <contexto>”.

Com base no *template* para objetivos do GQM, o objetivo destas provas de conceito é: **“Analisar a arquitetura e-ScienceNet com a finalidade de verificar a viabilidade de seu uso em rede ponto a ponto, a partir da visão do cientista no contexto de composição de aplicações científicas”.**

Para isso, este capítulo está dividido nas seguintes seções: seção 4.1, apresenta uma introdução às provas de conceito, abordando um contexto geral de configuração da rede de apoio em que foram especificados os *peers* da e-ScienceNet. Na seção 4.2, é exposta a primeira prova de conceito, abordando como tema principal uma composição simples para execução do teorema matemático de Pitágoras. Na seção 4.3, é apresentada a implementação e execução de uma composição apresentada originalmente em DA SILVA, BRAGA e CAMPOS [2012], detalhando desde a criação do modelo abstrato até a geração dos resultados da execução. A seção 4.4, especifica uma composição, com base em um modelo desenvolvido pelo grupo Núcleo de Engenharia do Conhecimento [NENC, 2014] no contexto de um projeto CNPq (Processo número 559998/2010-4). Na seção 4.5, a prova de conceito detalha o desenvolvimento do modelo abstrato e busca por serviços web relacionados ao modelo desenvolvido no trabalho de COSTA et al. [2013]. Na seção 4.6, é apresentada a análise da viabilidade de uso da e-ScienceNet. E finalmente, a seção 4.7 assinala as conclusões do capítulo.

4.1 CONTEXTO GERAL

Para apoiar tal perspectiva, foi implementado um protótipo baseado na tecnologia de criação de redes P2P chamado JXTA [WILSON, 2002]. Para que as provas de conceito pudessem ser executadas, um ambiente de rede foi desenvolvido a fim de que os nós da e-ScienceNet pudessem utilizar todos os recursos detalhados no capítulo 3.

Para tanto, esse ambiente desenvolvido se formou através de quatro configurações. Duas máquinas reais conectadas por meio de uma rede wi-fi. E em uma dessas máquinas foram criadas duas máquinas virtuais distintas ligadas por meio de uma rede virtual com conexão externa onde foi utilizado o software VMWare Workstation [VMWARE, 2014]. As configurações destas máquinas estão detalhadas abaixo:

- Máquina Real 1: a máquina real 1 recebeu o nome de *XPPeer* na e-ScienceNet e possui a seguinte configuração: sistema operacional Microsoft Windows XP *SP2*, processador Intel Pentium 4 2.0 GHz, 1 GB de memória RAM, 40 GB de disco rígido e conexão com a rede de 10 Mbps;
- Máquina Real 2: a máquina real 2 recebeu o nome de *Tadeu_Peer*, contemplando também as duas máquinas virtuais usadas no ambiente de rede. As suas configurações são: sistema operacional Microsoft Windows 8 *Pro*, processador Intel i7 Core 3.2 GHz, 8 GB de memória RAM, 1 TB de disco rígido e conexão com a rede de 10 Mbps;
- Máquina Virtual 1: a máquina virtual 1 recebeu o nome de *VIRTUALXP* e possui a seguinte configuração: sistema operacional Microsoft Windows XP *Professional*, processador de 4 núcleos, 2 GB de memória RAM, disco rígido de 10 GB, e conexão com a rede de 10 Mbps;
- Máquina Virtual 2: a máquina virtual 2 recebeu o nome de *VIRTUALW7* e possui as configurações: sistema operacional Microsoft Windows 7 *Ultimate*, processador de 4 núcleos, 1 GB de memória RAM e conexão com a rede de 10 Mbps.

Em cada um desses computadores foram criados *peers* da e-ScienceNet e eles foram divididos em duas comunidades semânticas chamadas *Mathematics* e *Bioinformatics*, cada uma dessas têm como super nó o *Tadeu_Peer* por ser uma máquina real com o maior poder de processamento entre os computadores do ambiente. De tal modo, na Figura 4.1 é possível ver o esquema de divisão semântica dos *peers*, onde *Tadeu_Peer* faz parte das duas comunidades semânticas assim como o *VIRTUALXP*, porém o *peer VIRTUALW7* só utiliza a comunidade semântica *Bioinformatics* e o *XPPeer* somente *Mathematics*.

A criação de tais grupos se deu com a conexão da máquina *Tadeu_Peer* na e-ScienceNet. Vale ressaltar que neste momento, ainda não existiam comunidades semânticas para o *peer* se conectar. Assim, foram criados, no *peer Tadeu_Peer*, os grupos semânticos *Mathematics* e *Bioinformatics*, fazendo o upload das ontologias de domínio *SimpleMathOntology*¹⁹ para a comunidade *Mathematics* e a ontologia *SequenceAligningOntology*²⁰ para a comunidade *Bioinformatics* e para seu repositório de domínios compartilhados, pois ele é o responsável pelas comunidades, ou seja, ele mesmo é o super nó. A partir disso, os outros *peers* foram se conectando ao sistema e selecionando suas comunidades de interesse. O *XPPeer* se conectou na comunidade *Mathematics*, realizando a requisição da ontologia de domínio *SimpleMathOntology* ao super nó. O *VIRTUALW7* ao entrar na rede, seleciona a comunidade *Bioinformatics* e solicita ao super nó a ontologia *SequenceAligningOntology*. E por último, o *peer VIRTUALXP* se conecta em ambos os grupos criados, solicitando, as duas ontologias de domínio ao super nó. Assim, o cenário mostrado pela Figura 4.1 é constituído.

A busca de arquivos entre os nós de uma comunidade semântica é ilustrada na Figura 4.2.

¹⁹ http://plscience.superdignus.com/e_ScienceNet/SimpleMathOntology.owl

²⁰ http://plscience.superdignus.com/e_ScienceNet/SequenceAligningOntology.owl

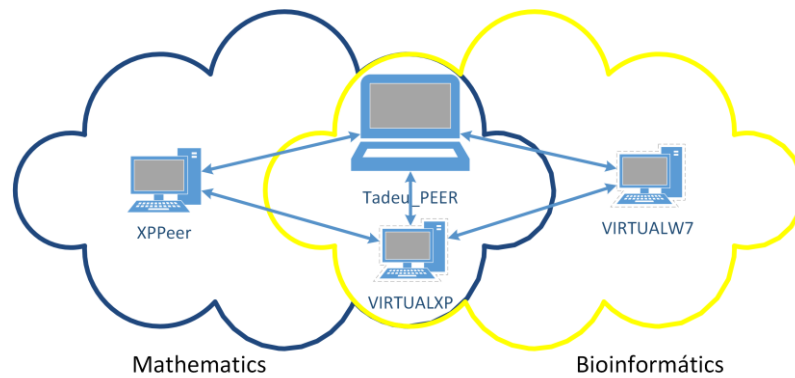


Figura 4.1: Esquema de divisão dos peers nos grupos semânticos no ambiente

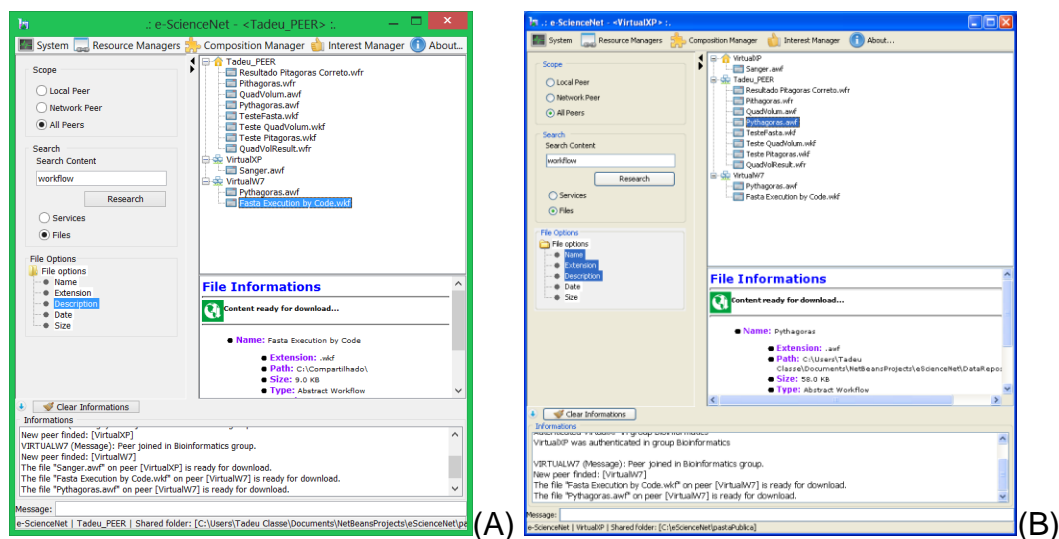


Figura 4.2 A) Busca de arquivo Tadeu_Peer. B) Busca de arquivo VIRTUALXP

Dada essa configuração da e-ScienceNet, as próximas seções detalham as provas de conceito. Essas provas de conceito são embasadas na possibilidade de criação de experimentos científicos em comunidades de interesse, utilizando modelos de composição, serviços web, regras e restrições existentes em ontologias de domínio.

4.2 PROVA DE CONCEITO 1

É possível pensar em uma situação onde um pesquisador matemático necessite realizar simulações computacionais sobre o Teorema de Pitágoras, no intuito de se obter respostas sobre as propriedades de triângulos retângulos.

Conjecturando que este mesmo pesquisador vislumbrasse utilizar este pequeno experimento para simular a idéia de um laboratório colaborativo, de acordo com os preceitos lançados por OLSON et, al. [2009], este cientista precisa de um ambiente colaborativo (Figura 4.3), onde existam ferramentas e recursos disponíveis, os quais possam ser utilizados para obtenção dos resultados esperados, e para o compartilhamento e troca de informações.

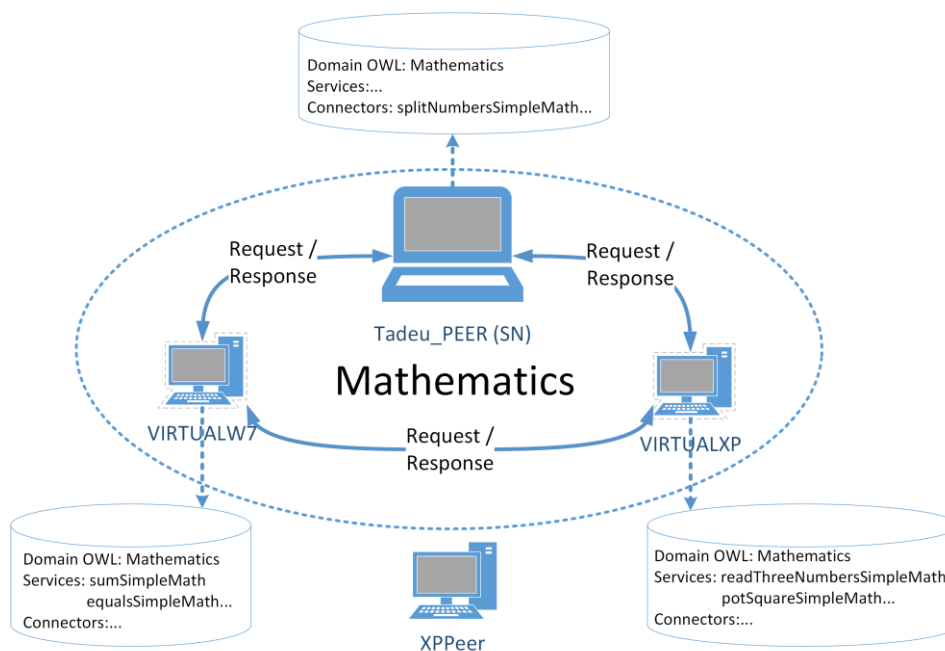


Figura 4.3: Ambiente colaborativo para Mathematics

Existem diversos serviços web disponíveis e que podem ser utilizados para compor este experimento. Entretanto, há a necessidade de colaboração entre os *peers* da rede, para que estes serviços possam ser descritos semanticamente e disponibilizados para buscas, a fim de que a simulação possa ser composta e executada.

O Teorema de Pitágoras [IMENES e LELLIS, 2000] especifica que em qualquer triângulo retângulo a soma dos quadrados dos catetos é igual ao quadrado da hipotenusa (Figura 4.4).

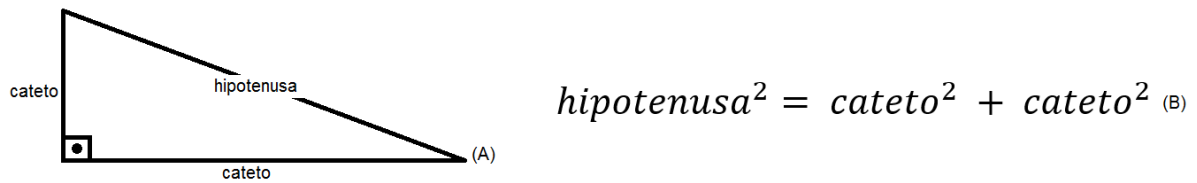


Figura 4.4: A) Representação geométrica. B) Representação algébrica

Uma ontologia de domínio foi desenvolvida para representar termos simples da área matemática recebendo ela o nome de *SimpleMathOntology*²¹ (Figura 4.5). Essa ontologia foi desenvolvida para servir como uma das ontologias de domínio para o grupo de interesse *Mathematics* contendo termos como conceitos matemáticos, conjuntos numéricos e operações algébricas. Nesse sentido, esses termos possibilitam que serviços web relacionados com essa área possam ser anotados com os termos presentes na ontologia.

Utilizando uma máquina de inferência para o processamento desta ontologia alguns resultados importantes para o domínio aparecem, como, por exemplo, nas operações matemáticas apontadas em Figura 4.5 (A) e Figura 4.6, onde a ontologia realiza a inferência de que operações como potência e numeração negativa podem ser consideradas como multiplicações. A partir dessas inferências novas opções de pesquisa são possíveis, pois a abrangência dos termos equivalentes enriquece a busca semântica por serviços.

²¹ http://plscience.superdignus.com/e_ScienceNet/SimpleMathOntology.owl

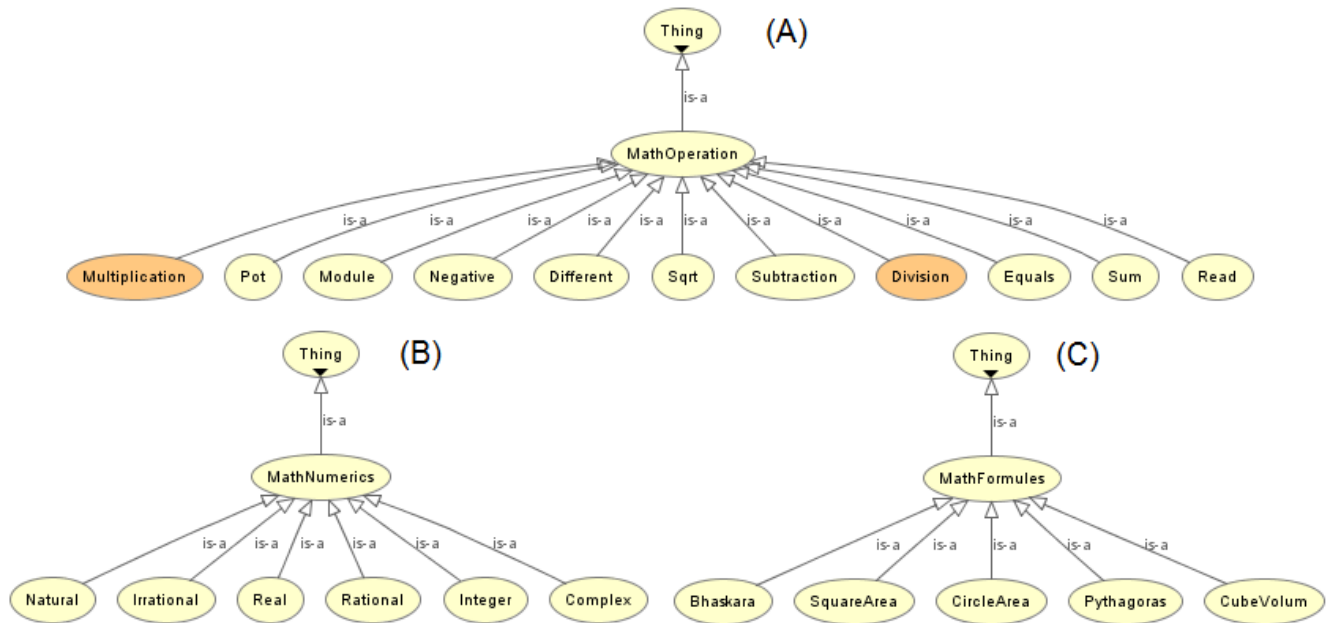


Figura 4.5: SimpleMathOntology

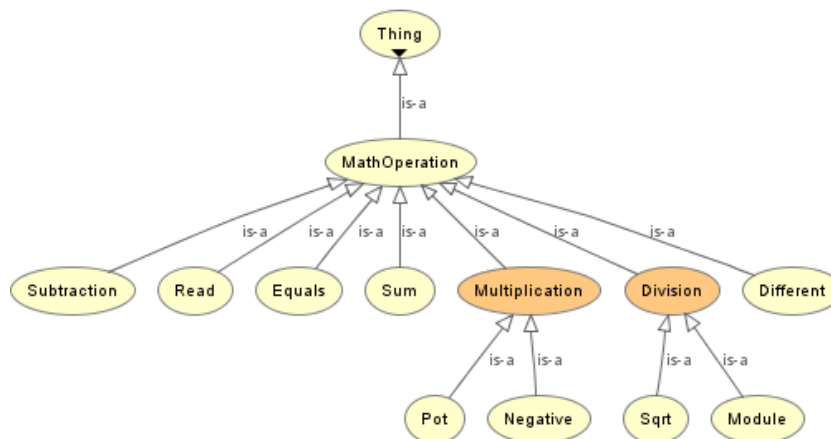


Figura 4.6: Modelo Inferido nas operações de SimpleMathOntology

Outra funcionalidade dessa ontologia é fornecer regras para a execução de conceitos matemáticos, englobados aqui através da classe *MathFormules* conceitos como fórmula de Bhaskara, áreas de círculo e quadrado e o teorema de Pitágoras. Cada uma dessas fórmulas é relacionada com operações matemáticas através das restrições como *workflow_task* (Figura 4.7 e Figura 4.8).

```

<owl:ObjectProperty rdf:about="http://www.esciencenet.com.br/tadeuclasse/esciencenet/SimpleMathOntology.owl#workflows_task">
  <rdfs:domain rdf:resource="http://www.esciencenet.com.br/tadeuclasse/esciencenet/SimpleMathOntology.owl#MathFormules"/>
  <rdfs:range rdf:resource="http://www.esciencenet.com.br/tadeuclasse/esciencenet/SimpleMathOntology.owl#MathOperation"/>
</owl:ObjectProperty>

```

Figura 4.7: Restrição workflow_task entre classes MathFormule e MathOperations

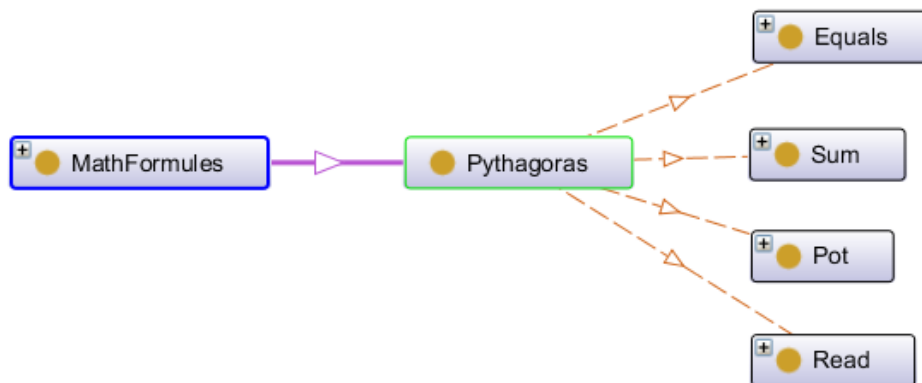


Figura 4.8: Operações relacionadas ao teorema de Pitágoras

Conforme mencionado no capítulo 3, para que sejam pesquisados recursos na e-ScienceNet é necessário que estes, sejam inclusos e compartilhados entre os *peers*. Sendo assim, existe a necessidade de inclusão de serviços matemáticos através de serviços web. Abaixo são apresentados os serviços inclusos para a execução da composição de Pitágoras:

- **Serviço Read:** o serviço *Read* é responsável por ler números informados pelos usuários. A operação usa o serviço web *readThreeNumbers*, pois o teorema de Pitágoras necessita de três valores (hipotenusa e dois catetos). A Figura 4.9 mostra a anotação semântica do serviço com termos da *SimpleMathOntology*. A Figura 4.10 mostra a associação do parâmetro *return* com termos da ontologia de domínio. E a Tabela 2, mostra os parâmetros do serviço e sua associação com os termos da ontologia de domínio.

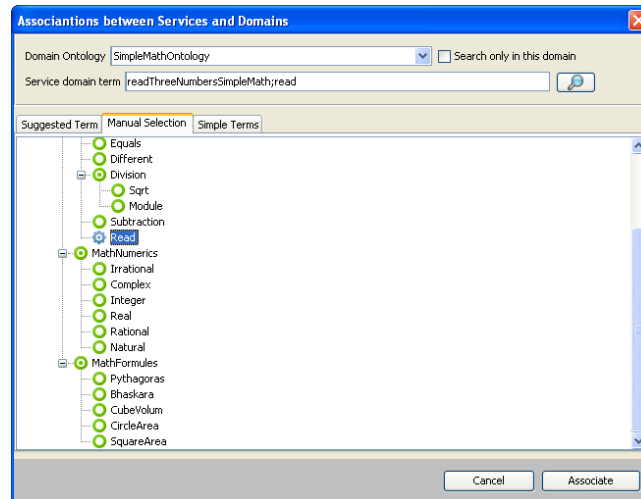


Figura 4.9: Anotação semântica do serviço Read

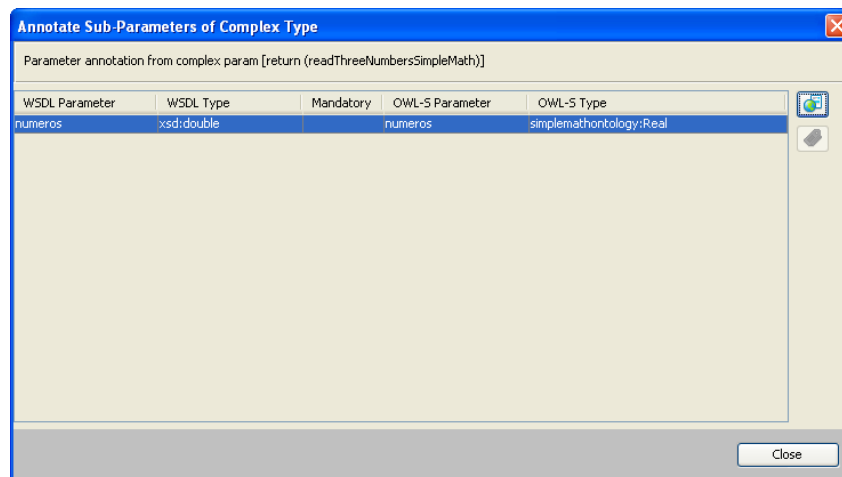


Figura 4.10: Anotação semântica do parâmetro return do serviço Read

Tabela 2: Parâmetros do serviço Read

Parâmetro	Tipo	É Complexo?	Termo de Domínio	Ontologia de Domínio
numberOne	Entrada	Não	:Real	SimpleMathOntology
numberTwo	Entrada	Não	:Real	SimpleMathOntology
numberThree	Entrada	Não	:Real	SimpleMathOntology
Return	Saída	Sim	Owl:Thing	
numeros	Saída	Não	:Real	SimpleMathOntology

- **Serviço Pot:** o serviço *Pot* é responsável por calcular a potência quadrada de um número. Nesta Prova de Conceito foi usado o serviço *potSquareSimpleMath*, pois o teorema de Pitágoras necessita da operação de potência para elevar hipotenusa e catetos ao quadrado. A Figura 4.11 mostra a interface de inclusão para o serviço *Pot* na e-

ScienceNet e a Figura 4.12 exibe a anotação semântica para ele. A relação da associação dos termos da ontologia de domínio com os parâmetros do serviço pode ser vista na Tabela 3.

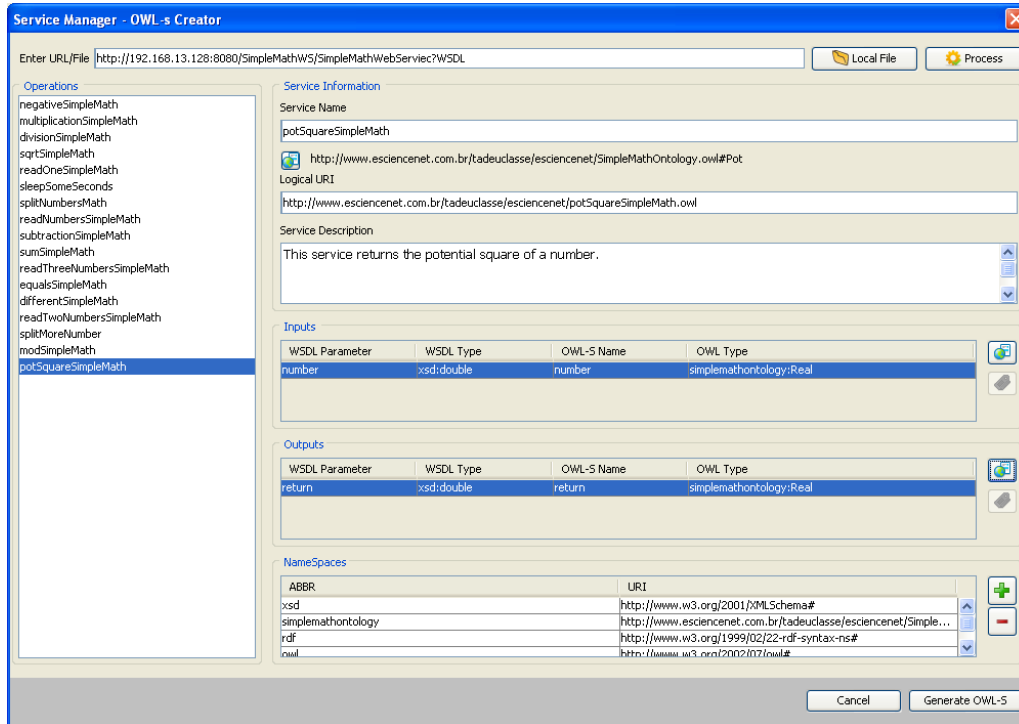


Figura 4.11: Inclusão do serviço Pot na e-ScienceNet

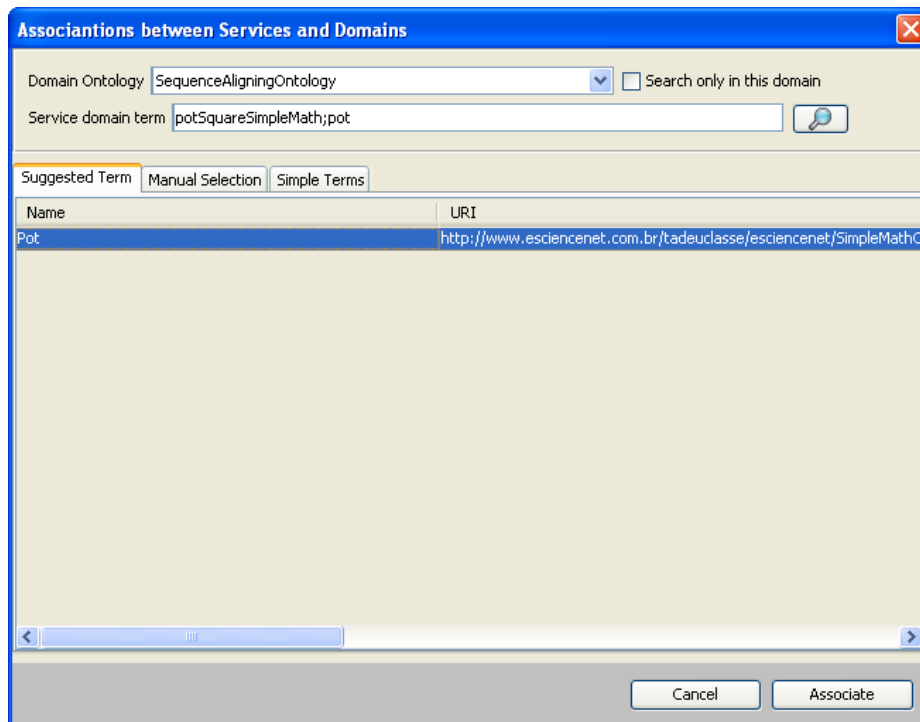


Figura 4.12: Anotação semântica do serviço Pot

Tabela 3: Parâmetros do Serviço Pot

Parâmetro	Tipo	É Complexo?	Termo de Domínio	Ontologia de Domínio
Number	Entrada	Não	:Real	SimpleMathOntology
Return	Saída	Não	:Real	SimpleMathOntology

- **Serviço *Sum*:** o serviço *Sum* (*sumSimpleMath*) é o responsável por realizar a soma entre dois números. A Tabela 4 mostra as anotações semânticas entre seus parâmetros.

Tabela 4: Parâmetros do serviço Sum

Parâmetro	Tipo	É Complexo?	Termo de Domínio	Ontologia de Domínio
NumberOne	Entrada	Não	:Real	SimpleMathOntology
NumberTwo	Entrada	Não	:Real	SimpleMathOntology
NumberThree	Entrada	Não	:Real	SimpleMathOntology
Return	Saída	Não	:Real	SimpleMathOntology

- **Serviço *Equals*:** este serviço na composição se torna necessário para comparar a igualdade entre os lados de uma equação retornando verdadeiro ou falso (*equalsSimpleMath*). A Tabela 5 mostra a relação para a anotação semântica dos parâmetros.

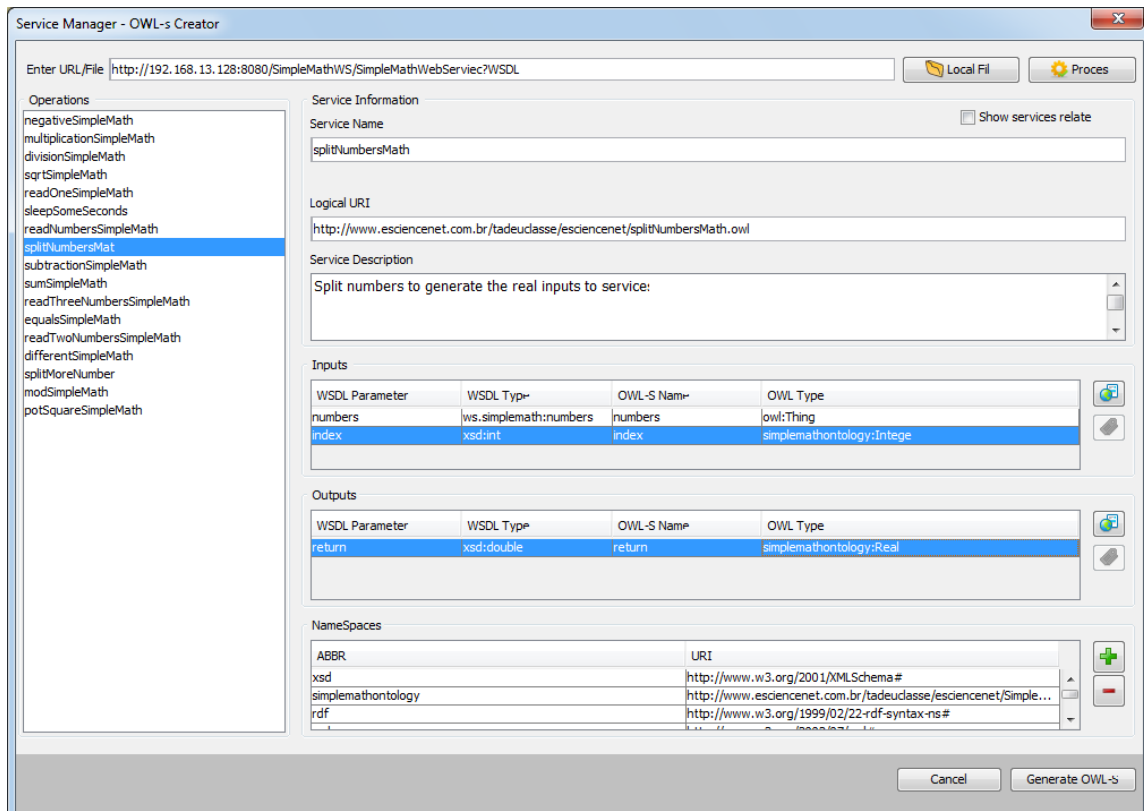
Tabela 5: Parâmetros do serviço Equals

Parâmetro	Tipo	É Complexo?	Termo de Domínio	Ontologia de Domínio
NumberOne	Entrada	Não	:Real	SimpleMathOntology
NumberTwo	Entrada	Não	:Real	SimpleMathOntology
Return	Saída	Não	Xsd:Boolean	

- **Conector *SplitNumbers*:** ao se utilizar a operação *Read*, a mesma retorna uma lista numérica, em outras palavras, para que seja utilizado corretamente os números e estes, passados para os próximos serviços corretamente, o uso de um conector que separa essa lista é necessário. Para isso, utilizou-se o serviço *splitNumbersSimpleMath*, o qual recebe uma lista de números e retorna uma de suas posições. A Figura 4.13 mostra a inclusão do conector na e-ScienceNet e a Tabela 6 mostra a relação entre as suas anotações semânticas;

Tabela 6: Parâmetros de SplitNumbers

Parâmetro	Tipo	É Complexo?	Termo de Domínio	Ontologia de Domínio
Numbers	Entrada	Sim	Owl:Thing	
Number	Entrada	Não	:Real	SimpleMathOntology
Index	Entrada	Não	:Integer	SimpleMathOntology
Return	Saída	Não	:Real	SimpleMathOntology

**Figura 4.13:** Inserindo o conector SplitNumbers

Com os serviços incluídos na e-ScienceNet, o modelo de composição para a criação do teorema de Pitágoras precisa ser criado. Este modelo é especificado na e-ScienceNet através do desenvolvimento do modelo abstrato de *workflow*. Este modelo é o ponto de partida para busca de serviços para as composições e execução do experimento.

A partir do carregamento da ontologia de domínio *SimpleMathOntology*, o usuário seleciona a fórmula que relaciona as tarefas que formam o fluxo de trabalho e, utilizando regras e restrições existentes na ontologia, ele pode sequencialmente modelar o fluxo de serviços. Como visto na Figura 4.4 B, além da leitura dos valores, é necessário especificar uma sequência de operações

matemáticas para a obtenção do resultado. Essa sequência de operações também pode ser obtida através de propriedades da ontologia de domínio como *previous_task* e *next_task*, as quais auxiliam o usuário na ordem correta de criação do modelo.

Um conceito importante que deve ser levado em consideração aqui é a seleção da conexão lógica entre as tarefas, que podem ser sequenciais ou não. Neste contexto, são utilizados conceitos relacionados a redes de Petri [PETRI, 1962], aos quais se aplicam o formalismo lógico nos fluxos de trabalho por meio de conexões como *SEQUENCE*, *AND-SPLIT*, *OR-SPLIT*, *AND-JOIN* ou *OR-JOIN*.

As figuras 4.14, 4.15, 4.16, 4.17 e 4.18 mostram respectivamente a inclusão das tarefas *Read*, *Pot*, *Sum*, *Equals* e a finalização da composição no modelo abstrato de *workflow* junto com suas conexões lógicas.

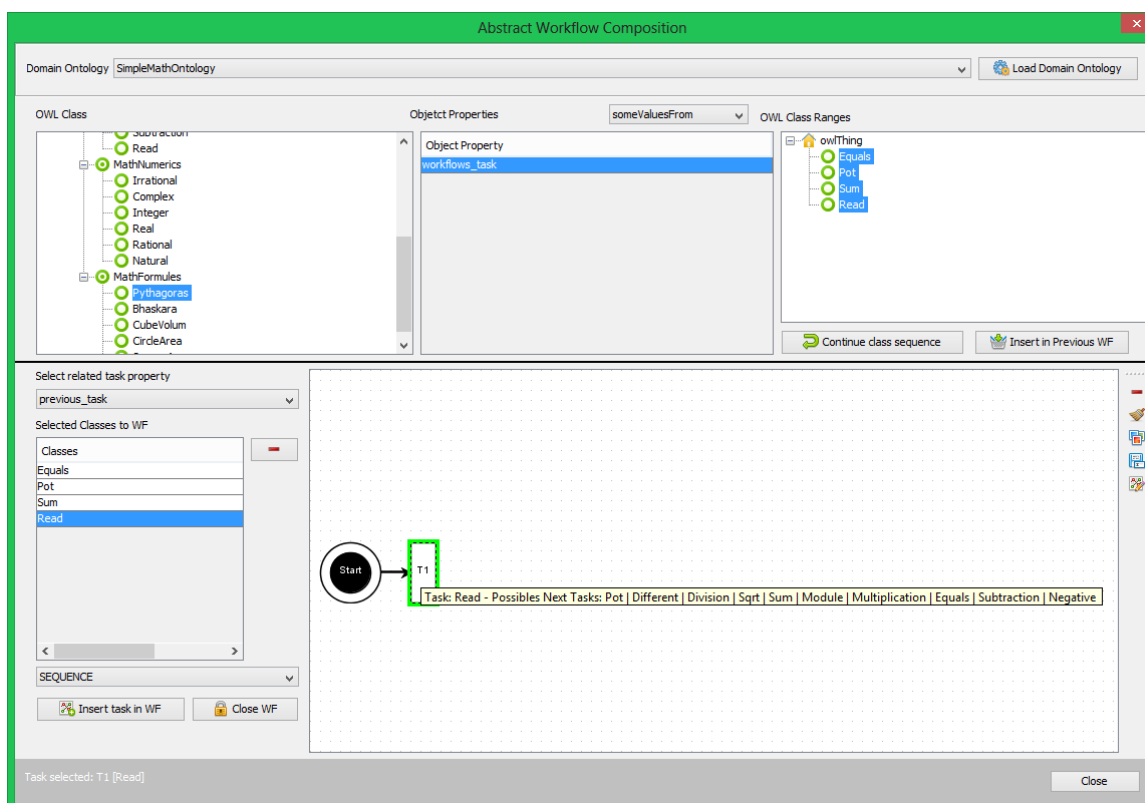


Figura 4.14: Inclusão da tarefa Read no AWF

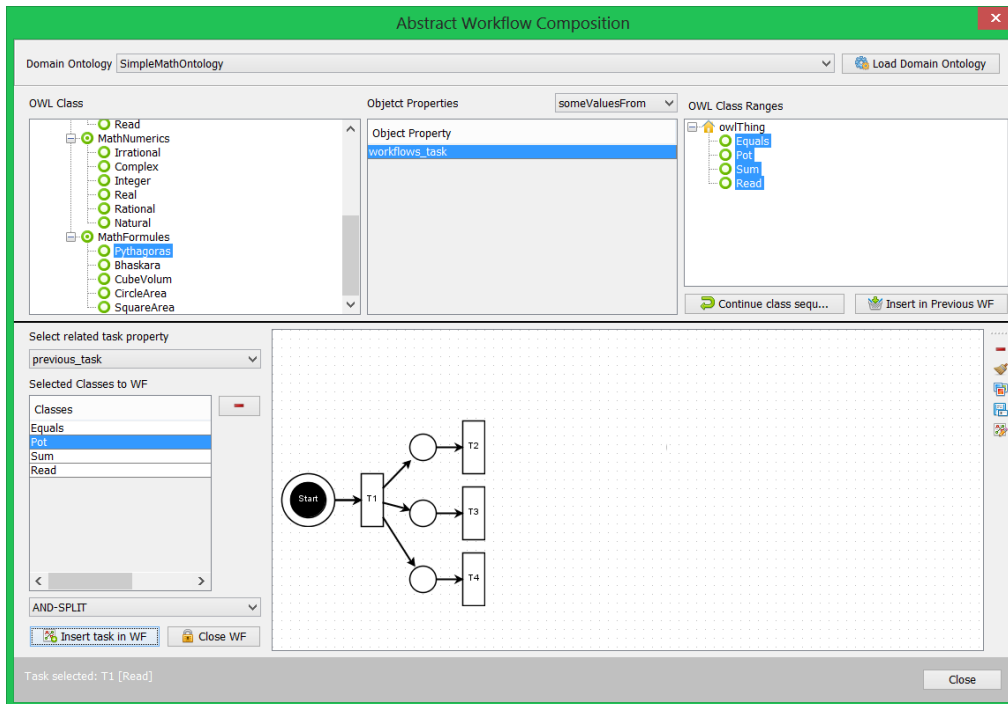


Figura 4.15: inclusão das tarefas Pot e sua conexão AND-SPLIT com a tarefa Read

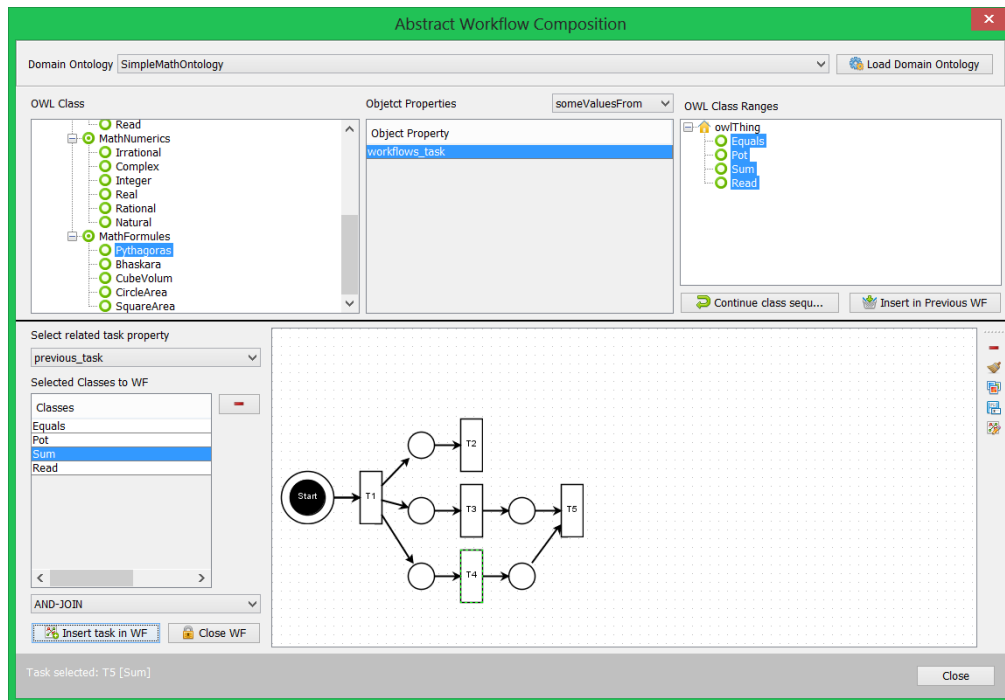


Figura 4.16: Inclusão da tarefa Sum e a conexão AND-JOIN entre os catetos

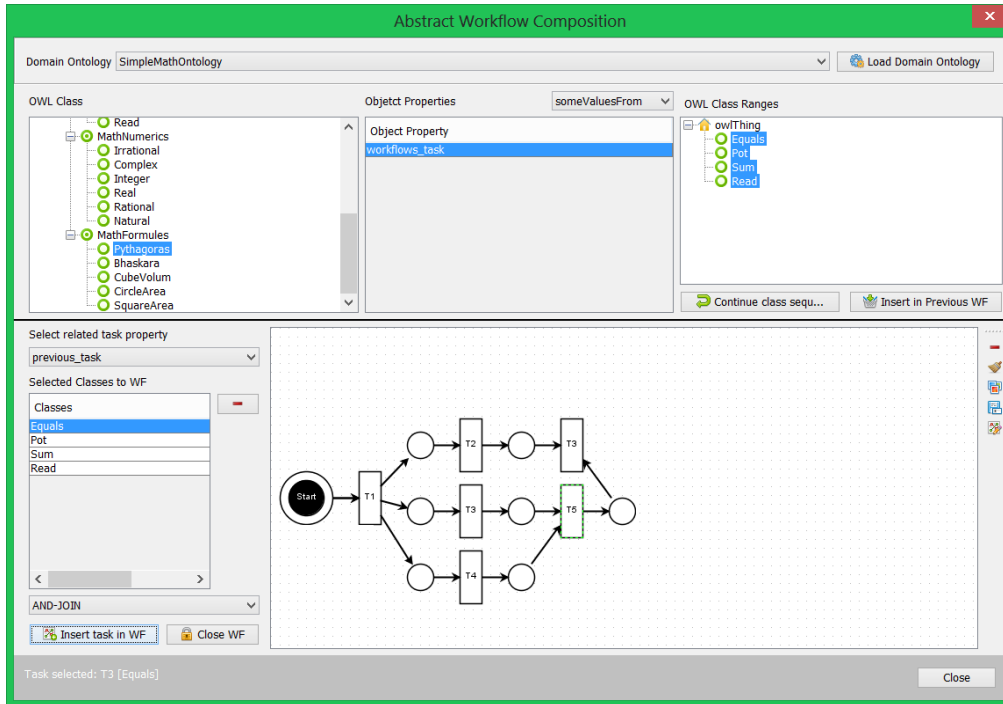


Figura 4.17: Inclusão da tarefa Equals e a conexão AND-JOIN com a hipotenusa

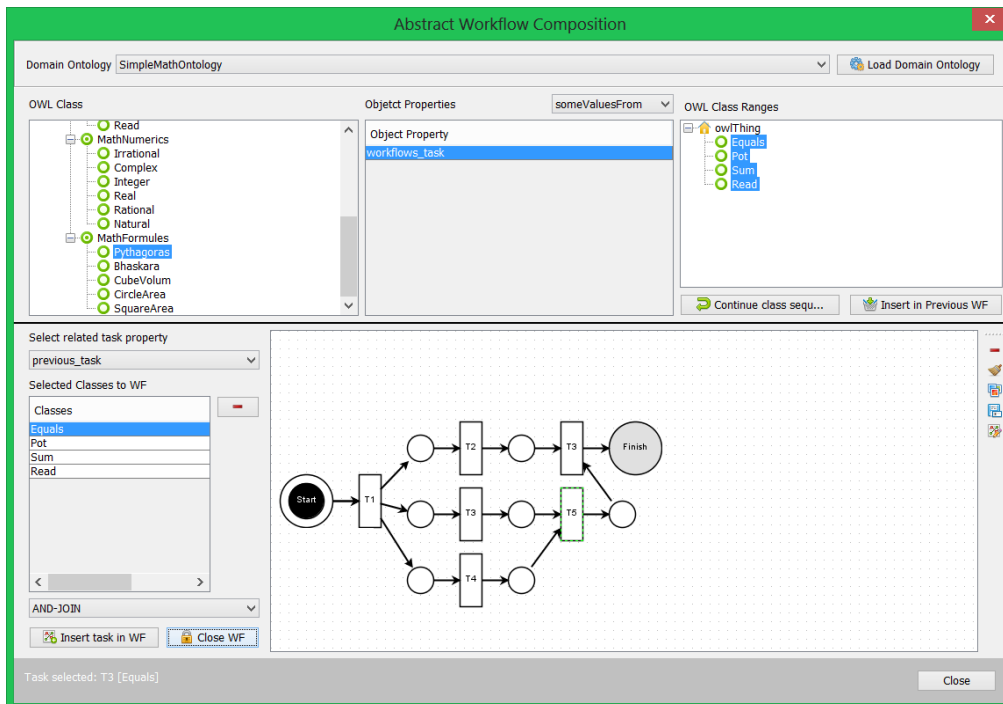


Figura 4.18: Finalização da composição de Pitágoras

Com o desenvolvimento deste modelo a próxima tarefa no processo do desenvolvimento do experimento é a pesquisa dos serviços web entre os *peers* disponíveis na comunidade semântica *Mathematics*. Baseado no modelo, o

usuário deve carregá-lo no Gerente de Pesquisa (Figura 4.19) o qual possibilita a realização de buscas de cada um dos serviços existentes e adicioná-los como potenciais serviços na sua composição.

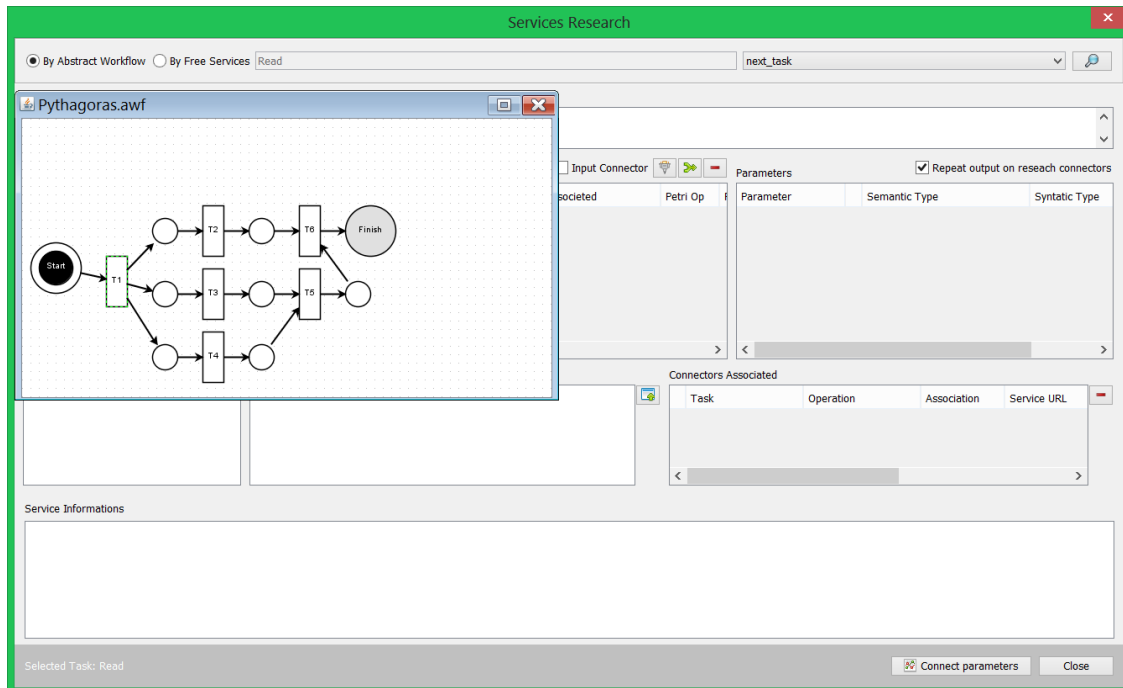


Figura 4.19: Carregando o modelo abstrato no gerente de pesquisa de serviços

Como visto nas seções sobre os Gerentes de Pesquisa (seção 3.2.7) e Composição (seção 3.2.8), a pesquisa não é feita somente levando em consideração os nomes das tarefas, mas sim, a inferência de termos relacionados a tarefa, o que leva à comparação semântica e sintática de serviços verificando a sua compatibilidade com a composição. Além disso, foi demonstrado que a tarefa *Read* necessita de um conector para separar os números recebidos para os próximos serviços. Obedecendo as tarefas dos modelos, as buscas por serviços são respectivamente: Figura 4.20, Figura 4.21, Figura 4.22, Figura 4.23 e Figura 4.24.

The screenshot shows the 'Services Research' window with the search criteria 'Read' and 'task_performed_by'. The 'Services Found' list includes 'VirtualW7', 'Tadeu_PEER', and 'readThreeNumbersSimpleMath'. The 'Selected Services' table is empty. The 'Parameters' table is also empty. The 'Service Informations' section for 'readThreeNumbersSimpleMath' shows the following details:

- Service Name:** readThreeNumbersSimpleMath
- Service Description:** This service read three numbers and return its in a List.
- Domain Term:** Read
- Inputs:**
 - numberOne [Semantic]:** <http://www.esciencenet.com.br/tadeuclasse/esciencenet/SimpleMathOntology.owl#Real> - [Syntactic]: <http://www.w3.org/2001/XMLSchema#double>
 - numberThree [Semantic]:** <http://www.esciencenet.com.br/tadeuclasse/esciencenet/SimpleMathOntology.owl#Real> - [Syntactic]: <http://www.w3.org/2001/XMLSchema#double>
 - numberTwo [Semantic]:** <http://www.esciencenet.com.br/tadeuclasse/esciencenet/SimpleMathOntology.owl#Real> - [Syntactic]: <http://www.w3.org/2001/XMLSchema#double>

Figura 4.20: Busca dos serviços relacionados a tarefa Read

The screenshot shows the 'Services Research' window with the search criteria 'Read' and 'task_performed_by'. The 'Services Found' list includes 'VirtualXP', 'VirtualW7', 'Tadeu_PEER', and 'splitNumbersMath_1'. The 'Selected Services' table is empty. The 'Parameters' table is populated with the following data:

Parameter	Semantic Type	Syntactic Type
numberOne	Real	double
numberThree	Real	double
numberTwo	Real	double
return	Real	double

The 'Connectors Associated' table shows the following data:

Task	Operation	Association	Service URL
readThreeNumbersSI...	splitNumbersMath	numeros == n...	http://192.168.1...

The 'Service Informations' section for 'splitNumbersMath' shows the following details:

- Connector Name:** splitNumbersMath
- Connector Description:** Auto generated from <http://192.168.13.128:8080/SimpleMathWS/SimpleMathWebService?WSDL>
- Inputs:**
 - index [Semantic]:** <http://www.esciencenet.com.br/tadeuclasse/esciencenet/SimpleMathOntology.owl#Integer> - [Syntactic]: <http://www.w3.org/2001/XMLSchema#int>
 - numeros [Semantic]:** <http://www.esciencenet.com.br/tadeuclasse/esciencenet/SimpleMathOntology.owl#Real> - [Syntactic]: <http://www.w3.org/2001/XMLSchema#double>
 - index [Semantic]:** <http://www.esciencenet.com.br/tadeuclasse/esciencenet/SimpleMathOntology.owl#Integer> - [Syntactic]: <http://www.w3.org/2001/XMLSchema#int>
- Outputs:**

Figura 4.21: Busca do conector Split para a tarefa Read

The screenshot shows the 'Services Research' application window. At the top, the search criteria are 'By Abstract Workflow' and 'By Free Services', with the search term 'Pot' and a dropdown menu set to 'task_performed_by'. The 'Related terms' section displays 'Pot - Negative - Multiplication'. The 'Services Finded' tree on the left lists several services, with 'potSquareSimpleMath - [Pot]' selected. The 'Selected Services' table is as follows:

Task	Operation	Domain Associated	Petri Op
Read	readThreeNumbersSi...	Read	AND-SPLIT

The 'Parameters' table is:

Parameter	Semantic Type	Syntatic Type
numberOne	Real	double
numberThree	Real	double
numberTwo	Real	double
return	Real	double
return	Real	double
return	Real	double

The 'Connectors Associated' table is:

Task	Operation	Association	Service URL
readThreeNumbersSi...	splitNumbersMath	numeros == n...	http://192.168.1...
readThreeNumbersSi...	splitNumbersMath_1	numeros == n...	http://192.168.1...
readThreeNumbersSi...	splitNumbersMath_2	numeros == n...	http://192.168.1...

The 'Service Informations' section for 'potSquareSimpleMath' includes:

- Service Name:** potSquareSimpleMath
- Service Description:** This service returns the potential square of a number.
- Domain Term:** Pot
- Inputs:**
 - number [Semantic]:** <http://www.esccenet.com.br/tadeuclasse/esccenet/SimpleMathOntology.owl#Real> - **[Syntatic]:** <http://www.w3.org/2001/XMLSchema#double>
- Outputs:**

The 'Selected Task' is 'Pot'. Buttons for 'Connect parameters' and 'Close' are visible at the bottom right.

Figura 4.22: Busca de serviços para a tarefa Pot

The screenshot shows the 'Services Research' application window. At the top, the search criteria are 'By Abstract Workflow' and 'By Free Services', with the search term 'Sum' and a dropdown menu set to 'task_performed_by'. The 'Related terms' section displays 'Sum'. The 'Services Finded' tree on the left lists several services, with 'sumSimpleMath - [Sum]' selected. The 'Selected Services' table is as follows:

Task	Operation	Domain Associated	Petri Op
Read	readThreeNumbersSi...	Read	AND-SPLIT
Pot	potSquareSimpleMath	Pot	AND-JOIN
Pot	potSquareSimpleMat...	Pot	AND-JOIN
Pot	potSquareSimpleMat...	Pot	AND-JOIN

The 'Parameters' table is:

Parameter	Semantic Type	Syntatic Type
number	Real	double
return	Real	double

The 'Connectors Associated' table is empty.

The 'Service Informations' section for 'potSquareSimpleMath_1' includes:

- Service Name:** potSquareSimpleMath_1
- Service Description:** This service returns the potential square of a number.
- Domain Term:** Pot
- Inputs:**
 - number [Semantic]:** <http://www.esccenet.com.br/tadeuclasse/esccenet/SimpleMathOntology.owl#Real> - **[Syntatic]:** <http://www.w3.org/2001/XMLSchema#double>
- Outputs:**

The 'Selected Task' is 'Sum'. Buttons for 'Connect parameters' and 'Close' are visible at the bottom right.

Figura 4.23: Busca de serviços relacionados a tarefa Sum

The screenshot shows the 'Services Research' window with the search criteria set to 'Equals'. The interface is divided into several sections:

- Related terms:** Shows 'Equals'.
- Services Found:** A tree view on the left shows a search tree with 'equalsSimpleMath - [Equals]' selected.
- Selected Services:** A table listing services found for the task 'Equals':

Task	Operation	Domain Associated	Petri Op
Read	readThreeNumbersSi...	Read	AND-SPLIT
Pot	potSquareSimpleMath...	Pot	AND-JOIN
Pot	potSquareSimpleMat...	Pot	AND-JOIN
Sum	sumSimpleMath	Sum	AND-JOIN
- Parameters:** A table showing parameters for the selected services:

Parameter	Semantic Type	Syntactic Type
numberOne	Real	double
numberTwo	Real	double
return	Real	double
- Connectors Results:** An empty table.
- Connectors Associated:** An empty table.
- Service Informations:**
 - Service Name:** equalsSimpleMath
 - Service Description:** Compare two numbers to discover if both are equals.
 - Domain Term:** Equals
 - Inputs:**
 - number [Semantic]:** <http://www.esciencenet.com.br/tadeuclasse/esciencenet/SimpleMathOntology.owl#Real> - [Syntactic]: <http://www.w3.org/2001/XMLSchema#double>
 - number2 [Semantic]:** <http://www.esciencenet.com.br/tadeuclasse/esciencenet/SimpleMathOntology.owl#Real> - [Syntactic]: <http://www.w3.org/2001/XMLSchema#double>

At the bottom, the 'Selected Task' is 'Equals', and there are 'Connect parameters' and 'Close' buttons.

Figura 4.24: Busca de serviços relacionados a tarefa Equals

Terminado o processo de busca dos serviços compatíveis é necessário que seja realizado o processo de conexão entre os parâmetros dos serviços (Figura 4.25). Alguns como é notório, precisam de valores manuais, como por exemplo, os primeiros serviços necessitam que sejam informados os valores iniciais para seus parâmetros de entrada, a fim de se obter um ponto de partida para todo o processo. As configurações de cada parâmetro segue o descrito na Tabela 7.

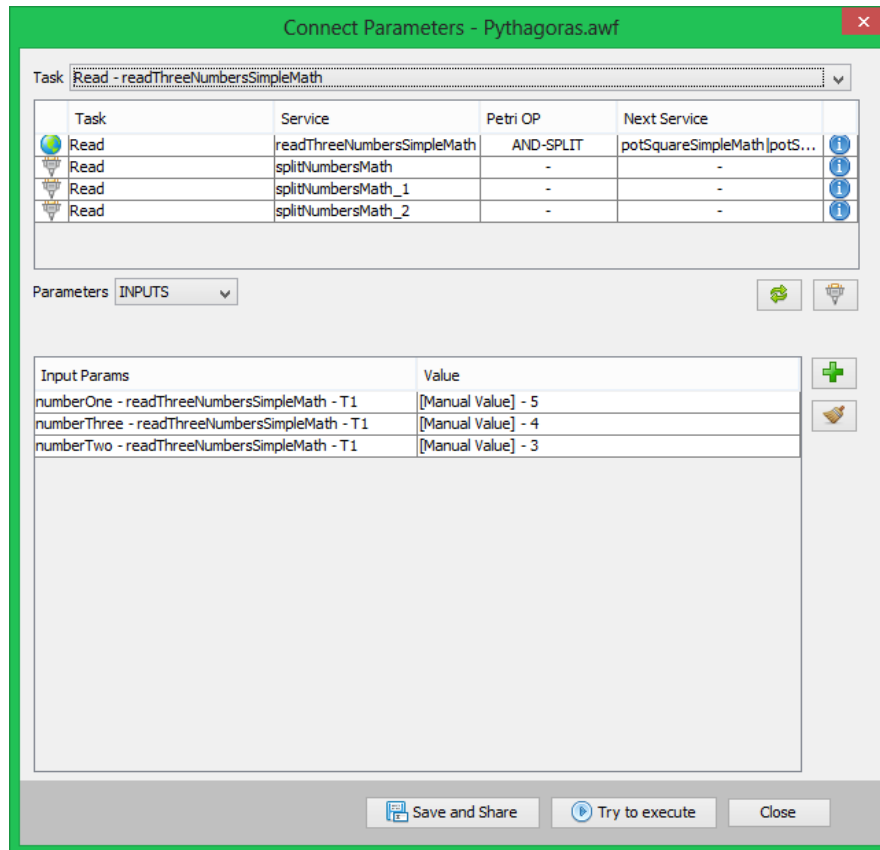


Figura 4.25: Configuração das conexões entre os serviços.

Tabela 7: Conexão entre parâmetros e valores manuais

Serviço / Tarefa	Parâmetro	Tipo	Manual	Valor
readThreeNumbers / Read	NumberOne	Entrada	X	5
	NumberTwo	Entrada	X	4
	NumberThree	Entrada	X	3
	Retun	Saída		splitNumbers:Numbers splitNumbers_1:Numbers splitNumbers_2:Numbers readThreeNumbers:Return
splitNumbers / Conector de Read	Numbers	Entrada		readThreeNumbers:Return
	Index	Entrada	X	0
	Retun	Saída		potSquare:Number
splitNumbers_1 / Conector de Read	Numbers	Entrada		readThreeNumbers:Return
	Index	Entrada	X	1
	Retun	Saída		potSquare_1:Number
splitNumbers_2 / Conector de Read	Numbers	Entrada		readThreeNumbers:Return
	Index	Entrada	X	2
	Retun	Saída		potSquare_2:Number
potSquare / Pot	Number	Entrada		splitNumber:Return
	Return	Saída		equals:NumberOne
potSquare_1 / Pot	Number	Entrada		splitNumber_1:Return
	Return	Saída		sum:NumberOne
potSquare_2 / Pot	Number	Entrada		splitNumber_2:Return
	Return	Saída		sum:NumberOne

sum / Sum	NumberOne	Entrada		potSquare_1:Return
	NumberTwo	Entrada		potSquare_2:Return
	Return	Saída		equals:NumberTwo
equals / Equals	NumberOne	Entrada		potSquare:Return
	NumberTwo	Entrada		sum:Return
	Return	Saída		Resultado Final

Finalizado o processo de conexão o usuário pode salvar o que é chamado de composição funcional, ou seja, um arquivo onde é armazenada toda a configuração de conexão entre os parâmetros permitindo seu compartilhamento para execução em outros *peers* e futuras alterações. Neste momento também é permitido o acesso a funcionalidade de execução da composição como pode ser visto na Figura 4.26.

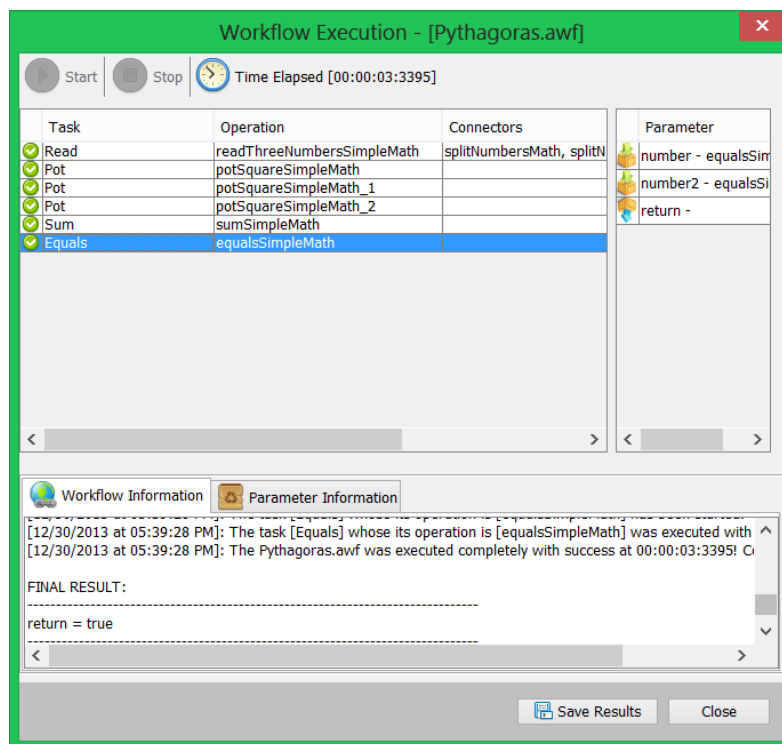


Figura 4.26: Execução da composição de Pitágoras

De acordo com a fórmula do teorema, ao final dessa execução é verificado se o quadrado da hipotenusa é igual à soma dos quadrados dos catetos, cabendo à tarefa *Equals* a resposta entre *true* ou *false*. O resultado dessa composição pode ser salvo em arquivo e este pode ser analisado, editado e compartilhado entre os *peers*.

Foi realizado também nesta Prova de Conceito o compartilhamento da composição funcional para outros *peers* conectados à comunidade semântica *Mathematics* e o resultado disso foi que, ao terminar de fazer o *download* do arquivo funcional, o *peer* conseguiu executar a mesma composição criada anteriormente sem realizar nenhuma alteração. Tal experimento é retratado através da Figura 4.27.

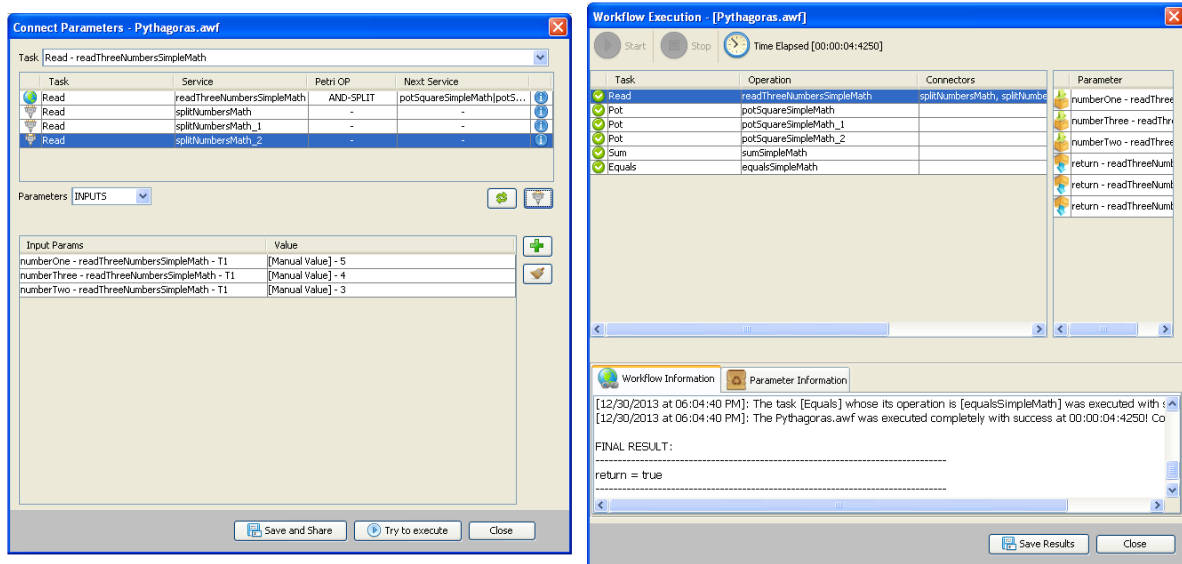


Figura 4.27: Execução do workflow funcional de Pitágoras em outro peer da e-ScienceNet

Apesar de apresentar uma composição relativamente simples, esta prova de conceito apresentou indícios de que um cientista, baseando-se na proposta de laboratórios colaborativos entre pesquisadores e compartilhamento de recursos, e, também, apoiado por termos e regras disponíveis nas ontologias de domínio, pode: especificar um modelo abstrato do seu experimento; e realizar a busca dos serviços web semânticos que estejam compartilhados entre os *peers* da comunidade semântica, contando ainda, com a inferência nas ontologias de domínio, de termos que estejam associados à tarefa a ser pesquisada. Após a descoberta dos serviços web e sua composição, o cientista conta com um modelo funcional de seu experimento, o qual pode ser executado para a visualização de resultados e compartilhamento.

4.3 PROVA DE CONCEITO 2

No intuito de demonstrar o uso da e-ScienceNet para a criação de experimentos científicos mais complexos em um ambiente colaborativo, esta prova de conceito irá tratar do domínio de Bioinformática, subárea de alinhamento/sequenciamento genético.

Partindo deste domínio, é possível imaginar um cenário em que um biólogo, após colher amostras de alguns seres, realiza o processo de sequenciamento de tal material em um sequenciador genético, desejando obter informações sobre elas através de mecanismos de comparação. Existe um conceito biológico afirmando que: “se duas sequências (nucleotídeos ou proteínas) apresentam similaridades, é possível que suas funções sejam razoavelmente similares.” [DA SILVA, BRAGA e CAMPOS, 2012]. Baseando-se neste conceito, o pesquisador deseja saber informações sobre as suas amostras, verificando, em primeiro lugar, se as mesmas já foram processadas por algum outro cientista. Para obter tais informações, o cientista utiliza algoritmos de alinhamento de sequências. Neste contexto, DA SILVA, BRAGA e CAMPOS [2012] replicou em seu trabalho um modelo de composição de serviços capaz de realizar o alinhamento entre sequências genéticas, o qual é apresentado na Figura 4.28.

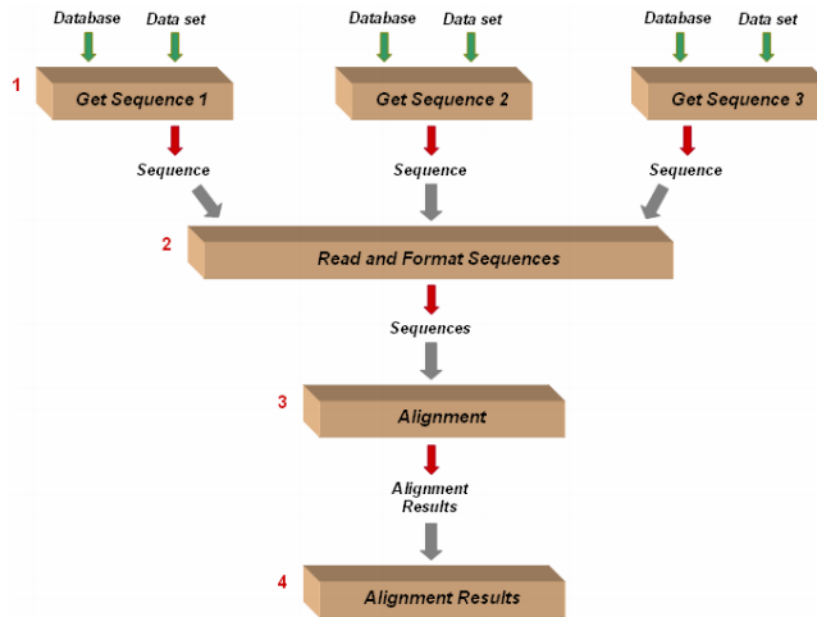


Figura 4.28: Workflow para alinhamento de seqüências proposto por DA SILVA, BRAGA e CAMPOS [2012]

Este modelo de composição apresenta a seguinte seqüência de execução: 1) as seqüências são recuperadas através de um serviço que faz acesso a uma base de dados de seqüências biológicas; 2) As seqüências recuperadas são lidas e formatadas; 3) De posse das seqüências formatadas, o serviço realiza o alinhamento baseando-se em algum algoritmo de alinhamento; 4) Os resultados são apresentados aos cientistas.

Com a ajuda semântica fornecida pelas ontologias de domínio existentes na e-ScienceNet, é possível que o cientista (que não tenha conhecimento suficiente do domínio computacional) consiga desenvolver, de maneira facilitada, o modelo a ser executado. Assim, esta PoC se baseia na implementação real do modelo de *workflow* proposto por DA SILVA, BRAGA e CAMPOS [2012] na e-ScienceNet, com o intuito de verificar a viabilidade do desenvolvimento deste *workflow* já especificado.

Considerando a proposta de laboratórios colaborativos, este cientista ao entrar na e-ScienceNet se depara com a comunidade *Bioinformatics*. Tal comunidade é a responsável por concentrar os pesquisadores que tenham interesse nas subáreas deste domínio, a fim de compartilhar recursos, conceitos e resultados entre seus colaboradores. Sendo assim, os cientistas podem solicitar recursos e serviços de outros membros, tendo em vista a

execução da pesquisa. Dentro desta comunidade semântica, existem as ontologias de domínio, as quais fornecem termos, regras e conhecimento sobre um domínio específico. Uma das ontologias de domínio disponíveis nesse grupo de interesse é a *SequenceAligningOntology* detalhada no trabalho de COSTA et al. [2013]. No entanto, algumas modificações foram necessárias na *SequenceAligningOntology* para que forneça suporte semântico a outros serviços e termos relacionados ao alinhamento de sequências (Figura 4.29). Notadamente, as tarefas definidas em DA SILVA, BRAGA e CAMPOS [2012]. A ontologia completa pode ser obtida em: http://plscience.superdignus.com/e_ScienceNet/SequenceAligningOntology.owl.

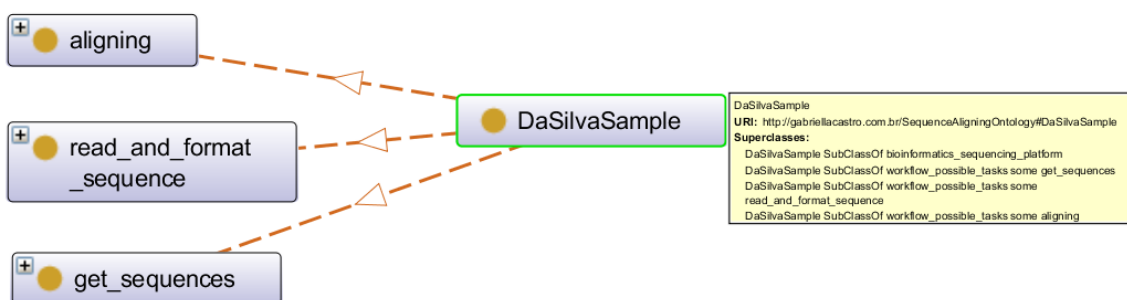


Figura 4.29: DaSilvaSampe workflows_possible_tasks em SequenceAligningOntology

Neste contexto, os serviços e os conectores correspondentes às tarefas que descrevem o modelo devem ser disponibilizados pelos *peers* da e-ScienceNet que estejam conectados na comunidade *Bioinformatics* para que seja possível a busca por estes serviços. O processo de anotação semântica de um *web service* na e-ScienceNet, é sempre similar alterando apenas os termos da ontologia de domínio para a anotação semântica. Abaixo, estão destacados os *web services* utilizados nesta PoC 2:

- **GetSequence:** a tarefa *getSequence* corresponde a busca por sequências em bases de dados biológicas. O serviço web relacionado e disponibilizado na e-ScienceNet é o *fetchBatch*, que busca uma sequência (nucleotídeo ou proteína), passando a ela o código fasta e o nome da base de dados como parâmetros principais. A Figura 4.30

mostra a anotação semântica para o serviço e a Tabela 8 exibe a relação dos termos para os parâmetros do serviço;

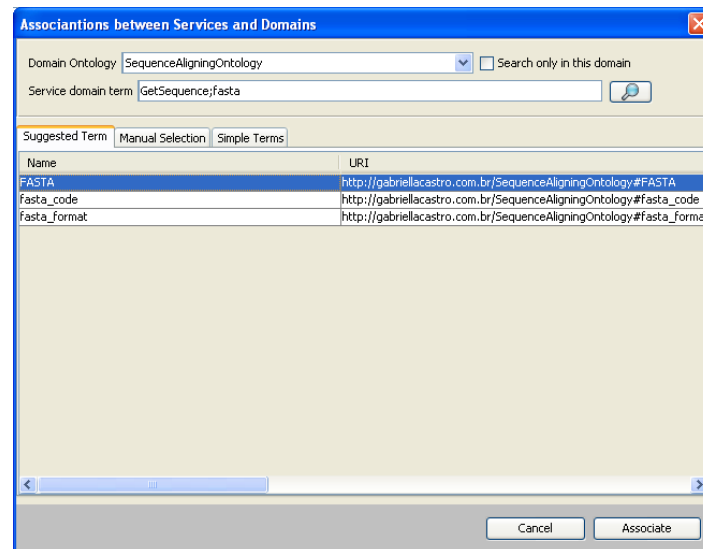


Figura 4.30: Anotação semântica para a tarefa GetSequence

Tabela 8: Parâmetros de GetSequence

Parâmetro	Tipo	É Complexo?	Termo de Domínio	Ontologia de Domínio
Db	Entrada	Não	:sequence_database	SequenceAligninOntolgy
Ids	Entrada	Não	:fasta_code	SequenceAligninOntolgy
format	Entrada	Não	xsd:String	
Style	Entrada	Não	:raw_sequence_format	SequenceAligninOntolgy
fetchBatchReturn	Saída	Não	:fasta_format	SequenceAligninOntolgy

- **Read and Format Sequence:** este serviço é responsável por formatar as sequências obtidas para o serviço de alinhamento. Para tal tarefa, foi associado um serviço web para remoção de espaços e caracteres indesejáveis. A Figura 4.31 mostra a anotação semântica para o serviço e a Tabela 9 mostra a relação dos termos para os parâmetros;

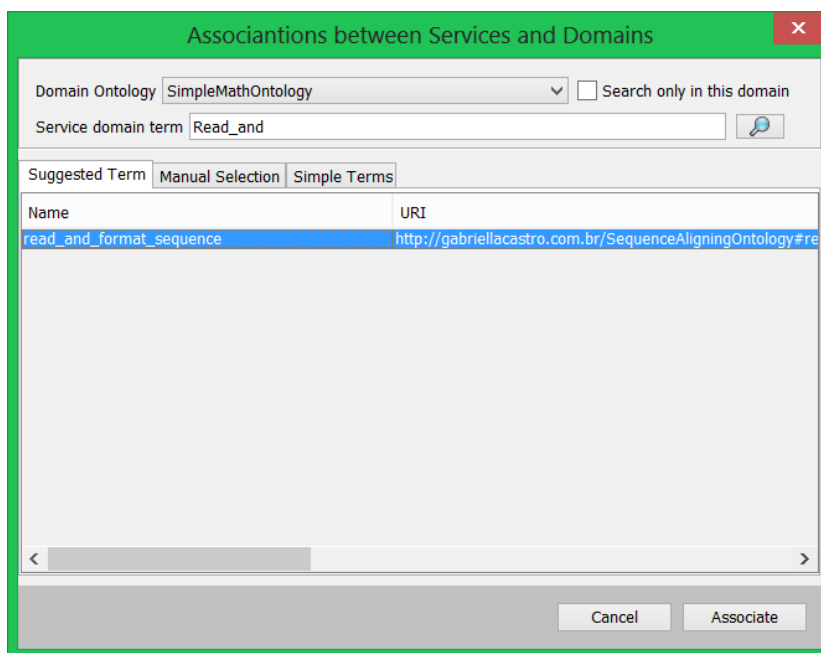


Figura 4.31: Anotação semântica para Read and Format Sequences

Tabela 9: Parâmetros de Read and Format Sequences

Parâmetro	Tipo	É Complexo?	Termo de Domínio	Ontologia de Domínio
sequences	Entrada	Não	:fasta_format	SequenceAligninOntolgy
separator	Entrada	Não	xsd:String	SequenceAligninOntolgy
Return	Saída	Não	:fasta_formar	SequenceAligninOntolgy

- *Alignment*: dentre os vários algoritmos para a realização de alinhamento de sequências, aqui foi selecionado o serviço web do *Clustal Omega*, devido ao fato de que o modelo proposto necessita de realizar o alinhamento de mais de uma sequência, portanto, este serviço foi selecionado. A anotação semântica relativa a este serviço no processo de inclusão pode ser vista na Figura 4.32, e seus parâmetros na Tabela 10.

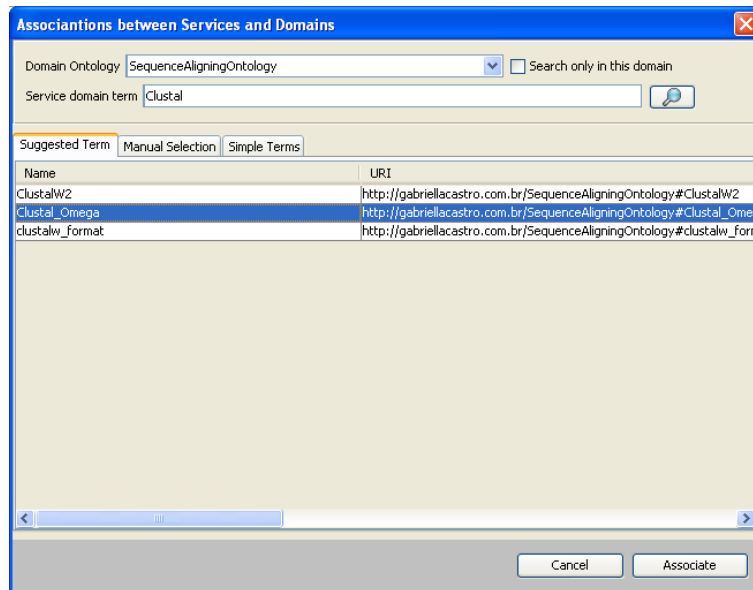


Figura 4.32: Anotação semântica para o serviço de Alignment

Tabela 10: Parâmetros de Clustal Omega

Parâmetro	Tipo	É Complexo?	Termo de Domínio	Ontologia de Domínio
Email	Entrada	Não	xsd:String	
Title	Entrada	Não	xsd:String	
Sequence	Entrada	Não	:fasta_format	SequenceAligninOntolgy
Outfmt	Entrada	Não	xsd:String	
Return	Saída	Não	xsd:String	

O serviço *Clustal Omega*, por depender de recursos computacionais mais que os outros, é disponibilizado como um serviço web assíncrono, ou seja, ele devolve um identificador ao usuário e fica executando o processo. O usuário é quem tem que verificar através de outros serviços, se o processo terminou e se obteve resultados. Sendo assim, existe a necessidade de inclusão de outros dois serviços para se comunicar com o *Clustal* por meio da categoria conectores, sendo eles:

- **Conector *getStatus* de *Clustal Omega*:** este conector é responsável por verificar se o serviço *clustal omega* terminou seu processo de execução. Ele realiza a requisição aos servidores trazendo de volta mensagens como: *RUNNING* para em execução, *FINISHED* para concluído e *NOT FOUND* para os processos não encontrados. A Figura 4.33 mostra a inclusão deste conector na e-ScienceNet.

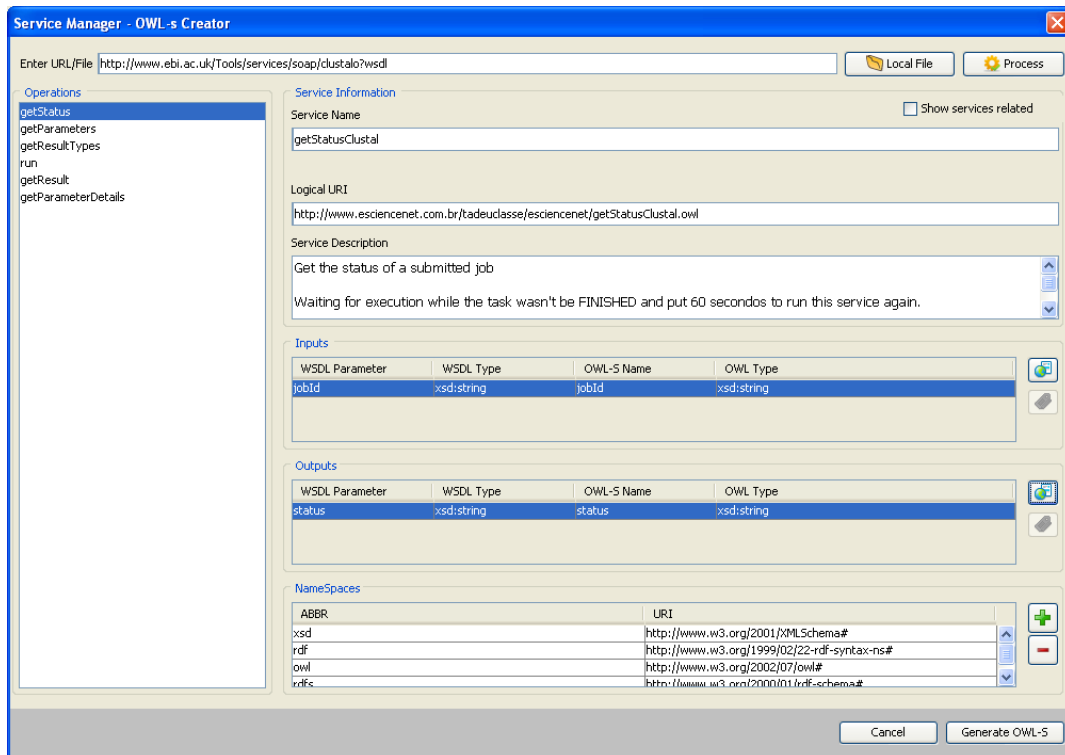


Figura 4.33: Inclusão do conector getStatus para Clustal Omega

- Conector *getResult* de *Clustal Omega*: este conector é o responsável por obter o resultado da execução de um alinhamento realizado pelo serviço *Clustal Omega*. Porém, o mesmo só deve ser executado após o conector *getStatus* retornar a mensagem *FINISHED*. Na Figura 4.34 é possível visualizar a *interface* de inclusão do conector *getStatus*.

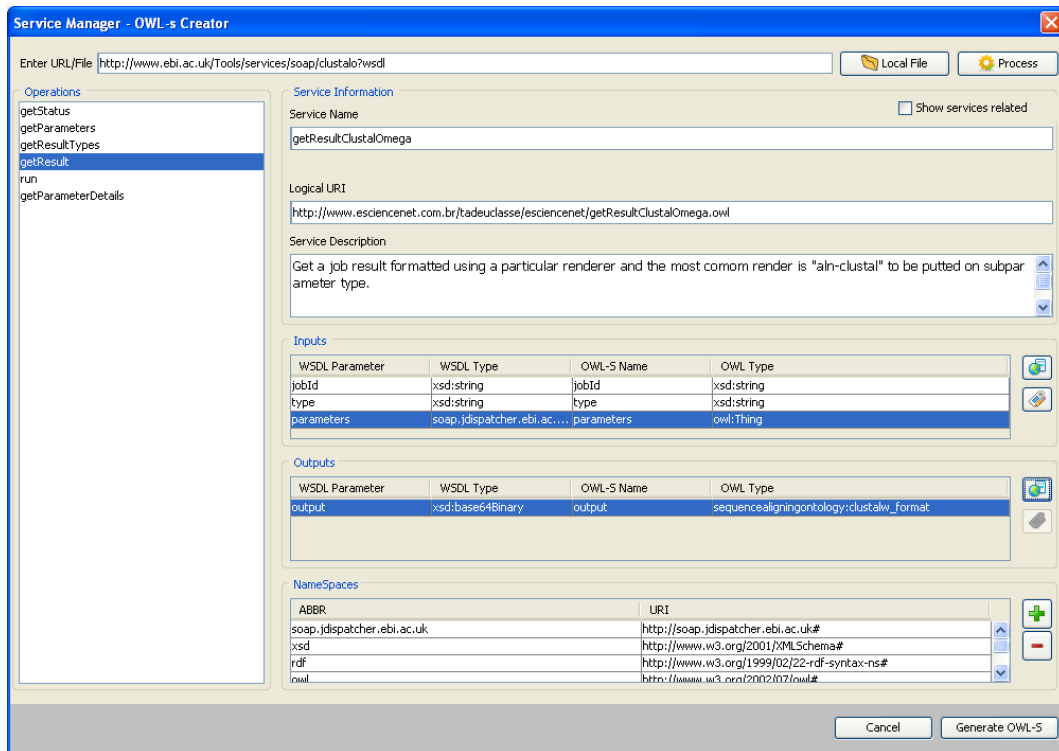


Figura 4.34: Inclusão do conector getResult para Clustal Omega.

Como já dito, a primeira etapa no desenvolvimento de uma composição na e-ScienceNet é a criação do modelo abstrato. Neste caso, a *interface* de desenvolvimento deste modelo carrega as ontologia de domínio e as tarefas relacionadas à composição do modelo. Com o conhecimento obtido na ontologia, é possível saber que as tarefas relacionadas pelo modelo proposto por DA SILVA, BRAGA e CAMPOS [2012] são *Get_Sequences*, *Read_and_Format_Sequences* e *Aligning*. Baseando-se nas regras de sequência de tarefas, também descritas na ontologia de domínio, o cientista consegue especificar o modelo abstrato para a pesquisa de serviços web. A Figura 4.35, apresenta o modelo abstrato especificado.

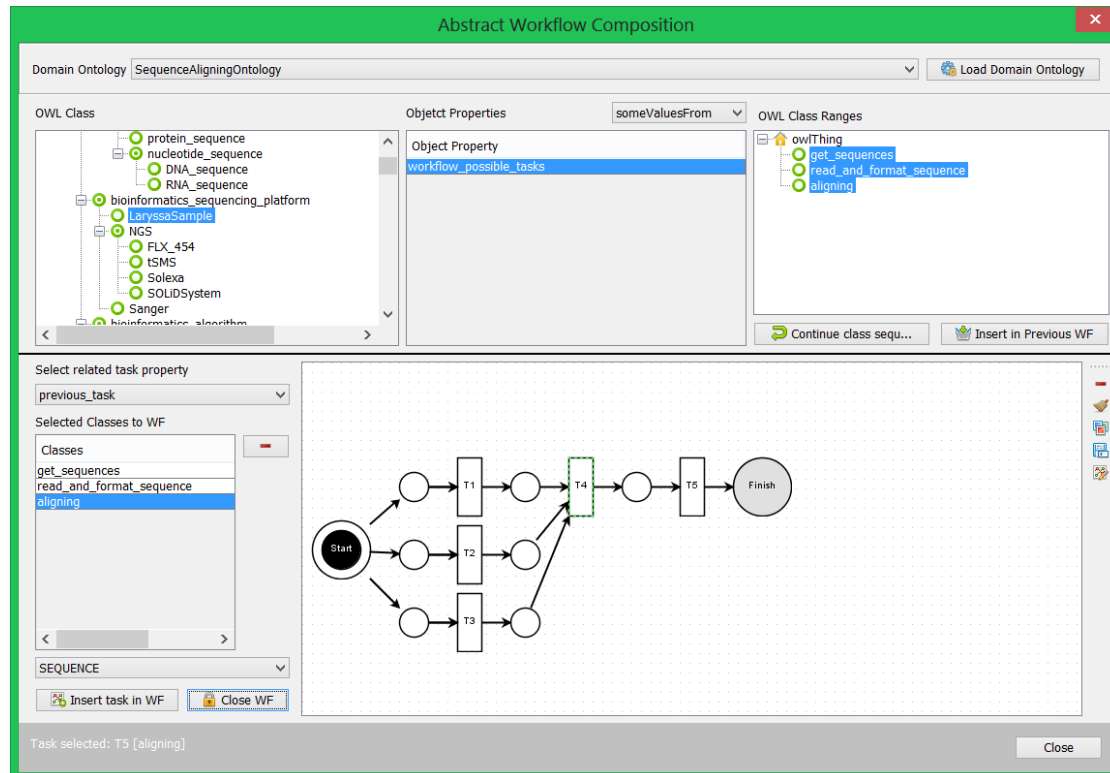


Figura 4.35: Inclusão da tarefa Aligning e finalização do modelo abstrato

É possível perceber, comparando a Figura 4.28 com a Figura 4.35, que o modelo abstrato gerado pela e-ScienceNet corresponde exatamente ao modelo proposto por DA SILVA, BRAGA e CAMPOS [2012].

Com a criação do modelo abstrato da composição, a etapa seguinte consiste na busca pelos serviços a fim de realizar a composição dos mesmos em um modelo funcional. Essa busca é realizada através do compartilhamento de informações entre os *peers* da comunidade semântica (Figura 4.36), os quais possuem o conhecimento sobre serviços web e disponibilizam as suas anotações semânticas tornando-os pesquisáveis para as composições a serem realizadas.

O Gerente de Serviço, baseando-se no modelo criado, realiza as pesquisas nos *peers* da e-ScienceNet a partir das tarefas, utilizando ou não máquinas de inferência para a descoberta de termos relacionados às tarefas propostas no modelo. Após isso, são pesquisados serviços web, realizando o alinhamento semântico e sintático dos serviços por meio das informações inferidas pelas ontologias OWL-SE. Por exemplo, a Figura 4.37, mostra as

inferências obtidas na ontologia de domínio *SequenceAligningOntology* para a realização da tarefa *aligning*, obtendo como resposta, *clustal_omega*, *clustalW*, *blast* e outras.

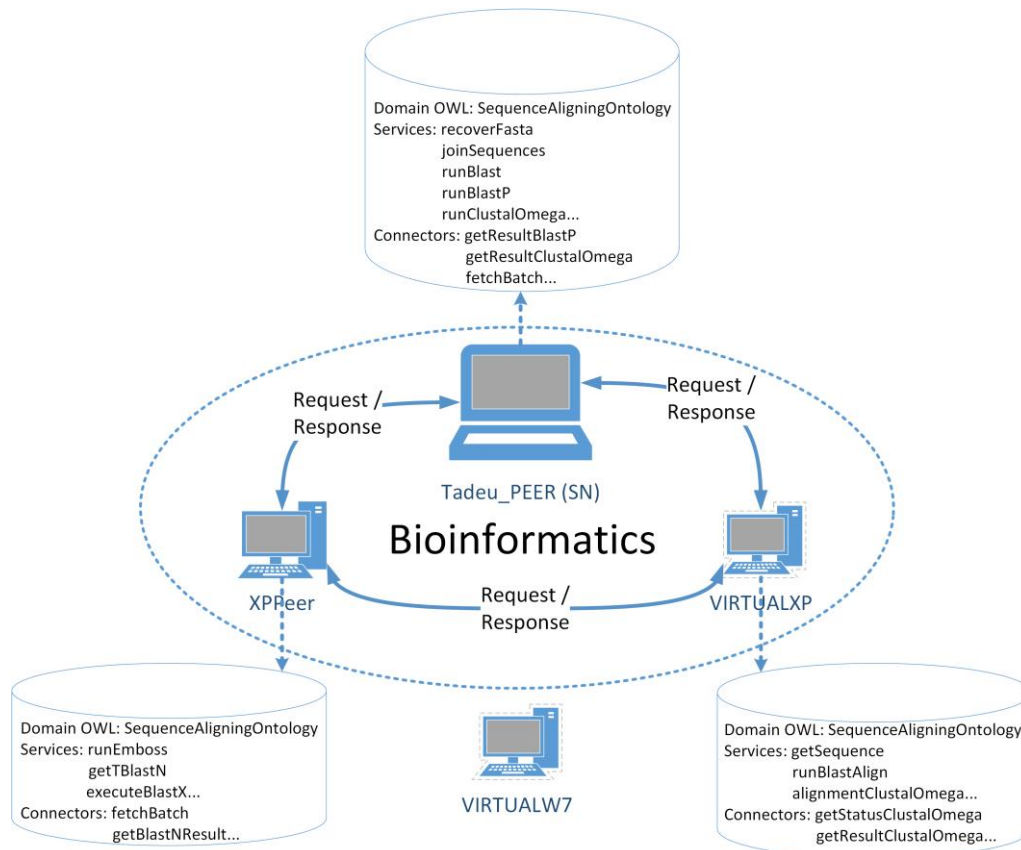


Figura 4.36: Ambiente colaborativo para Bioinformatics

By Abstract Workflow By Free Services

Related terms

TBLASTX - BLASTN - BLASTP - BLASTX - TBLASTN - BLAST_Basic_Local_Alignment_Search_Tool - Smith_Waterman - EMBOSS - Clustal_Omega - MultiAlign - ClustalW2

Figura 4.37: Inferência de serviços relacionadas com a tarefa *aligning*

Respectivamente as figuras 4.38, 4.39, 4.40, 4.41 e 4.42 mostram o processo de pesquisa dos serviços para as tarefas *getSequence*, *read and format sequence* e *aligning*, e os conectores *getStatus* e *getResult* da tarefa *aligning*.

The screenshot shows the 'Services Research' window with the search criteria 'get_sequences' and 'task_performed_by'. The interface is divided into several sections:

- Related terms:** FASTA
- Services Found:** Tadeu_PEER, RecoverFasta_2 - [FASTA], VirtualXP, and getSequence_2 - [FASTA].
- Selected Services:** A table listing tasks and their associated operations and domains.

Task	Operation	Domain Associated	Petri Op
get_sequences	GetSequence	FASTA	AND-JOIN
get_sequences	RecoverFasta_1	FASTA	AND-JOIN
- Parameters:** A table with columns for Parameter, Semantic Type, and Syntactic Type. It is currently empty.
- Connectors Results:** Empty.
- Connectors Associated:** Empty.
- Service Informations:**
 - Service Name:** GetSequence_2
 - Service Description:** Parameters: -database: to nucleotide: genbank; to protein: uniprotkb -ids: any fasta code -format: fasta -style: raw
 - Domain Term:** FASTA
 - Inputs:**
 - db [Semantic]:** http://gabriellacastro.com.br/SequenceAligningOntology#sequence_database - **[Syntactic]:** <http://www.w3.org/2001/XMLSchema#string>
 - format [Semantic]:** <http://www.w3.org/2001/XMLSchema#string> - **[Syntactic]:** <http://www.w3.org/2001/XMLSchema#string>
 - ids [Semantic]:** http://gabriellacastro.com.br/SequenceAligningOntology#fasta_code - **[Syntactic]:** <http://www.w3.org/2001/XMLSchema#string>

Selected Task: get_sequences. Buttons: Connect parameters, Close.

Figura 4.38: Busca dos serviços relacionados a tarefa *getSequence*

The screenshot shows the 'Services Research' window with the search criteria 'read_and_format_sequence' and 'task_performed_by'. The interface is divided into several sections:

- Related terms:** GroupSequences
- Services Found:** VirtualXP, Tadeu_PEER, and joinSequences - (GroupSequences). A message states: "The search don't had any results of Tadeu_PEER".
- Selected Services:** A table listing tasks and their associated operations and domains.

Task	Operation	Domain Associated	Petri Op
get_sequences	GetSequence	FASTA	AND-JOIN
get_sequences	RecoverFasta_1	FASTA	AND-JOIN
get_sequences	GetSequence_2	FASTA	AND-JOIN
- Parameters:** A table with columns for Parameter, Semantic Type, and Syntactic Type.

Parameter	Semantic Type	Syntactic Type
db	sequence_database	string
format	string	string
ids	fasta_code	string
style	raw_sequence_format	string
fetchBatchReturn	fasta_format	string
- Connectors Results:** Empty.
- Connectors Associated:** Empty.
- Service Informations:**
 - Service Name:** joinSequences
 - Service Description:** Auto generated from <http://192.168.13.128:8080/BioinformaticsWS/BioWebService?WSDL>
 - Domain Term:** GroupSequences
 - Inputs:**
 - separator [Semantic]:** <http://www.w3.org/2001/XMLSchema#string> - **[Syntactic]:** <http://www.w3.org/2001/XMLSchema#string>
 - sequences [Semantic]:** http://gabriellacastro.com.br/SequenceAligningOntology#fasta_format - **[Syntactic]:** <http://www.w3.org/2001/XMLSchema#string>
 - sequences [Semantic]:** http://gabriellacastro.com.br/SequenceAligningOntology#fasta_format - **[Syntactic]:** <http://www.w3.org/2001/XMLSchema#string>

Selected Task: read_and_format_sequence. Buttons: Connect parameters, Close.

Figura 4.39: Busca do serviço relacionado a tarefa *read and format sequences*

The screenshot shows the 'Services Research' window with the search term 'aligning'. The 'Selected Services' table is as follows:

Task	Operation	Domain Associated	Petri Op
get_sequences	GetSequence	FASTA	AND-JOIN
get_sequences	RecoverFasta_1	FASTA	AND-JOIN
get_sequences	GetSequence_2	FASTA	AND-JOIN
read_and_format_se...	joinSequences	GroupSequences	SEQUENCE

The 'Parameters' table is as follows:

Parameter	Semantic Type	Syntactic Type
separator	string	string
sequences	fasta_format	string
sequences	fasta_format	sequence
return	fasta_format	string

The 'Service Informations' section for 'AlignmentClustalOmega' includes:

- Service Name:** AlignmentClustalOmega
- Service Description:** This is a sequence align algorithm. To parameter: -outfmt: "fa" or "selex" or "clustal" -sequence: any fasta format Obs: is necessary to connect some connector getStatus and before a connector getResult.
- Domain Term:** Clustal_Omega
- Inputs:**
 - email [Semantic]:** <http://www.w3.org/2001/XMLSchema#string> - **[Syntactic]:** <http://www.w3.org/2001/XMLSchema#string>
 - title [Semantic]:** <http://www.w3.org/2001/XMLSchema#string> - **[Syntactic]:** <http://www.w3.org/2001/XMLSchema#string>

Figura 4.40: Busca de serviço relacionado a tarefa *aligning*

The screenshot shows the 'Services Research' window with the search term 'aligning'. The 'Selected Services' table is as follows:

Task	Operation	Domain Associated	Petri Op
get_sequences	GetSequence	FASTA	AND-JOIN
get_sequences	RecoverFasta_1	FASTA	AND-JOIN
get_sequences	GetSequence_2	FASTA	AND-JOIN
read_and_format_se...	joinSequences	GroupSequences	SEQUENCE
aligning	AlignmentClustalOmega	Clustal_Omega	

The 'Parameters' table is as follows:

Parameter	Semantic Type	Syntactic Type
email	string	string
title	string	string
outfmt	string	string
sequence	fasta_format	string
status	string	string

The 'Connectors Associated' table is as follows:

Task	Operation	Association	Service URL
AlignmentClustalOme...	getStatusClustalOmega	jobId == jobId	http://www.ebi.ac.uk

The 'Service Informations' section for 'getStatusClustalOmega' includes:

- Connector Name:** getStatusClustalOmega
- Connector Description:** Get the status of a submitted job Put a loop while output wasn't FINISHED and retrieve again at 60 seconds.
- Inputs:**
 - jobId [Semantic]:** <http://www.w3.org/2001/XMLSchema#string> - **[Syntactic]:** <http://www.w3.org/2001/XMLSchema#string>
- Outputs:**
 - status [Semantic]:** <http://www.w3.org/2001/XMLSchema#string> - **[Syntactic]:** <http://www.w3.org/2001/XMLSchema#string>

Figura 4.41: Busca do conector *getStatus* para a tarefa *aligning*

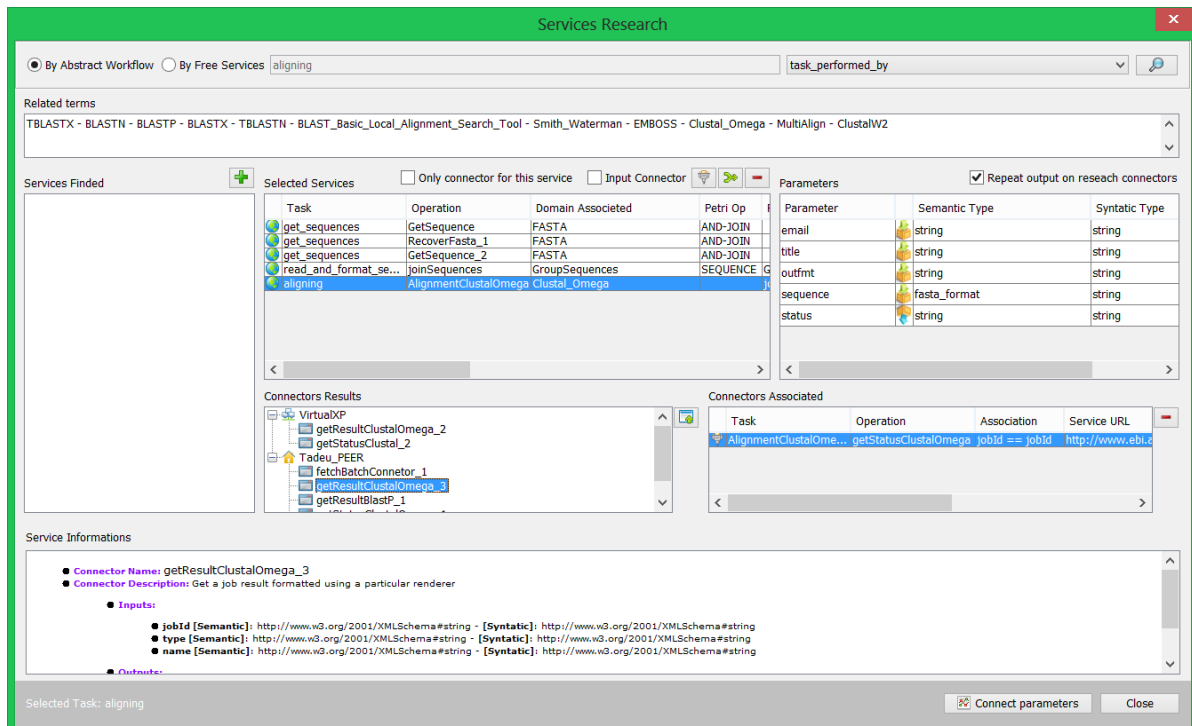


Figura 4.42: Busca do conector *getResult* para a tarefa *aligning*

Com a busca por serviços finalizada, o cientista deve especificar a ligação entre os parâmetros dos serviços relacionados para a criação do modelo funcional. A Tabela 11 mostra a conexão entre os parâmetros e a Figura 4.43 mostra a configuração de repetição para o conector *getStatus* do serviço *Clustal Omega*.

Tabela 11: Conexão entre os parâmetros do serviço

Serviço / Tarefa	Parâmetro	Tipo	Manual	Valor
getSequence / get_sequences	Db	Entrada	X	genbank
	Format	Entrada	X	fasta
	Ids	Entrada	X	J00124
	Style	Entrada	X	raw
	fetchBatchReturn	Saída		joinSequences:sequences
getSequence_1 / get_sequences	Db	Entrada	X	genbank
	Format	Entrada	X	fasta
	Ids	Entrada	X	J00145
	Style	Entrada	X	raw
	fetchBatchReturn	Saída		joinSequences:sequences
getSequence_2 / get_sequences	Db	Entrada	X	genbank
	Format	Entrada	X	fasta
	Ids	Entrada	X	J00189
	Style	Entrada	X	raw
	fetchBatchReturn	Saída		joinSequences:sequences
joinSequences /	sequences	Entrada		getSequence:fetchBatchReturn

read_and_ format_sequences				getSequence_1:fetchBatchReturn getSequence_2:fetchBatchReturn
	separator	Entrada	X	\n
	Retun	Saída		AlinhamentoClustalOmega:sequen ce
AlinhamentoClustalOme ga / aligning	Email	Entrada	X	tadeu.classe@gmail.com
	Title	Entrada	X	Clustal Omega
	Sequence	Entrada		joinSequences:return
	Outfmt	Entrada	X	fa
	jobId	Saída		getStatusClustalO:jobId
getStatusClustalO / Conector de aligning	jobId	Entrada		AlinhamentoClustalOmega:jobId
	Status	Saída		
getResultClustalO/ Conector de aligning	jobId	Entrada		AlinhamentoClustalOmega:jobId
	Type	Entrada	X	aln-clustal
	Output	Saída		

Figura 4.43: Inclusão estrutura de repetição no conector getStatus

Após o término das configurações dos parâmetros é possível que a composição seja executada. É possível perceber na Figura 4.44 que todos os serviços web foram executados gerando no campo *workflow* Information os dados gerados pelo alinhamento das sequências. Este resultado, assim como o modelo funcional pode ser salvo e compartilhado como recursos pesquisáveis para os demais *peers* da comunidade semântica.

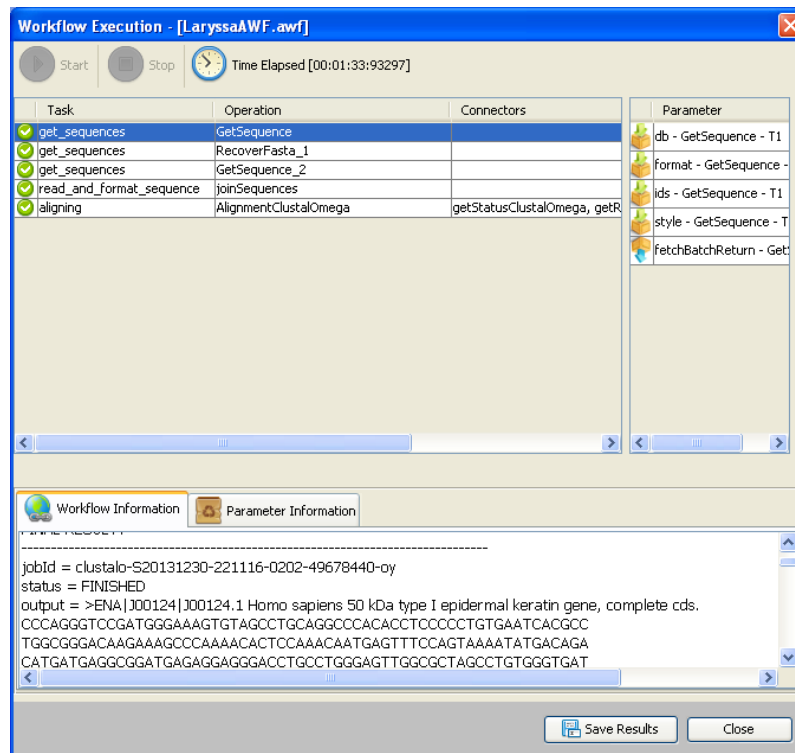


Figura 4.44: Resultado final do workflow gerado pela proposta de DA SILVA, BRAGA e CAMPOS [2012]

Esta prova de conceito teve como base o desenvolvimento de um experimento, envolvendo o domínio de sequenciamento/alinhamento genético, e sendo mais complexo que o primeiro cenário por se tratar da composição de serviços com várias opções de entradas e saídas de resultados. Nesta prova de conceito, foi possível obter indícios da possibilidade de se desenvolver experimentos científicos em domínios mais complexos, detalhando os passos de criação do modelo abstrato da composição, baseado nas regras da ontologia de domínio, as pesquisas de serviços web semânticos e a execução do experimento, mostrando os resultados de um alinhamento entre sequências genéticas. No contexto de laboratórios colaborativos, essa prova de conceito demonstrou a possibilidade de compartilhamento de recursos entre os integrantes de comunidades semânticas, destacando a descoberta de serviços web entre os peers, e o compartilhamento dos resultados obtidos.

4.4 PROVA DE CONCEITO 3

Levando em consideração o domínio de pesquisa abordada na prova de conceito 2 e o propósito das Provas de Conceito aqui apresentadas, que é o de obter indícios sobre a viabilidade da proposta, esta Prova de Conceito detalha o outro método de realização de busca de serviços na e-ScienceNet. Apresentamos a seguir o método manual de busca por serviços, ou seja, sem que haja a necessidade de desenvolvimento de um modelo abstrato para guiar as pesquisas por serviços web, em um cenário colaborativo entre os pesquisadores.

Imaginando um ambiente semelhante ao apresentado pela Prova de Conceito 2, nesta PoC é apresentada a possibilidade de obter toda a informação possível sobre uma única amostra genética, um pesquisador necessita da criação de um experimento científico para alinhá-la e obter resultados. Diferente da PoC 2, imagina-se que este cientista tem uma única amostra e a partir dela, ele deseja descobrir quais são as sequências similares. E em um segundo momento, a partir das similaridades obtidas, caso o cientista queira submetê-las em algoritmos de sequenciamentos múltiplos para ampliar os resultados do alinhamento.

Entretanto, nesta Prova de Conceito, oposto à PoC2, o pesquisador, conhecendo quais são os algoritmos, serviços e os termos necessários para a composição de seu experimento, opta pela não criação do modelo abstrato de composição e decide realizar a pesquisa manual de serviços entre os *peers* da e-ScienceNet.

O objetivo é demonstrar a viabilidade de composição de serviços usando a pesquisa e descoberta manual de serviços web na e-ScienceNet, a fim de desenvolver um modelo funcional capaz de obter uma sequência biológica (nucleotídeo ou proteína) e submeter o resultado obtido em: 1º: algum algoritmo para alinhamento de uma única sequência (*Blast* ou outro), responsável por alinhá-la com outras sequências retornar os resultados similares; 2º: a partir dos resultados obtidos no primeiro alinhamento, deseja-se submetê-los em dois outros serviços para alinhamentos múltiplos de

sequências (*Clustal* e *EMBOSS*), a fim de ampliar as informações a serem obtidas sobre a sequência inicial e as demais.

Para demonstrar a viabilidade da criação de tal experimento, foi desenvolvido um *workflow* utilizando a ferramenta de gerenciamento de *workflow* Taverna [TAVERNA, 2014], pelo Núcleo de Engenharia do Conhecimento (NeNC) [NENC, 2014] da Universidade Federal de Juiz de Fora (UFJF). O *workflow* gerado por essa ferramenta é apresentado na Figura 4.45.

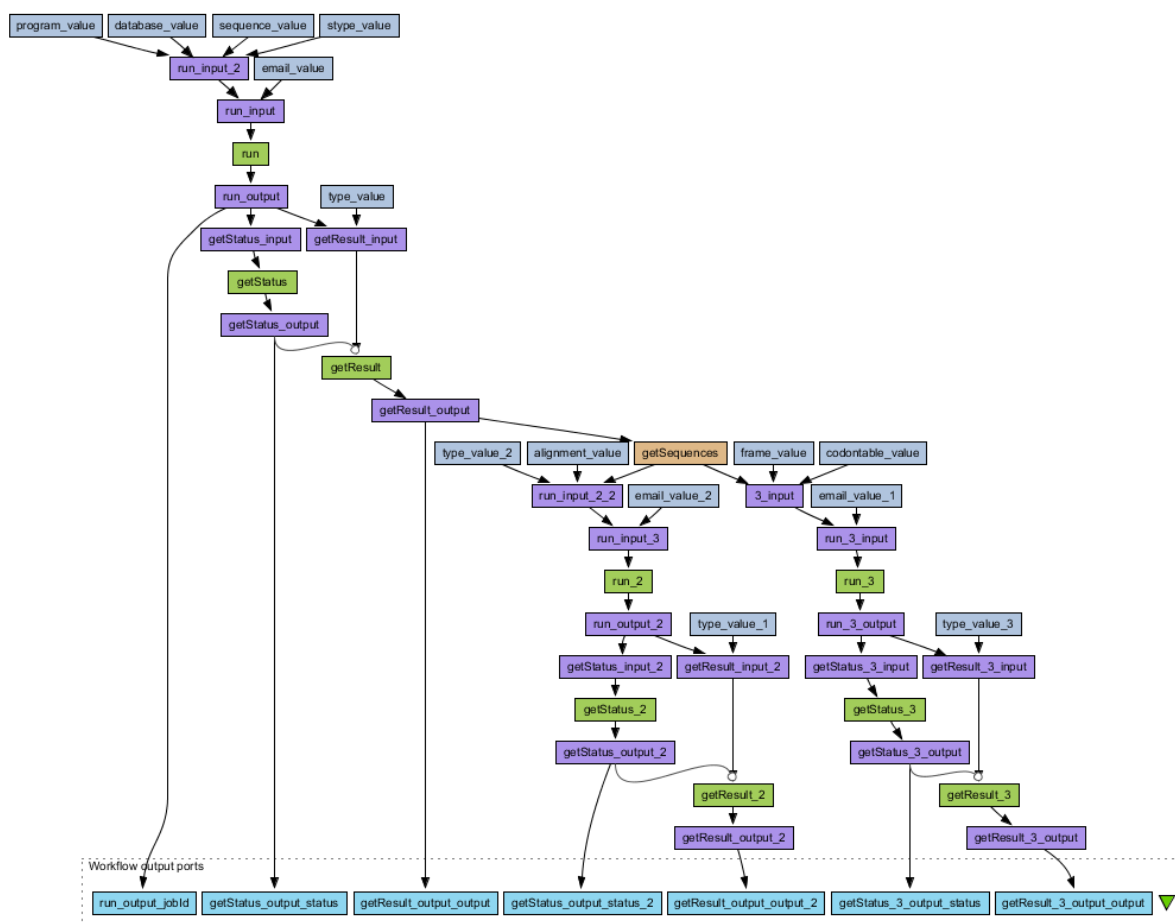


Figura 4.45: Workflow Blast, Clustal, EMBOSS gerado pelo Taverna

É possível perceber que não se trata de um *workflow* muito simples com a execução sequencial de serviços. Nele existe a separação e união de diversos processos em diferentes estágios. Outra informação importante, é que é possível notar a presença de serviços como *getStatus* e *getResult*. Conforme exposto na seção 4.3, também aqui existem serviços assíncronos para executarem determinadas tarefas. A execução dos serviços web, *Blast*, *Clustal* e *EMBOSS* necessitam desses serviços a fim de obter seu resultado final. Na

e-ScienceNet, estes serviços secundários são retratados como os conectores do serviço principal. Além disso, os serviços de alinhamento só são executados quando existem sequências como parâmetros de entrada, assim, há a necessidade de inserir um parâmetro de entrada ao primeiro serviço para a busca da sequência a partir de um identificador. Com a execução do *workflow* no Taverna, foi verificado, ainda, que o serviço *blast*, assim como *clustal* e *emboss*, retorna somente os identificadores dos alinhamentos, portanto houve a necessidade de inserir um serviço que recupere todos as sequências para esses identificadores no intuito de serem passados às próximas tarefas. Assim, foi incluso um serviço conector para organizar os códigos, bem como recuperar as sequências.

Desse modo, nessa PoC, o mesmo *workflow* desenvolvido no Taverna foi especificado na e-ScienceNet, com o objetivo de verificar a viabilidade de desenvolvimento de *workflows* especificados em Sistemas de Gerenciamento de Workflows Científicos, como é o caso do Taverna.

Como dito acima, essa PoC retrata a pesquisa de serviços para a criação de um *workflow* sem o uso de modelos abstratos pelo Gerente de Pesquisa. Porém, para que fosse possível guiar-se no processo de busca, um mapeamento para os serviços foi gerado o qual pode ser visto na Figura 4.46.

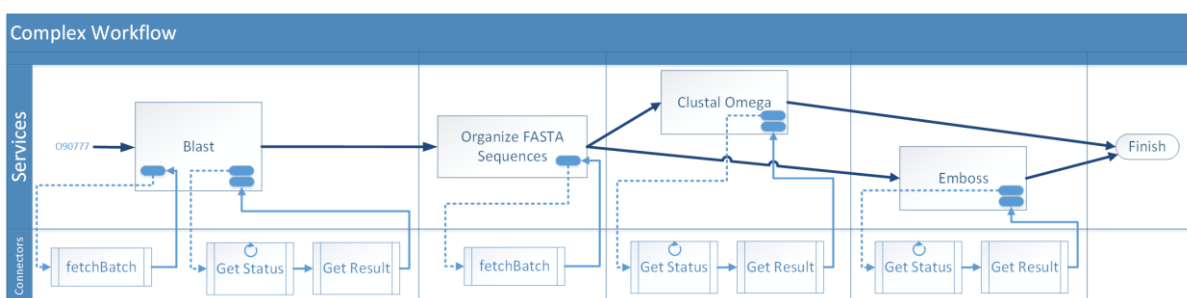


Figura 4.46: Modelo de conexão de web services

Considerando aqui que todos os serviços e seus conectores já estão disponíveis e compartilhados pelos *peers* na comunidade semântica *Bioinformatics*, é possível dar início as pesquisas de serviços. Este nível de pesquisa requer o conhecimento de quais serviços o cientista deseja utilizar, pois ele deve inserir o seu nome (ou parte dele) no campo de pesquisa disponível. Como na pesquisa por modelo abstrato, são aqui também inferidos

termos equivalentes e similares nas ontologias de domínio relativos ao conteúdo digitado pelo pesquisador. O restante do processo ocorre da mesma forma, com as variações dos termos gerados, o sistema realiza a pesquisa nos *peers*, comparando semântica e sintaticamente os parâmetros dos serviços. Para ilustrar, as Figuras 4.47 e 4.48 mostram respectivamente a busca pelo serviço *blast* e o conector de entrada de *blast* para recuperar a sequência a partir de um identificador.

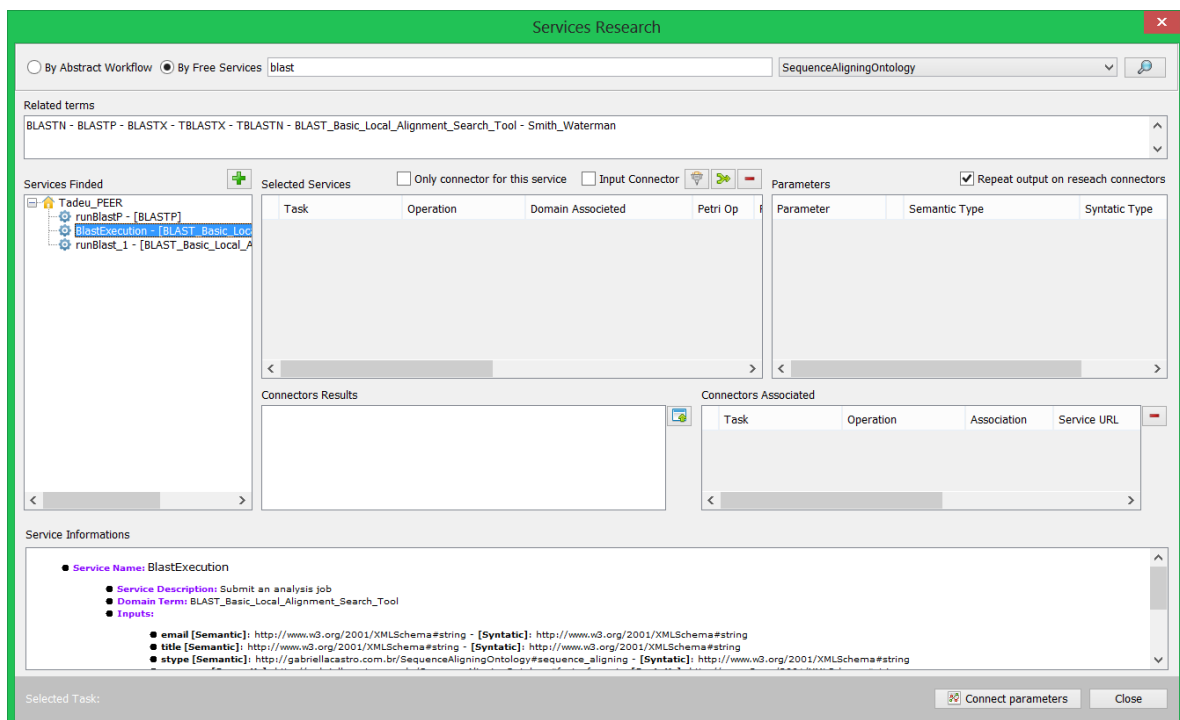


Figura 4.47: Pesquisa para o serviço blast

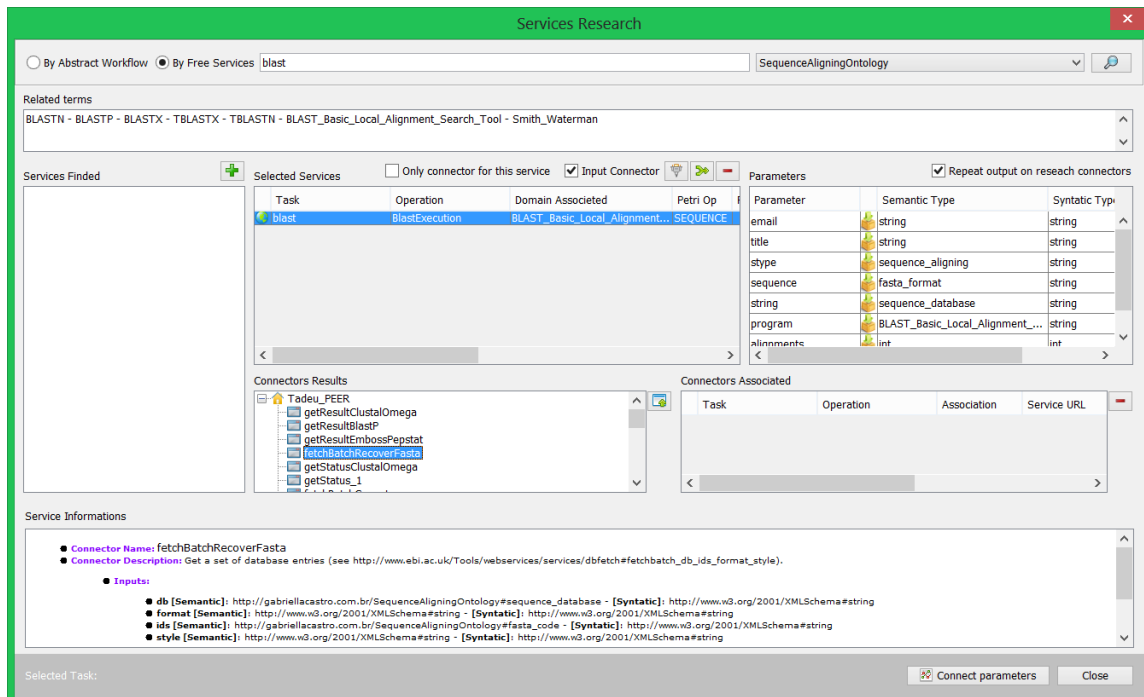


Figura 4.48: Conector de entrada para o serviço blast (fetchBatch)

Após a pesquisa, é apresentada a tela para conexão entre os parâmetros dos serviços e a configuração de conectores e estruturas de repetição, gerando, assim, o modelo funcional que pode ser compartilhado com os demais *peers* (Figura 4.49). Assim, a composição de serviço foi executada pela e-ScienceNet (Figura 4.50), exibindo os resultados dos algoritmos de alinhamento de seqüências.

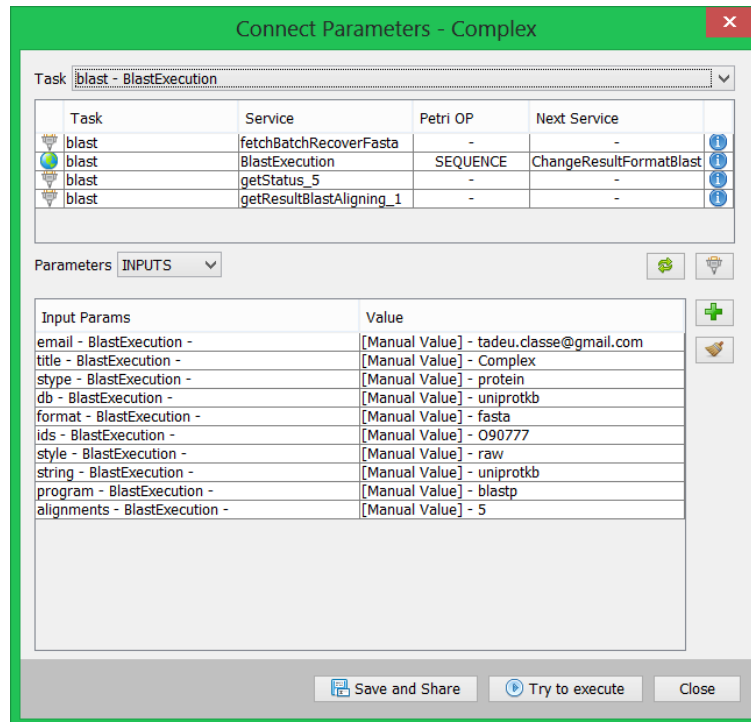


Figura 4.49: Conexão e inclusão de valores nos parâmetros dos serviços

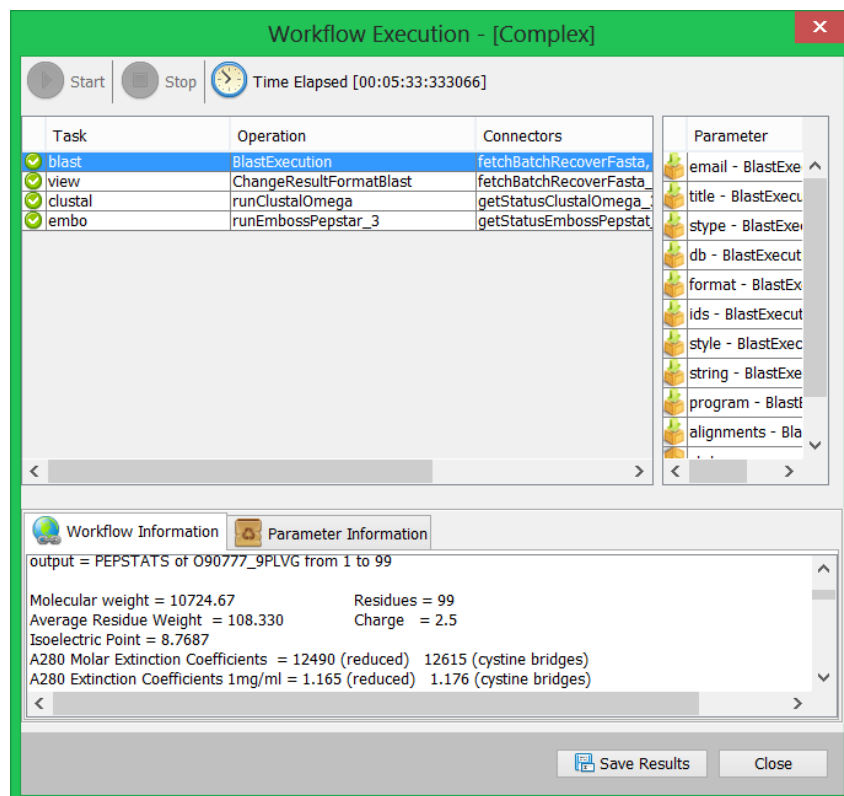


Figura 4.50: Resultado final de Blast, Clustal e Emboss

Nesse sentido, o objetivo desta PoC foi demonstrar a viabilidade de um pesquisador que tenha conhecimento suficiente sobre o domínio, realizar a composição de um experimento a partir de buscas de serviços web semânticos utilizando a busca manual fornecida pela e-ScienceNet. Neste experimento, foi considerado que o cientista possui o conhecimento sobre os termos, tarefas e serviços do domínio, necessário a geração da composição. Além disso, essa prova de conceito demonstrou a possibilidade de composição de serviços científicos de um *workflow* de sequenciamento múltiplo já utilizado pelo NeNC [NENC, 2014] e especificado no SGWC Taverna. Esta composição é formada por um grande número de serviços e conectores. Assim, foi possível obter indícios de que existe a possibilidade de realizar a composição manual de um experimento na e-ScienceNet, desde que o pesquisador tenha o conhecimento de quais são os serviços necessários para a sua composição.

4.5 ANÁLISE DAS PROVAS DE CONCEITO

Analisando as provas de conceitos apresentadas neste capítulo, foi possível obter indícios de que o uso da e-ScienceNet pode auxiliar os cientistas no processo de especificação, construção e execução de experimentos em uma configuração de laboratório colaborativo, por meio de suas funcionalidades, como a composição de serviços web, por exemplo.

Considerando os Sistemas de Gerenciamento de Workflows Científicos (SGWC), tais como Taverna [Taverna, 2014], Kepler [Kepler, 2014], Vistrails [Vistrails, 2014] e outros, a e-ScienceNet tem como diferencial ser um sistema que se baseia em rede ponto a ponto para compartilhar recursos científicos, e com isso, compor experimentos científicos através da descoberta de serviços web semânticos que estejam disponíveis na rede. Outra característica é a semântica, uma vez que utiliza ontologias no processo de modelagem da composição e na busca pelos serviços web, auxiliando o cientista na montagem e execução de seu experimento. No entanto, a e-ScienceNet é ainda um protótipo e por isso não tem a robustez de um SGWCs. Por exemplo, na e-ScienceNet não existe ainda um *interface* gráfica dinâmica onde o usuário possa conectar os parâmetros dos serviços apenas arrastando os pontos de

conexão; ou a atual impossibilidade de inclusão de Scripts, no qual o usuário pode construí-los e incorporá-los como serviços a seu *workflow*, além de características relacionadas ao desempenho de execução.

Desta maneira, é possível dizer foram obtidos indícios da viabilidade do uso da e-ScienceNet para auxiliar na construção de experimentos científicos baseado no conceito de compartilhamento de recursos e resultados previstos pelos laboratórios colaborativos. Este indícios foram demonstrados por meio das provas de conceitos apresentadas, as quais detalharam a possibilidade de especificar experimentos e executá-los.

4.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Visando obter indícios sobre a viabilidade da utilização da abordagem proposta pela e-ScienceNet para prover suporte ao desenvolvimento de experimentos e ao compartilhamento de resultados entre os cientistas baseados nos conceitos de laboratórios colaborativos, foram especificadas algumas provas de conceito.

Nestas provas de conceito, alguns cenários foram apresentados no intuito de demonstrar a possibilidade do uso da arquitetura para ambientes científicos. Com este cenário, foi possível detalhar o uso das funcionalidades propostas pela arquitetura, como por exemplo: a criação de modelos de experimentos científicos, compondo-os com serviços web semânticos e, em alguns casos, executando modelos funcionais das composições. Outra característica importante apresentada foi a utilização de ontologias de domínio, auxiliando os usuários na construção dos modelos e sua utilização na busca por serviços web, através de regras, restrições e termos existentes nas ontologias.

Entretanto, essas provas de conceito demonstram possíveis cenários de uso da e-ScienceNet, no intuito de apresentar o seu funcionamento. Cabe ressaltar que os cenários aqui tratados não foram usados em ambientes reais de pesquisa, e nem por cientistas dos domínios descritos.

Outra questão a ser ressaltada é que caso não haja ontologias de domínio nas comunidades semânticas, algumas funcionalidades não serão realizadas de maneira adequada como, por exemplo, a anotação semântica de

serviços e os seus parâmetros, ou a pesquisa por termos relativos a alguma tarefa. Outra falha que contempla a inexistência de ontologias de domínio seria que o cientista obrigatoriamente necessita saber sobre todos os algoritmos necessários para a composição do seu experimento.

Caso uma composição venha a falhar, devido à inatividade momentânea de algum serviço, a e-ScienceNet não apresenta mecanismos de recuperação, ou seja, o experimento simplesmente para sem retornar os resultados esperados pelo usuário.

Mesmo assim, a partir dessas provas de conceito acredita-se que haja indícios sobre a viabilidade de se utilizar a e-ScienceNet e sua funcionalidade no apoio aos cientistas na construção e especificação de experimentos científicos.

5 CONCLUSÕES E TRABALHOS FUTUROS

A hipótese levantada neste trabalho é "a especificação de uma infraestrutura distribuída, baseada na tecnologia de redes ponto a ponto e web semântica, pode facilitar o desenvolvimento de experimentos científicos colaborativos".

A necessidade do uso de tecnologias computacionais avançadas para o processamento de informações para a criação de modelos científicos que auxiliem no entendimento e desenvolvimento de pesquisas vem crescendo cada vez mais. Tecnologias de processamento distribuído como redes ponto a ponto e grades computacionais (grid) são apontadas em muitas pesquisas como possíveis bases para a criação de aplicações voltadas para o contexto da *e-Science*, devido a sua capacidade de crescimento e compartilhamento de recursos. Essas aplicações científicas devem ser capazes de armazenar e processar um grande volume de dados distribuídos, separados geograficamente em diferentes bases. Na e-ScienceNet, estes dados são guardados e processados por cada um dos peers existentes na comunidade semântica, gerando a distribuição geográfica de recursos necessária. Além disso, é imprescindível que este tipo de abordagem forneça mecanismos de fácil utilização, no intuito de apoiar a especificação de experimentos científicos, pois muitos cientistas não possuem conhecimento computacional necessário para o desenvolvimento e composição de aplicações científicas. Neste contexto, existe uma enorme necessidade de que tais sistemas forneçam algum suporte semântico, no sentido de ajudar esses usuários na especificação de modelos de experimentos voltados a domínios científicos, estes, representados por *workflows*.

A abordagem aqui apresentada pela e-ScienceNet faz o uso das tecnologias de redes ponto a ponto, em conjunto com tecnologias da web semântica como ontologias, motores de inferência e serviços descritos por termos ontológicos, no intuito de fornecer uma plataforma onde os pesquisadores possam criar, compartilhar, reutilizar e executar modelos experimentais e seus resultados de maneira facilitada.

Considerando a hipótese levantada neste trabalho, em conjunto com a implementação de um protótipo e as provas de conceito, foi possível obter indícios de que é possível que composições colaborativas de aplicações científicas, que podem ser desenvolvidas utilizando a abordagem proposta nesta dissertação. Estas composições são baseadas no desenvolvimento de modelos semânticos, os quais podem ser utilizados como base para as buscas de recursos entre os nós da rede, no intuito de ajudar o pesquisador na construção do experimento.

Baseando nos objetivos destacados na introdução, é possível dizer que as contribuições deste trabalho são:

- i. ***Propor uma arquitetura de redes ponto a ponto semântica para fornecer suporte a e-Science.*** Este objetivo foi atendido devido à especificação e desenvolvimento do protótipo da e-ScienceNet. A implementação conta com mecanismos semânticos, os quais fazem uso de motores de inferências, ontologias e linguagens de consulta.
- ii. ***Especificar experimentos científicos distribuídos.*** Através da e-ScienceNet, foi possível demonstrar a viabilidade de desenvolvimento de experimentos com auxílio de outros peers existentes na rede, os quais compartilharam o conhecimento sobre serviços web no intuito de realizar uma composição de serviços.
- iii. ***Implementar mecanismos de pesquisa por serviços web baseados em modelos semânticos.*** Este objetivo foi concluído com os mecanismos de buscas desenvolvidos no gerente de semântica, composição e serviço. Estes mecanismos realizam pesquisas em serviços web, anotados semanticamente através de consultas SPARQL, e após isso, é realizada a sua comparação para verificação de compatibilidade;
- iv. ***Possibilitar a execução de experimentos científicos.*** Apesar de simples, o protótipo foi capaz de demonstrar uma interface para a execução de experimentos, permitindo, também, que seus resultados e modelos fossem executados e compartilhados;

- v. ***Fornecer resultados e compartilhar modelos que possam ser adaptados e reutilizados pelos pesquisadores em outros experimentos.*** Este objetivo foi alcançado na medida em que os usuários compartilhassem seus modelos de composições. Estes modelos são então usados para reprodução do experimento ou sua reutilização em outros processos. Além disso, com a execução de uma composição, todos os resultados podem ser compartilhados para os outros pesquisadores.

Alguns trabalhos futuros são destacados abaixo: Algumas sugestões para trabalhos futuros são destacadas abaixo:

- A composição de serviços especificada na e-ScienceNet é uma proposta inicial e, portanto, mecanismos como controle de falhas, entre outros não foram especificados. Uma evolução da proposta seria a proposição de um mecanismo de composição mais robusto, onde questões específicas relacionadas a controle de processos fossem propostas.
- Apesar da e-ScienceNet permitir a execução das composições em seu contexto, a proposta da arquitetura não é ser um Sistema de Gerenciamento de Workflow Científico (SGWC). Assim, uma evolução necessária da abordagem é sua possível integração com um SGWC que permita a execução do experimento de maneira distribuída.
- A proveniência de informações sobre as composições de serviços é uma necessidade que precisa ser desenvolvida em trabalhos futuros, para que os cientistas tenham informações concisas de como reproduzir tais experimentos em outros locais.

Além disso, algumas melhorias pontuais merecem destaque:

- Uma possível evolução da *PeerOntology* com objetivo de proporcionar maior número de informações sobre os *peers*, como: pesquisas em andamento e nível de contribuição na comunidade semântica. Novas

classes e restrições também podem ser propostas com o intuito de melhorar o processamento de conhecimento implícito na ontologia;

- Atualmente, um grupo de interesse é selecionado manualmente por um *peer*. Neste contexto, existe a possibilidade de criação de mecanismos de classificação automática dos *peers* em comunidades semânticas;
- Atualmente somente serviços baseados em SOAP podem ser inclusos na e-ScienceNet, assim é necessária a criação de mecanismos para a inclusão de serviços web baseados em outras tecnologias;
- É necessário realizar um aprimoramento entre a conexão de parâmetros de serviços, pois essa conexão é realizada de maneira semi-automatizada, ou seja, atualmente, os possíveis parâmetros do serviço são sugeridos para o usuário, o qual faz a sua seleção;
- Especificar um modelo gráfico na *interface* de execução da composição para melhorar o *feedback* com o usuário;

Contudo, acredita-se que este trabalho tenha atingido os seus objetivos, fornecendo uma abordagem colaborativa onde os cientistas possam compartilhar suas pesquisas, obter recursos e construir experimentos de maneira menos complexa. É possível que esta dissertação apresente idéias que possam ser trabalhadas não só para o desenvolvimento de aplicações para a área de bioinformática, mas para a composição de experimentos em diferentes domínios como, física, matemática, química e outros. Outro uso interessante seria no contexto de educação, onde os usuários poderiam ter uma rede de colaboração, onde compartilhariam recursos e criariam fluxos de estudos baseados na composição e no uso de ontologias de domínio.

REFERÊNCIAS

BASILI et al. GQL **Paradigm. Computer Encyclopedia of Software Engineering**, John Wiley & Sons, 1994.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The Semantic Web. **Scientific American**, v. 284 n. 5, p. 28-37, 2001.

BIANCHINI, D.; De ANTONELLIS, V.; MELCHIORI, M. P2P-SDSD: On-the-fly Service-based Collaboration in Distributed Systems. **International Journal of Metadata, Semantics and Ontologies**, v. 5, n. 3, p. 222-237, 2010.

BIBTEX. **Your bibTeX resource**. BibTeX, 2014. Disponível em: <<http://www.bibtex.org/>>. Acesso em: 02 de Fevereiro de 2014.

BLAKE, M.; TAN, W.; ROSENBERG, F. Composition as a service [web-scale workflow]. **IEEE Internet Computing**, v. 14, n. 1, p. 78-82, 2010.

BONIFACIO, M. et al. Kex: a peer to peer solution for distributed knowledge management. **Practical Aspects of Knowledge Management**, Springer, p. 490-500, 2002.

BOUHINI, C. et al. Discovery and selection of web services fragments for re-composition. In: **IEEE International Conference on Service-Oriented Computing and Applications (SOCA)**, p. 1-4, 2010.

BROEKSTRA, J.; KAMPMAN, A. SeRQL: a second generation RDF query language. **Proc. SWAD-Europe Workshop on Semantic Web Storage and Retrieval**, p. 13-14, 2003.

CARVALHO, M. C. D. C. G. D.; SILVA, D. C. G. D. Sequenciamento de DNA de nova geração e suas aplicações na genômica de plantas. **Ciência Rural**, v. 40, n. 3, p. 735-744, 2010.

CAVANAUGH, E. Web services: Benefits, challenges, and a unique, visual development solution. **Product Marketing Manager**, Altova WhitePaper, 2006.

CHIRITA, P. et al. Designing Semantic Publish/Subscribe Networks Using Super-Peers. In: **Semantic Web and Peer to peer**, Springer, p. 159-179, 2006.

CLASSE, T. M.; BRAGA, R. M. M.; CAMPOS, F.; DAVID, J. M. N. A Semantic Peer to peer Network For Service Composition In Scientific Domains. In: Conference on ENTERprise Information Systems (CENTERIS'2013), 2013, Lisboa. **Book of industry papers, poster papers and abstracts of the CENTERIS 2013**, Lisboa: SciKA, p. 215-225, 2013a.

CLASSE, T. M.; BRAGA, R. M. M.; Campos, F.; DAVID, J. M. N. e-ScienceNet: Service Composition in Scientific Domains. In: IADIS International Conference WWW/INTERNET (ICWI 2013), 2013, Forth Worth. **Proceedings of The IADIS International Conference WWW/Internet 2013 (ICWI'2013)**, p. 195-202, 2013b.

COSTA, G. et al. PL-Science: Using Ontologies and Feature Models in Scientific Software Product Line. In: ICCS - International Conference on Computational Science, 2013, Barcelona. **Proceedings of ICCS**, v. 18, p. 759-768, 2013.

CRESPO, A.; GARCIA-MOLINA, H. Semantic Overlay Networks for P2P Systems. **Lecture Notes in Computer Science**, v. 3601, p. 1-13, 2005.

DA SILVA, L. M.; BRAGA, R.; CAMPOS, F. Composer-Science: A semantic service based framework for workflow composition in e-Science projects. **Information Sciences**, v. 186, n. 1, p. 186-208, 2012.

DIAS, J. et al. SciMulator: Um Ambiente de Simulação de Workflows Científicos em Redes P2P. In: **VI Workshop de Redes Dinâmicas e Sistemas Peer to peer**, Gramado, Rio Grande do Sul, 2011.

DUCHAME, B. **Learning SPARQL**. Sebastopol: O'Realy Media, 2011.

FERREIRA FILHO, J. B. et al. An approach for semantic enrichment of software product lines. In: 16th International Software Product Line Conference. **Proceedings of the 16th International Software Product Line Conference**. New York: AMC, p. 188-195, 2012.

FINHOLT, T. A.; Collaboratories. **Annual review of information science and technology**, v. 36, n. 1, p. 73-107, 2002.

FREIRE, A. B.; FERREIRA, D.; FURTADO, A. Exemplo de Aplicação JXTA: um GIS P2P. In: **II Workshop de Redes Peer to peer**, Curitiba, 2006.

FURTADO, W. M.; FARIAS, P. P. M. CABO-P2P - Compartilhamento de Arquivos Baseado em Ontologias Sobre Infraestrutura P2P. In: **I Workshop de Redes Peer to peer**, Fortaleza, 2005.

GHAZOUALI, M.; BOUFAIDA, M. PM4SWS: A P2P Model for Semantic Web Services Discovery and Composition. **Journal of Advances in Information Technology**, v. 2, n. 1, p. 15-26, 2011.

GIL, Y. et al. Examining the Challenges of Scientific Workflows. **Computer**, v. 40, n. 12, p. 24-32, 2007.

GOBLE, C.; ROURE, D. The Impact of Workflow Tools on Data-centric Research. In: HEY, T.; TANSLEY, S.; TOLLE, K. (Org.). **The Fourth Paradigm**. Microsoft Research Redmond, p. 137-145, 2009.

GREENWOOD, M. et al. Provenance of e-science experiments-experience from bioinformatics. **Proceedings of UK e-Science All Hands Meeting 2003**, p. 223-225, 2003.

HAASE, P. et al. Bibster - A Semantics-Based Bibliographic Peer to peer System. **The Semantic Web**, 2004.

HEATH, L. S. e RAMAKRISHNAN, N. The emerging landscape of bioinformatics software systems. **Computer**, v. 35, n. 7, p. 41-45, 2002.

HENDLER, J. Science and the Semantic Web. **Science**, v. 299, n. 5606, p. 520-521, 2003.

HEY, T.; TREFETHEN, A. The Data Deluge: An e-Science Perspective. In: BERMAN, F.; FIX, G. C.; HEY, A. J. G. (Org.). **Grid Computing: Making the Global Infrastructure a Reality**. Wiley and Sons, . p. 809–824, 2003.

HOLT, A. P. Management-Oriented Models of Business Process. In: AALST, V. D. W. D. J. O. A. **Business Process Management: Models, Techniques, and Empirical Studies**. Berlin: Springer, 2000.

HOLLAND, R. C. G. et al. BioJava: an open-source framework for bioinformatics. **Bioinformatics**, v. 24, n. 18, p. 2096-2097, 2008.

IMENES, L. M.; LELLIS, M. **Descobrimos o Teorema de Pitágoras**. 1ª edição. Brasil: Scipione, 2000.

JAVA. **Java.com: Java + Você**. Java, 2014. Disponível em: <https://www.java.com/pt_BR/>. Acesso em: 02 de Janeiro de 2014.

JGRAPH. **JGraph: connecting the dots**. JGraph, 2013. Disponível em: <<http://www.jgraph.com/>>. Acesso em: 17 de Abril de 2013.

JOUNG, Y. J.; CHUANG, F.-Y. OntoZilla: An ontology-based, semi-structured, and evolutionary peer to peer network for information systems and services. **Future Generation Computer System**, v. 25, n. 1, p. 53-63, 2009.

JXTA. **JXTA(TM) Community Project**. JXTA - Project Kenai, 2013. Disponível em: <<http://java.net/projects/jxta/>>. Acesso em: 3 de Abril de 2013.

KEPLER. **The Kepler Project**. Kepler, Your Science Enabled, 2014. Disponível em: <<https://kepler-project.org/>>. Acesso em: 21 de Janeiro de 2014.

KLIPP, E. et al. **Systems biology in practice: concepts, implementation and application**. John Wiley & Sons, 2008.

KONA, S. et al. Generalized semantics-based service composition. In. **IEEE International Conference on Web Services, 2008**, p. 219-227, 2008.

KUMAR, S. e DUDLEY, J. Bioinformatics software for biologists in the genomics era. **Bioinformatics**, v. 23, n. 14, p. 1713-1717, 2007.

KUMAR, S. et al. MEGA: a biologist-centric software for evolutionary analysis of DNA and protein sequences. **Briefings in bioinformatics**, v. 9, n. 4, p. 299-306, 2008,

LLADÓ, E.; SALAMANCA, I. Xarop, a Semantic Peer to peer System for a Virtual Organization. In. **Semantic Web and Peer to peer**, Springer, p. 335-347, 2006.

LUA, E. K. et al. A survey and comparison of peer to peer overlay network schemes. **IEEE Communications Surveys and Tutorials**, v. 7, n. 2, p. 72-93, 2005.

MANDLER, B. et al. COMPOSE - A journey from the Internet of Things to the Internet of Services. In. 27th International Conference on Advanced Information

Networking and Applications Workshops. **27th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2013**, IEEE, p. 1217-1222, 2013.

MATTOSO, M.; WERNER, C.; TRAVASSOS, G.; BRAGANHOLO, V.; MURTA, L. Gerenciando Experimentos Científicos em Larga Escala. In: **SEMISH, CSBC**, Belém, Brasil, 121-135, 2008

MAWLOOD-YUNIS, A.-R.; WEISS, M.; SANTORO, N. A Reference Model for Semantic Peer to peer Networks. **Journal on Data Semantics**, 2011.

MEDJAHED, B.; BOUGUETTAYA, A. **Service composition for the Semantic Web**. Springer, p. 207, 2011.

MODICA, G. D.; TOMARCHIO, O.; VITA, L. A P2P Based Architecture For Semantic Web Service Discovery. **International Journal of Software Engineering and Knowledge Engineering**, 2011.

MURATA, T. Petri nets: Properties, analysis and applications. **Proceedings of the IEEE**, v. 77, n. 4, p. 541-580, 1989.

MYGRID. **myGrid**. myGrid, 2014. Disponível em: <<http://www.mygrid.org.uk/>>. Acesso em: 02 de Fevereiro de 2014.

NEJDL, W. et al. EDUTELLA: a P2P networking infrastructure based on RDF. In: 11th international conference on World Wide Web. **Proceedings of the 11th international conference on World Wide Web**, ACM, p. 604-615, 2002.

NENC. **Núcleo de Engenharia do Conhecimento**. Universidade Federal de Juiz de Fora, 2014. Disponível em <<http://plsql1.cnpq.br/buscaoperacional/detalhegrupo.jsp?grupo=0804103E8CXAC2>>. Acesso em: 02 de Fevereiro de 2014.

NETBEANS. **NetBeans IDE**. Welcome to NetBeans, 2014. Disponível em: <<https://netbeans.org/>>. Acesso em: 2 de Janeiro de 2014.

NOVAIS, E. B. **e-ScienceNet: uma Rede Ponto a Ponto Semântica para aplicações em e-Science**. Dissertação (Mestrado em Modelagem Computacional). Universidade Federal de Juiz de Fora. 2012.

NUMBERG, B. Introduction to the Peer to peer Sockets Project. O'Reilly on java.com. 2003. Disponível em: <<http://www.onjava.com/pub/a/onjava/2003/12/03/p2psockets.html>> Acesso em: 04 de Abril de 2014.

OLSON, G. M. The next generation of science collaboratories. In: **CTS '09: Proceedings of the 2009 International Symposium on Collaborative Technologies and Systems**, IEEE Computer Society: Washington, DC, USA, pp. xv–xvi, 2009.

ORACLE. **Oracle BPEL Process Manager**. Oracle, 2014. Disponível em: <<http://www.oracle.com/technetwork/middleware/bpel/overview/index.html>>. Acesso em: 21 de Janeiro de 2014.

OWL API. **The OWL API**. OWL API, 2013. Disponível em: <<http://owlapi.sourceforge.net/>>. Acesso em: 3 de Abril de 2013.

OWL. **Web Ontology Language (OWL)**. W3C Semantic Web, 2013. Disponível em: <<http://www.w3.org/2004/OWL/>>. Acesso em: 3 de Abril de 2013.

OWL2.0. **OWL 2 Web Ontology Language Document Overview (Second Edition)**. W3C Recommendation, 2012. Disponível em: <<http://www.w3.org/TR/owl2-overview/>>. Acesso em: 15 de Janeiro de 2014.

OWL-S API. **OWL-S API**. OWL-S API, 2013. Disponível em: <<http://www.mindswap.org/2004/owl-s/api/>>. Acesso em: 3 de Abril de 2013.

OWL-S. **OWL-S: Semantic Markup for Web Services**. W3C Member Submission, 2014. Disponível em: <<http://www.w3.org/Submission/OWL-S/>>. Acesso em: 15 de Janeiro de 2014.

PÁDUA, S. I. D. et al. Redes de petri aplicadas aos sistemas de gerenciamento de Workflow. In: **Encontro Nacional de Engenharia de Produção**. Curitiba: 2002.

PETRI, C. A. **Kommunikation mit Automaten**. Institut fur instrumentelle Mathematik. Bonn. 1962.

PREDICT8. **Web Services Monitoring and SOA Governance - predic8**. Predict8, 2014. Disponível em: <<http://predic8.com/>>.

PROSDOCIMI, F. et al. Bioinformática: Manual do usuário. Biotecnologia, **Ciência e Desenvolvimento**, v. 5, n. 29, p. 12-25, 2002.

ROCHA, J. et al. Peer to peer: Computação Colaborativa na Internet. In: **Simpósio Brasileiro de Redes de Computadores**, 2004.

SANGER, F.; NICKLEN, S.; COULSON, A. R. DNA sequencing with chain-terminating inhibitors. **Proceedings of the National Academy of Sciences**, v. 74, n. 12, p. 5463-5467, 1977.

SAWSDL. **Semantic Annotations for WSDL Working Group**. W3C Architecture Domain, 2014. Disponível em: <<http://www.w3.org/2002/ws/sawSDL/>>. Acesso em: 15 Janeiro 2014.

SLETHOLT, M. T. et al. A Literature Review of Agile Practices and Their Effects in Scientific Software Development. In. 4th International Workshop on Software

Engineering for Computational Science and Engineering. **Proceedings of SE-CSE11**, p. 1-9, 2011.

SOUZA, N. B. O. **Caracterização de Software Científico: Um Estudo de Caso em Modelagem Computacional**. Dissertação (Mestrado em Modelagem Computacional). Universidade Federal de Juiz de Fora. 2011.

STREULE, A. **Abstract Views on BPEL Proccess**. Dissertação. Universität Stuttgart, 2009.

STUCKENSCHMIDT, H. et al. Peer to peer and semantic web. In: **Semantic Web and Peer to Peer**. Berlin: Springer, p. 1-17, 2006.

SYCARA, K. et al. Automated discovery, interaction and composition of semantic web services. **Web Semantics: Science, Services and Agents on the World Wide Web**, v. 1, n. 1, p. 27-46, 2011.

STEVENS, R. D.; ROBINSON, A. J.; GOBLE, C. A. myGrid: personalised bioinformatics on the information grid. **Bioinformatics**, v. 19, n. suppl 1, p. i302-i304, 2003.

TANENBAUM, A. S.; VAN STEEN, M. **Distributed systems: principles and paradigms**. 2ª edição. Upper River Side: Pearson Prentice Hall, 2006.

TAVERNA. **Taverna - open source and domain independent Workflow Management System**. Taverna Workflow Management System, 2014.
Disponível em: <<http://www.taverna.org.uk/>>. Acesso em: 07 de Janeiro de 2014.

VAN DER AALST, W. M. The application of Petri nets to workflow management. **Journal of circuits, systems, and computers**, v. 8, n. 1, p. 21-66, 1998.

VISTRAILS. **VisTrails**. VisTrailsWiki, 2014. Disponível em: <http://www.vistrails.org/index.php/Main_Page>. Acesso em: 21 de Janeiro de 2014.

VMWARE. **Virtualização da VMware para desktop e servidor, aplicativos, nuvens públicas e híbridas**. VmWare, 2014. Disponível em: <<http://www.vmware.com/br/>>. Acesso em: 02 de Fevereiro de 2014.

WFMC. 1995. **Workflow Management Coalition – The Workflow Reference Model**. Workflow Management Coalition, 2010. Disponível em: <<http://www.wfmc.org/reference-model.html>>. Acesso em: 10 de Abril de 2010.

WILKINSON, M. Web Science Studies into Semantic Web Service-Based Research Environments. In: **The Fifth International Conferences on Advanced Service Computing**. Service Computation 2013, p. 71-74, 2013.

WILSON, B. J. **JXTA**. Indianapolis: New Riders Publishing, 2002.

WHITE, S. A. Introduction to BPMN. IBM Cooperation, v. 2, 2004.

WSDL-S. **Web Service Semantics - WSDL-S**. W3C Member Submission, 2014. Disponível em: <<http://www.w3.org/Submission/WSDL-S/>>. Acesso em: 15 de Janeiro de 2014.

WSMO. **Web Service Modeling Ontology (WSMO)**. W3C Member Submission, 2014. Disponível em: <<http://www.w3.org/Submission/WSMO/>>. Acesso em: 15 de Janeiro de 2014.

XSTREAM. **XStream - About Xstream**. XStream, 2013. Disponível em: <<http://xstream.codehaus.org/>>. Acesso em: 17 de Abril de 2013.

ZHAO, J. et al. Mining Taverna's semantic web of provenance. **Concurrency and Computation: Practice and Experience**, v. 20, n. 5, p. 463-472, 2008.

APÊNDICE A – XML DE COMUNICAÇÃO DE GRUPOS SEMÂNTICOS

```
<list>
  <com.esciencenet.models.PeerGroupModel>
    <groupName>Bioinformatics</groupName>
    <groupID>urn:jxta:uuid-17DA279E047547A1BDAD8EC10EBE08C402</groupID>
    <groupArea>Bioinformatics</groupArea>
    <groupDescription>The semantic group base on bioinformatics
concepts.</groupDescription>
    <groupCreator>Tadeu PEER</groupCreator>
    <dataCriacao>29/11/2013</dataCriacao>
  </com.esciencenet.models.PeerGroupModel>
  <com.esciencenet.models.PeerGroupModel>
    <groupName>Mathematics</groupName>
    <groupID>urn:jxta:uuid-17DA279E047547A1BDAD8EC10EBE08AS56</groupID>
    <groupArea>Mathematics</groupArea>
    <groupDescription>The semantic group is based on math concepts.</groupDescription>
    <groupCreator>VIRTUALXP</groupCreator>
    <dataCriacao>29/11/2013</dataCriacao>
  </com.esciencenet.models.PeerGroupModel>
</list>
```

APÊNDICE B – ALGORITMO DE INFERÊNCIA DE TERMOS RELACIONADOS PARA A PESQUISA DE SERVIÇOS

```

public List<String> getAllSubAndEquivalentClasses(String owlPath, String owlClass){
    try{
        List<String> lstClasses = new ArrayList<>();

        OWLOntologyManager manager = OWLManager.createOWLOntologyManager();
        File owlFile = new File(owlPath);
        OWLOntology ont = manager.loadOntologyFromOntologyDocument(owlFile);

        PelletReasoner pelletReasoner =
PelletReasonerFactory.getInstance().createNonBufferingReasoner(ont);
        KnowledgeBase kb = pelletReasoner.getKB();
        PelletInfGraph graph = new org.mindswap.pellet.jena.PelletReasoner().bind(kb);

        InfModel infModel = ModelFactory.createInfModel(graph);

        String sql = "PREFIX owl: <http://www.w3.org/2002/07/owl#>\n"+
                    "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>\n" +
                    "PREFIX t: <"+ ont.getOntologyID().getOntologyIRI().toString() + "#>\n"
+
                    "SELECT * \n" +
                    "WHERE {?subject ?property t:"+ owlClass +"}";

        QueryExecution qe = QueryExecutionFactory.create(sql, infModel);
        ResultSet results = qe.execSelect();

        while(results.hasNext()){
            QuerySolution row = results.next();

            if(!row.getResource("subject").getLocalName().contains("Nothing")){
                if((row.getResource("property").getLocalName().contains("subClassOf")) ||
(row.getResource("property").getLocalName().contains("equivalentClass"))){
                    if(lstClasses.indexOf(row.getResource("subject").getLocalName()) == -
1){
                        lstClasses.add(row.getResource("subject").getLocalName());
                    }
                }
            }
        }

        return lstClasses;
    }catch(ReasonerException | OWLOntologyCreationException e){
        JOptionPane.showMessageDialog(null, "Wasn't possible to get all classes terms.",
":: e-ScienceNet ::", JOptionPane.ERROR_MESSAGE);
        return null;
    }
}

```


APÊNDICE C – BUSCA DE INFORMAÇÕES SOBRE SERVIÇOS NA OWLSE

```

public OWLServiceOperation serviceSemanticResearch(String owlsPath, String serviceDomainTerm, boolean
isToLoadAll){
    try{
        OWLOntologyManager manager = OWLManager.createOWLOntologyManager();
        File owlFile = new File(owlsPath);
        OWLOntology ont = manager.loadOntologyFromOntologyDocument(owlFile);

        OWLKnowledgeBase kb = OWLFactory.createKB();
        kb.setReasoner("Pellet");

        Service service = kb.readService(owlFile.toURI().toURL().toExternalForm());
        Property domainTermProp = new
PropertyImpl(ont.getOntologyID().getOntologyIRI().toString() + "#domainTerm");
        OWLDataPropertyImpl dataProperty = new OWLDataPropertyImpl(service.getOntology(),
domainTermProp);
        OWLDataValue domainTermValue = service.getProperty(dataProperty);

        String domainTerm = "";
        if(domainTermValue != null){
            domainTerm =
domainTermValue.getValue().toString().substring(domainTermValue.getValue().toString().lastIndex
Of("#") + 1,
domainTermValue.getValue().toString().length());
        }

        String tempDomainTerm = domainTerm;
        domainTerm = (isToLoadAll ? "" : domainTerm);

        OWLServiceOperation owlsOperation = null;

        if(domainTerm.equals(serviceDomainTerm)){
            final String URI_OWLS_PROFILE = "http://www.daml.org/services/owl-
s/1.1/Profile.owl#";
            Profile profile = service.getProfile();

            Property serviceRealNameProp = new
PropertyImpl(ont.getOntologyID().getOntologyIRI().toString() + "#serviceRealName");
            OWLDataPropertyImpl serviceRealNameProperty = new
OWLDataPropertyImpl(profile.getOntology(), serviceRealNameProp);
            OWLDataValue serviceRealNameValue =
profile.getProperty(serviceRealNameProperty);

            Property textDescriptionProp = new PropertyImpl(URI_OWLS_PROFILE +
"textDescription");
            OWLDataPropertyImpl textDescriptionProperty = new
OWLDataPropertyImpl(profile.getOntology(), textDescriptionProp);
            OWLDataValue textDescriptionValue =
profile.getProperty(textDescriptionProperty);

            owlsOperation = new OWLServiceOperation();
            owlsOperation.setOperationName(service.getLocalName().replace("Service", ""));
            owlsOperation.setOperationDomainTerm(tempDomainTerm);
            if(serviceRealNameValue != null){
owlsOperation.setOperationRealName(serviceRealNameValue.getValue().toString());
            }
            if(textDescriptionValue != null){
owlsOperation.setServiceDescription(textDescriptionValue.getValue().toString());
            }

            OWLOntology owlsProcess = this.getOWLClass(ont, "process");

```

```

        for(Input input : profile.getInputs()){
            Property parameterTypeProp = new
PropertyImpl(owlsProcess.getOntologyID().getOntologyIRI().toString() + "#parameterType");
            OWLDataPropertyImpl parameterType = new
OWLDataPropertyImpl(input.getOntology(), parameterTypeProp);
            OWLDataValue parameterTypeValue = input.getProperty(parameterType);

            Property parameterTypeXSDProp = new
PropertyImpl(ont.getOntologyID().getOntologyIRI().toString() + "#parameterTypeXSD");
            OWLDataPropertyImpl parameterTypeXSD = new
OWLDataPropertyImpl(input.getOntology(), parameterTypeXSDProp);
            OWLDataValue parameterTypeXSDValue = input.getProperty(parameterTypeXSD);

            if(parameterTypeValue != null){
                OWLSParam owlsParam = new OWLSParam();
                owlsParam.setParamName(input.getLabel());
                owlsParam.setParamDomainTerm(parameterTypeValue.getValue().toString());
                owlsParam.setParamType(parameterTypeXSDValue.getValue().toString());

                owlsOperation.getInputs().add(owlsParam);
            }
        }

        for(Output output : profile.getOutputs()){
            Property parameterTypeProp = new
PropertyImpl(owlsProcess.getOntologyID().getOntologyIRI().toString() + "#parameterType");
            OWLDataPropertyImpl parameterType = new
OWLDataPropertyImpl(output.getOntology(), parameterTypeProp);
            OWLDataValue parameterTypeValue = output.getProperty(parameterType);

            Property parameterTypeXSDProp = new
PropertyImpl(ont.getOntologyID().getOntologyIRI().toString() + "#parameterTypeXSD");
            OWLDataPropertyImpl parameterTypeXSD = new
OWLDataPropertyImpl(output.getOntology(), parameterTypeXSDProp);
            OWLDataValue parameterTypeXSDValue = output.getProperty(parameterTypeXSD);

            if(parameterTypeValue != null){
                OWLSParam owlsParam = new OWLSParam();
                owlsParam.setParamName(output.getLabel());
                owlsParam.setParamDomainTerm(parameterTypeValue.getValue().toString());
                owlsParam.setParamType(parameterTypeXSDValue.getValue().toString());

                owlsOperation.getOutputs().add(owlsParam);
            }
        }

        for(AtomicGrounding atomicGrounding :
service.getGrounding().getAtomicGroundings()){
            owlsOperation.setServiceURL(atomicGrounding.getDescriptionURL().toString());
        }
    }

    return owlsOperation;
} catch(OWLOntologyCreationException | IOException | URISyntaxException e){
    JOptionPane.showMessageDialog(null,
        "Wasn't possible to get all information about the service in OWL-S
file.\n\n" + e,
        ".: e-ScienceNet .:", JOptionPane.ERROR_MESSAGE);
    return null;
}
}

```

APÊNDICE D – MENSAGEM SOAP PARA REQUISIÇÃO DE WEB SERVICE FETCHBATCH

```
<s11:Envelope xmlns:s11='http://schemas.xmlsoap.org/soap/envelope/'>
  <s11:Body>
    <ns1:fetchBatch xmlns:ns1='http://www.ebi.ac.uk/ws/services/WSDbfetchDoclit'>
      <ns1:db>uniprotkb</ns1:db>
      <ns1:ids>o90777</ns1:ids>
      <ns1:format>fasta</ns1:format>
      <ns1:style>raw</ns1:style>
    </ns1:fetchBatch>
  </s11:Body>
</s11:Envelope>
```

APÊNDICE E – MENSAGEM SOAP DE RESPOSTA DO WEB SERVICE FETCHBATCH

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <fetchBatchResponse
xmlns="http://www.ebi.ac.uk/ws/services/WSDbfetchDoclit">
      <fetchBatchReturn>
        &gt;tr|090777|090777_9PLVG HIV-1 protease (Fragment) OS=Human immunodeficiency
virus GN=HIV-1 protease PE=2 SV=1
PQVTLWQRPIVTIKIGGQLKEALLDTGADDTVLEEMSLPGKWKPKMIGGIGGFIKVRQYD
QVSIIEICGHKAIGTVLIGPTPVNIIGRNLLTQLGCTLNF
      </fetchBatchReturn>
    </fetchBatchResponse>
  </soapenv:Body>
</soapenv:Envelope>
```