

Francisco Araujo de Almeida Neto

**Um Framework para  
Criação de Serviços Inteligentes  
em Ambientes Flexíveis na Web**

Manaus, Amazonas, Brasil

24/11/2017

Francisco Araujo de Almeida Neto

**Um Framework para  
Criação de Serviços Inteligentes  
em Ambientes Flexíveis na Web**

Tese submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas como requisito parcial para a obtenção do Doutorado em Informática.

Universidade Federal do Amazonas – UFAM

Instituto de Computação

Programa de Pós-Graduação em Informática

Orientador: Alberto Nogueira de Castro Jr.

Manaus, Amazonas, Brasil

24/11/2017

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

A447u Almeida Neto, Francisco Araujo de  
Um Framework para Criação de Serviços Inteligentes em  
Ambientes Flexíveis na Web / Francisco Araujo de Almeida Neto.  
2017  
139 f.: il. color; 31 cm.

Orientador: Alberto Nogueira de Castro Jr  
Tese (Doutorado em Informática) - Universidade Federal do  
Amazonas.

1. Serviço Inteligente. 2. Ambiente Colaborativo Flexível. 3.  
Sistema Multiagente. 4. Inteligência Artificial. I. Jr, Alberto Nogueira  
de Castro II. Universidade Federal do Amazonas III. Título



PODER EXECUTIVO  
MINISTÉRIO DA EDUCAÇÃO  
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



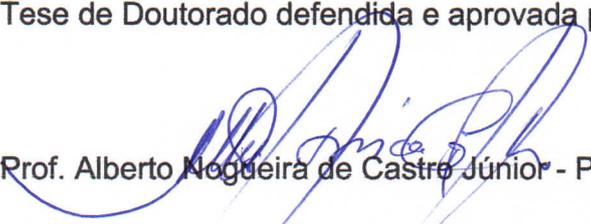
UFAM

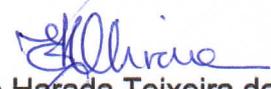
## FOLHA DE APROVAÇÃO

**"Um Framework para Criação de Serviços Inteligentes em Ambientes Flexíveis na Web"**

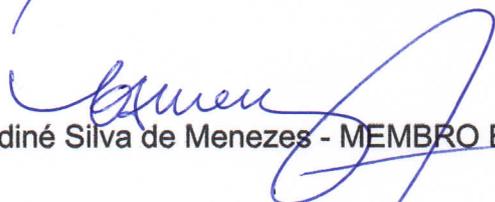
**FRANCISCO ARAUJO DE ALMEIDA NETO**

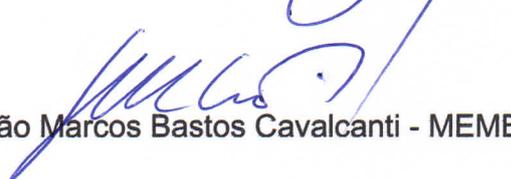
Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

  
Prof. Alberto Nogueira de Castro Júnior - PRESIDENTE

  
Profa. Elaine Harada Teixeira de Oliveira - MEMBRO INTERNO

  
Prof. Altigran Soares da Silva - MEMBRO INTERNO

  
Prof. Crediné Silva de Menezes - MEMBRO EXTERNO

  
Prof. João Marcos Bastos Cavalcanti - MEMBRO EXTERNO

Manaus, 24 de Novembro de 2017

*Este trabalho é dedicado às minhas filhas Ana Carolina e Maria Helena,  
que elas sempre lembrem que perseverança e dedicação são valores maiores que talento.*

# Agradecimentos

Agradeço a minha esposa Josane Albuquerque, pelo apoio e compreensão às minhas necessidades durante a execução desse projeto.

Agradeço ao meu Professor Orientador Alberto Castro, que me orientou e aconselhou durante a execução deste projeto.

Agradeço a CAPES por me conceder essa oportunidade.

*“A tarefa não é tanto ver aquilo que ninguém viu,  
mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê.”*  
*(Arthur Schopenhauer)*

# Resumo

Ambientes flexíveis na internet precisam de um meio de prover serviços inteligentes a seus usuários, o que envolve a definição de protocolos, linguagens de especificação, esquemas para a geração e integração de tais serviços. No projeto aqui descrito, mostramos a concepção e desenvolvimento de um *framework* que resolve esse problema de forma genérica. A validação da proposta envolveu a implementação das instâncias dos elementos formalizados.

Identificamos como serviços inteligentes são implementados atualmente em ambientes colaborativos na web e propusemos uma classificação diferenciada para os tipos de tarefas inteligentes sob o ponto de vista de suas funcionalidades, o que foi posteriormente utilizado na criação de um modelo de referência para uma camada de serviços inteligentes.

A partir do modelo de referência foi construído uma arquitetura na qual descrevemos os principais componentes necessários bem como fazemos uma descrição de seus comportamentos e relacionamentos.

Por fim, foi desenvolvido uma implementação para a arquitetura proposta.

**Palavras-chave:** Serviços Inteligentes. Camada de Serviços. Ambientes Flexíveis para web.

# Abstract

Internet flexible environments need a more formal process to provide intelligent services to their users. This process should define protocols, specification languages and schemes for generation and integration of such intelligent services. In the project reported here, we show the design and development of a framework that solves this problem in a generic way. The validation of this framework proposal involved an implementation of an instance of the formal elements described in this work.

We have studied and classified how intelligent services are currently implemented in CSCW environments in the web and we proposed a differentiated interpretation for the types of tasks intelligent based on the point of view of its functionalities. This classification was later used in the creation of a reference model for a layer of intelligent services.

From the reference model a reference architecture was assembled in which we describe its elements as well as a description of their behaviors and relationships.

Finally, an implementation was produced for the proposed architecture.

**Keywords:** Intelligent Services. Flexible Environments. Intelligent Service Layer.

# Lista de ilustrações

Figura 1 – Funcionamento de um Agente. . . . .	36
Figura 2 – Subsumption Architecture. . . . .	38
Figura 3 – Representação do Modelo 3C. . . . .	40
Figura 4 – Esboço informal da arquitetura do MOrFEU. . . . .	48
Figura 5 – Funcionamento de um VCom no MOrFEU. . . . .	49
Figura 6 – Processo de publicação de um UPL. . . . .	50
Figura 7 – Processo de visualização de um VCom. . . . .	51
Figura 8 – Cenário de Exemplo. . . . .	71
Figura 9 – Nossa proposta de visão para serviços inteligentes. . . . .	73
Figura 10 – Mapa conceitual de exemplo. . . . .	75
Figura 11 – Relações do conceito Usuário. . . . .	76
Figura 12 – Relações do conceito Interface com Usuário. . . . .	77
Figura 13 – Relações do conceito Ferramentas do Ambiente. . . . .	78
Figura 14 – Relações do conceito Tarefa. . . . .	79
Figura 15 – Relações do conceito Tarefa Aprendida. . . . .	80
Figura 16 – Relações do conceito Base de Conhecimento. . . . .	82
Figura 17 – Relações do conceito Regras para Aprendizagem. . . . .	83
Figura 18 – Relações do conceito Informação. . . . .	84
Figura 19 – Relações do conceito Ambiente. . . . .	85
Figura 20 – Modelo de referência para uma Camada de Serviços Inteligentes. . . . .	86
Figura 21 – Processo de criação das tarefas aprendidas. . . . .	87
Figura 22 – Arquitetura de referência utilizada. . . . .	92
Figura 23 – Cenário Um. . . . .	99
Figura 24 – Cenário Dois. . . . .	100
Figura 25 – Sintaxe do LCC. . . . .	106
Figura 26 – Aumento da complexidade do serviço com combinação de classes. . . . .	108
Figura 27 – Modelo de um sistema computacional como uma caixa preta. . . . .	108
Figura 28 – Visão caixa preta do Modelo de Interação. . . . .	109
Figura 29 – Autômato representando o fluxo de execução do serviço. . . . .	111
Figura 30 – Comparação Subsumption e o Modelo de Interação Proposto. . . . .	112
Figura 31 – Serviços no Modelo de Interação. . . . .	112
Figura 32 – Arquitetura do Fantasos. . . . .	117
Figura 33 – Níveis de acesso a interface. . . . .	119
Figura 34 – Transferência do DataLayer. . . . .	121
Figura 35 – Infraestrutura do Fantasos. . . . .	123
Figura 36 – Esboço do framework. . . . .	170

Figura 37 – Fluxo de execução de um serviço inteligente. . . . .	183
Figura 38 – Tela de debug do OKK. . . . .	192
Figura 39 – Mineração dos dados do Moodle. . . . .	195
Figura 40 – Modelo para Learning Analytics. . . . .	195
Figura 41 – Descrição da tabela de logs do Moodle 3.1.3. . . . .	197
Figura 42 – Modelo de referência. . . . .	199
Figura 43 – Arquitetura de referência. . . . .	204
Figura 44 – Principais Casos de Uso. . . . .	209
Figura 45 – Modelo ER do plugin. . . . .	211
Figura 46 – Diagrama de Atividades do Plugin . . . . .	211

# Lista de códigos

Código 1 – Código LCC de Exemplo . . . . .	106
Código 2 – Código do Modelo de Interação . . . . .	113
Código 3 – Código Agente SugereBibliografia . . . . .	131
Código 4 – Código Agente SugereGrupos . . . . .	131
Código 5 – Código Agente PostProfessor . . . . .	132
Código 6 – Código Agente LembreteAluno . . . . .	134
Código 7 – String de Busca em Português . . . . .	154
Código 8 – String de Busca em Inglês . . . . .	155
Código 9 – Código LCC Teste do Modelo de Interação . . . . .	183

# Lista de tabelas

Tabela 1 – Exemplos de ação, tarefa e serviço . . . . .	33
Tabela 2 – Matriz de espaço-tempo proposta por Johansen. . . . .	43
Tabela 3 – Extensão da matriz de espaço-tempo proposta por Grudin. . . . .	43
Tabela 4 – Classificação de CSCW proposta por Penichet. . . . .	44
Tabela 5 – Artigos classificados pelo tipo de ambiente . . . . .	58
Tabela 6 – Classificação pelas técnicas de IA utilizadas . . . . .	58
Tabela 7 – Distribuição dos Tipos de Serviços Inteligentes na Revisão Sistemática	61
Tabela 8 – Distribuição das Formas de Auxílio ao Usuário com Serviços Inteligentes	62
Tabela 9 – Exemplos de possíveis tarefas aprendidas. . . . .	82
Tabela 10 – Elementos : Modelo de Referência X Arquitetura de Referência . . . . .	97
Tabela 11 – Agentes e descrição . . . . .	128
Tabela 12 – Artigos coletados para a revisão sistemática . . . . .	156
Tabela 13 – Artigos CSCW x CSCL . . . . .	162
Tabela 14 – Artigos por técnicas de IA utilizadas . . . . .	162
Tabela 15 – Artigos por Tipos de Serviços Inteligentes . . . . .	166
Tabela 16 – Artigos por Formas de Auxílio ao Usuário com Serviços Inteligentes . .	168
Tabela 17 – Total de tarefas encontradas por classe . . . . .	176
Tabela 18 – Configurações do plugin . . . . .	209
Tabela 19 – Itens minerados . . . . .	213

# Lista de abreviaturas e siglas

ACFW	Ambiente Colaborativo Flexível na Web
AHV	Assistente Humano Virtual
ASV	Assistente Software Virtual
AVA	Ambiente Virtual de Aprendizagem
BNF	<i>Backus-Naur Form</i>
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CED	Centro de Educação a Distância
CSCL	<i>Computer Supported Collaborative Learning</i>
CSCW	<i>Computer Supported Cooperative Work</i>
CSI	Camada de Serviços Inteligentes
CVA	Companheiro Virtual de Aprendizado
ETL	<i>Extract, Transform, Load</i>
FIPA	<i>Foundation for Intelligent Physical Agents</i>
HAV	Humano Assistente Virtual
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
IA	Inteligência Artificial
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IETF	<i>Internet Engineering Task Force</i>
IP	Internet Protocol
JADE	<i>Java Agent DEvelopment Framework</i>
JSON	<i>JavaScript Object Notation</i>

JSON	<i>JavaScript Object Notation</i>
LCC	<i>Lightweight Coordination Calculus</i>
LGPL	<i>Lesser General Public License,</i>
MEC	Ministério da Educação
MI	Modelo de Interação
MOrFEU	Multi-Organizador Flexível para Espaços virtuais
MVC	<i>Model View Control</i>
OKK	<i>Open Kernel Knowledge</i>
PHP	<i>PHP: Hypertext Preprocessor</i>
PPGI	Programa de Pós-graduação em Informática da UFAM
RA	Regras de Associação
SA	<i>Subsumption Architecture</i>
SAV	Software Assistente Virtual
SE	Sistema Especialista
SGBD	Sistema Gerenciador de Banco de Dados
SMA	Sistema Multiagente
SQL	<i>Structured Query Language</i>
STI	Sistema Tutor Inteligente
TCP	<i>Transmission Control Protocol</i>
UAB	Universidade Aberta do Brasil
UFAM	Universidade Federal do Amazonas
UPI	Unidade de Produção Intelectual
URL	<i>Uniform Resource Locator</i>
VCom	Veículo de Comunicação
XML	<i>eXtensible Markup Language</i>
XPath	<i>XML Path Language</i>
XSLT	<i>eXtensible Stylesheet Language for Transformation</i>

# Sumário

<b>I</b>	<b>APRESENTAÇÃO DA PESQUISA</b>	<b>18</b>
1	INTRODUÇÃO . . . . .	19
1.1	Identificação e Relevância do Problema . . . . .	21
1.2	Questões de Pesquisa . . . . .	22
1.3	Objetivos . . . . .	24
1.4	Delimitação da Proposta e Pressupostos . . . . .	24
1.5	Originalidade . . . . .	26
1.6	Adequação às Linhas de Pesquisa da Pós-graduação . . . . .	27
1.7	Integração a Outros Projetos no Tema . . . . .	27
1.8	Elementos de Pesquisa . . . . .	28
1.9	Enquadramento Metodológico . . . . .	28
1.10	Etapas de Execução do Trabalho . . . . .	29
1.11	Organização do Documento . . . . .	30
<b>II</b>	<b>REFERENCIAIS TEÓRICOS</b>	<b>31</b>
2	FUNDAMENTAÇÃO TEÓRICA . . . . .	32
2.1	Serviço Inteligente . . . . .	32
2.2	Desenvolvimento de aplicações inteligentes . . . . .	34
2.3	Modelo de Colaboração 3C . . . . .	39
2.4	CSCW . . . . .	41
2.5	Ambientes Colaborativos na <i>Web</i> . . . . .	46
2.6	Ambientes Colaborativos Flexíveis na <i>Web</i> . . . . .	47
2.7	MOrFEU . . . . .	48
2.8	Conclusão do Capítulo . . . . .	51
3	TRABALHOS RELACIONADOS . . . . .	53
3.1	Problemas Correlatos . . . . .	53
3.2	Resumo da Revisão Sistemática . . . . .	57
3.3	Validação da Classificação de Serviço Inteligente . . . . .	63
3.4	Conclusão do Capítulo . . . . .	67

<b>III</b>	<b>RESULTADOS</b>	<b>69</b>
<b>4</b>	<b>UM MODELO DE REFERÊNCIA PARA SERVIÇOS INTELIGENTES NA WEB</b>	<b>70</b>
4.1	A Analogia do Assistente Humano Virtual	70
4.2	Definição de Modelo de Referência	73
4.3	Convenções Utilizadas na Descrição	74
4.4	O Modelo de Referência	75
4.5	Usando o Modelo de Referência	87
4.6	Conclusão do Capítulo	88
<b>5</b>	<b>O <i>FRAMEWORK</i> PROPOSTO</b>	<b>90</b>
5.1	Definição de Arquitetura de Referência	90
5.2	Conversões Utilizadas na Descrição	90
5.3	Arquitetura de Referência: Detalhamento	91
5.4	Do Modelo para a Arquitetura	97
5.5	Exemplos de Uso	97
5.6	Conclusão do Capítulo	100
<b>6</b>	<b>IMPLEMENTAÇÃO DA CAMADA DE SERVIÇOS INTELIGENTES</b>	<b>103</b>
6.1	O Protocolo de Interação	103
6.2	Fantasmagoria - Uma Camada de Serviços Inteligentes	116
6.3	Conclusão do Capítulo	124
<b>7</b>	<b>POSSIBILIDADES DE UTILIZAÇÃO</b>	<b>125</b>
7.1	Criando Serviços Inteligentes	125
7.2	Projeto de Mineração de Dados UAB	127
7.3	Um Exemplo no MOrFEU	128
7.4	Conclusão do Capítulo	135
<b>8</b>	<b>CONCLUSÃO</b>	<b>136</b>
8.1	Resultados Obtidos	136
8.2	Desdobramentos	138
8.3	Trabalhos Futuros	138
8.4	Considerações Finais	139
	<b>REFERÊNCIAS</b>	<b>140</b>

## APÊNDICES

149

### APÊNDICE A – REVISÃO SISTEMÁTICA SOBRE SERVIÇOS INTELIGENTES EM AMBIENTES COLABORATIVOS . . . . .

A.1	Introdução . . . . .	150
A.2	A Revisão Sistemática . . . . .	152
A.3	Análise dos Resultados . . . . .	161
A.4	Discussão sobre os Resultados . . . . .	168
A.5	Conclusão . . . . .	171

### APÊNDICE B – LISTA DE TAREFAS SUGERIDAS PARA TERCEIRIZAÇÃO . . . . .

B.1	Procedimento . . . . .	172
B.2	Lista de Tarefas . . . . .	173
B.3	Classificação das Tarefas . . . . .	176
B.4	Conclusão . . . . .	181

### APÊNDICE C – USO DO LCC COMO PROTOCOLO DE INTERAÇÃO . . . . .

C.1	Definição do Problema . . . . .	182
C.2	Metodologia . . . . .	182
C.3	Criação de um Modelo de Interação . . . . .	183
C.4	Implementação de uma Instância Funcional . . . . .	186
C.5	Testes do Modelo de Interação . . . . .	192
C.6	Conclusões . . . . .	192

### APÊNDICE D – PROJETO DE MINERAÇÃO DE DADOS EDUCACIONAIS UAB . . . . .

D.1	Introdução . . . . .	193
D.2	Contextualização do Problema . . . . .	194
D.3	Especificação do Problema . . . . .	196
D.4	Modelo de Referência . . . . .	199
D.5	Arquitetura de Referência . . . . .	203
D.6	Implementação de uma Instância da Arquitetura . . . . .	207
D.7	Resultados Encontrados . . . . .	212

# Parte I

## Apresentação da pesquisa

# 1 Introdução

Como forma de atender a seus usuários os ambientes na Web estão criando serviços considerados cada vez mais inteligentes, usados para melhorar a interação tanto entre os próprios usuários como entre o usuário e o ambiente. Isso ocorre também em outros tipos de sistemas de software com intensa interação com usuários. Um exemplo é o demonstrado pelo experimento *AutoDraw*<sup>1</sup> onde o computador consegue identificar o objeto enquanto o usuário ainda está desenhando. Isto pode auxiliar, por exemplo, na criação de funções inteligentes de desenhos em ambientes de colaboração onde os usuários poderiam se comunicarem melhor ao fazer desenhos em uma tela virtual compartilhada com outros usuários do ambiente. Um outro exemplo seria a criação de sistemas de recomendação dentro de outras soluções computacionais como forma de auxiliar os usuários na escolha de quais opções seriam melhores para eles em um dado sistema. Outro exemplo é o desenvolvimento de sistemas de apoio à decisão com objetivo de fornecer planos de ação juntamente com justificativas sobre a escolha de um determinado plano em detrimento de outro, cabendo ao usuário a escolha final.

Apesar de existirem exemplos do desenvolvimento de serviços inteligentes em várias áreas de pesquisa, neste trabalho buscou-se por exemplos mais próximos do contexto aqui apresentado. Assim, como exemplos de pesquisa, especificação e desenvolvimento de serviços inteligentes em sistemas computacionais com contexto parecido ao objeto desta tese temos:

- **Correção automática de questões discursivas.** Em (FIGUEIRA et al., 2013) é apresentado um sistema especialista para a correção automática de questões discursivas sobre SQL na disciplina Banco de Dados. O sistema utiliza de técnicas de Processamento de Linguagem Natural, Comparação Textual e N-Gramas para realizar a correção das questões e apresenta acurácia satisfatória.
- **Detecção de estudantes com frustração.** Em (IEPSEN; BERCHT; REATEGUI, 2013) é apresentado um trabalho para detecção de estudantes com frustração durante o processo de aprendizagem de algoritmos usando um ambiente colaborativo. Ao investigar as possíveis relações entre os dados das interações do usuário com o ambiente e o seu estado psicológico juntamente com técnicas de Aprendizagem de Máquinas, são geradas regras para detecção do estado de frustração. Essas regras são então utilizadas para adaptar o processo de aprendizagem ao aluno.

---

<sup>1</sup> <<https://www.autodraw.com/>>

- **Sistema Tutor Inteligente.** Em (FONTES et al., 2013) é descrito um método para criação de um Sistema Tutor Inteligente capaz de não apenas auxiliar os alunos no processo de ensino-aprendizagem como também exercer um papel motivacional durante a aprendizagem. O Tutor é criado através da utilização de um agente pedagógico aninado com características socioafetivas e outros três agentes para detecção de estudantes passivos e recomendação de Objetos de Aprendizagem.
- **Apresentação de dicas ao usuário.** Em (MELO et al., 2013) é descrito o desenvolvimento de um módulo de software capaz de apresentar dicas a um estudante de como alcançar a melhor resposta para um exercício. O sistema auxilia os estudantes nas resoluções de exercícios, atividades em grupo e avaliações através de um feedback automático na forma de dicas.

Mais recentemente, devido ao melhoramento das técnicas de Aprendizagem de Máquina, nota-se um direcionamento para o desenvolvimento de ferramentas que possibilitam a criação de novos serviços inteligentes para os atuais ambientes que já estão em uso, como exemplo podemos citar o *TensorFlow*<sup>2</sup> (ABADI et al., 2016) que foi transformado recentemente em um projeto de código aberto e que é utilizado por grandes empresas da área de informática para criação de novos serviços.

No entanto, apesar dos avanços conseguidos com as pesquisas na área persiste o problema de que a concepção, organização e o desenvolvimento de serviços inteligentes em ambientes Web ainda é pouco estruturado, especialmente em contextos mais dinâmicos e flexíveis.

De modo geral, serviços inteligentes são acrescentados aos ambientes na Web, como resultado da tentativa de resolução de algum problema específico e não com o objetivo de se criar serviços inteligentes “genéricos”. Por exemplo, em um determinado Ambiente Virtual de Aprendizagem (AVA) os professores e tutores passam muito tempo com correções manuais de questões discursivas, caracterizando a demanda pela criação de um serviço no ambiente que possa corrigir a totalidade ou pelo menos a maioria das questões de forma automática. Uma possível solução para este problema seria a criação de um Sistema Especialista (SE) capaz de reproduzir o comportamento apresentado por um professor ou tutor ao corrigir as questões discursivas dentro do AVA, assim ao final da implantação desse SE o AVA teria um novo serviço inteligente disponível aos seus usuários. De forma análoga, outros serviços poderiam ser criados, porém essa abordagem apresenta os seguintes problemas:

1. Os serviços criados são fortemente acoplados ao ambiente onde operam, dificultando a sua reutilização em outros ambientes;

---

<sup>2</sup> <<https://www.tensorflow.org/>>

2. Falta de padronização na criação e na interface dos serviços;
3. Falta de comunicação entre os serviços criados, inclusive no mesmo ambiente;
4. Desperdício de recursos e retrabalho ao se recriar o mesmo serviço em ambientes diferentes.

Como atualmente os sistemas de informática, principalmente os desenvolvidos para a Web, estão em um constante processo de melhoria (pois a quantidade de pedidos de mudança dos usuários, corretiva ou evolutiva, em geral supera a capacidade dos desenvolvedores em atendê-las em tempo hábil), nota-se a necessidade de uma solução para criação de serviços inteligentes de forma eficiente e eficaz, especialmente no contexto de ambientes colaborativos flexíveis na Web. As chamadas metodologias de desenvolvimento ágeis surgiram como candidatas naturais a tratar tal problema, porém embora tais metodologias amenizem a situação, elas não resolvem o problema, e o desenvolvimento de um *framework* para toda uma classe de problemas viria a ser uma contribuição mais efetiva na corrida para atender as demandas dos usuários.

## 1.1 Identificação e Relevância do Problema

A popularização da Internet, o aumento do acesso com banda larga, e o crescimento das redes sociais tem incentivado os ambientes na *Web* a adotarem uma estrutura social em seus sítios. O que tem facilitado a coprodução de conteúdos por cada um dos seus usuários. Este excesso de informações geradas faz os usuários serem atraídos para ambientes capazes de se acomodarem às necessidades pessoais daqueles usuários. Portanto para a *Web* dos próximos anos é necessário conceber, desenvolver e aplicar tecnologias para atuarem de forma dinâmica e pró-ativa no apoio aos indivíduos e comunidades na realização de suas tarefas através da rede. Há então, uma evidente demanda por paradigmas para o desenvolvimento de ambientes virtuais que apoiem ações individuais e colaborativas aderentes a múltiplos contextos e tecnologias. Assim, com base nesse cenário têm surgido trabalhos propondo e estudando o uso de Ambientes Colaborativos Flexíveis na *Web*.

Tais ambientes, no contexto de AVAs, são descritos em (MENEZES et al., 2008), (SANTOS; CASTRO; MENEZES, 2012) e (SANTOS; CASTRO; MENEZES, 2010) onde podemos citar como suas características os seguintes itens: múltiplas formas de organizar os relacionamentos entre seus usuários, espaço pessoal do usuário flexível e reorganizável, visão da totalidade das produções individuais, vários ambientes dentro do site, autoria coletiva, agilidade no acesso às produções de todos os participantes de um mesmo ambiente e mudanças em tempo de execução nas suas estruturas e interfaces.

A criação de ambientes colaborativos flexíveis na *Web* fez surgir a necessidade da incorporação de serviços inteligentes dentro das ferramentas e apesar de já terem uma

parte da sua necessidade atendida, os usuários são cada vez mais exigentes, o que tem levado a um constante aprimoramento.

Assim, partindo-se de uma instância específica da classe dos problemas, buscou-se encontrar uma forma genérica de tratar o problema. A instância específica utilizada nessa pesquisa foi o MOrFEU, um paradigma para desenvolvimento de ambientes colaborativos virtuais flexíveis na *Web*.

A infraestrutura de serviços básicos do MOrFEU, baseada no arcabouço teórico desenvolvido em um projeto de doutorado no PPGI/UFAM, necessita, para ser utilizada em sua plenitude, de serviços e funcionalidades usualmente associados ao comportamento inteligente.

Tais funcionalidades se apresentam sob a forma de agentes que processam os dados e os relacionamentos entre os dados podendo, durante a sua execução, sugerir formas de reorganizar o ambiente para cada usuário, executar tarefas pelos usuários, prever atividades futuras a serem realizadas pelo usuário, entre outros usos. Esses agentes poderiam coletar a utilização dos subambientes internos para que ao se criar um novo subambiente, fossem sugeridas ao usuário as melhores formas de organizar os dados para resolver problemas ou ainda se adiantando na execução de algumas tarefas ou atividades para o usuário.

## 1.2 Questões de Pesquisa

A concepção e desenvolvimento de uma solução para a criação de uma camada de serviços inteligentes é um trabalho complexo que pode envolver as respostas a várias questões, como exemplo pode-se citar questões da seguinte natureza:

- Como o usuário humano de um sistema computacional percebe se o sistema é inteligente ou não?
- Como poderiam os diversos serviços inteligentes de um determinado ambiente conversarem entre si?
- É possível reutilizar serviços inteligentes desenvolvidos para um ambiente em outro ambiente?
- É possível comparar a inteligência de ambientes diferentes?
- É possível comparar dois serviços inteligentes que tenham o mesmo objetivo, porém construídos de formas diferentes?
- É possível padronizar a criação de serviços inteligentes?
- Podemos construir serviços inteligentes de forma distribuída?

Portanto, ao invés de buscar respostas a essas e outras possíveis perguntas através da proposta de um único, complexo e específico padrão, a tese que orientou nosso projeto de doutorado foi a concepção de um *framework* genérico e aberto, porém com nível de abstração suficiente para implementação de provas de conceito que conseguissem evidenciar sua factibilidade.

Neste sentido, por ser aberto e genérico, o *framework* proposto possibilitou que as questões e requisitos elencados fossem englobados e unificados, além de permitir a utilização de outras contribuições e trabalhos (de outros autores) sejam incluídos e disponibilizados dentro da solução, possibilitando o foco no desenvolvimento de ideias, conceitos, estratégias e procedimentos, inclusive considerando elementos que ainda estejam em um nível de maturidade abaixo do tradicional.

A pergunta geral da pesquisa aqui relatada pode ser enunciada da seguinte forma:

- Como uma “camada de serviços inteligentes” pode fornecer aos usuários de um ambiente colaborativo virtual flexível na Web, instâncias implementadas e utilizáveis de serviços inteligentes utilizando outros serviços, dados e informações já existentes dentro do ambiente?

Antecipando que uma “arquitetura de referência” foi o elemento conceitual utilizado na investigação orientada pela questão central de pesquisa, apresentamos as seguintes descrições relacionadas:

- **Arquitetura de Referência:** Um modelo dos elementos arquiteturais abstratos existente no domínio do problema de forma independente de tecnologias, protocolos ou soluções pré-existentes, conforme descrito em (BROWN et al., 2012).
- **Tarefa Inteligente:** Um conjunto de ações executadas por um sistema computacional de forma semelhante ao que seria feito por um humano em igual contexto ao tentar alcançar um objetivo em uma aplicação específica de um domínio, conforme descrito no [Capítulo 2](#).
- **Serviço Inteligente:** Instanciação e oferta em um ambiente computacional de uma tarefa inteligente, conforme descrito no [Capítulo 2](#).
- **Ambiente Colaborativo Virtual Flexível na Web:** Ambientes colaborativos existentes na Web com múltiplas formas de organizar: os relacionamentos entre seus usuários, os espaços pessoais dos usuários e os subambientes do sítio web. Tal reorganização podendo ser feita em tempo de execução nas suas estruturas e interfaces. Conforme descrito em (MENEZES et al., 2008), (SANTOS; CASTRO; MENEZES, 2012) e (SANTOS; CASTRO; MENEZES, 2010).

## 1.3 Objetivos

Para uma melhor caracterização do alcance da pesquisa, apresentamos os objetivos geral e específicos buscados neste projeto.

### 1.3.1 Objetivo Geral

Desenvolver um *framework* que possibilite a criação de soluções específicas para o problema de fornecimento de um conjunto básico de serviços inteligentes em ambientes colaborativos flexíveis na Web.

### 1.3.2 Objetivos Específicos

1. Deduzir um esquema de definição de serviços inteligentes para ambientes colaborativos flexíveis na Web;
2. Construir um Modelo de Referência da solução;
3. Criar uma Arquitetura de Referência da solução;
4. Definir um protocolo de comunicação entre agentes para esse tipo de ambiente;
5. Explicar instâncias de utilização do arcabouço (*framework*).

## 1.4 Delimitação da Proposta e Pressupostos

Em relação ao problema de criar serviços inteligentes para ambientes colaborativos flexíveis na Web temos vários subproblemas e questões relevantes, que são elencados a seguir e discutidos mais profundamente nos capítulos seguintes:

- **Definição de serviço inteligente.** Neste problema temos questões do tipo: Como definir se um serviço é inteligente? Essa definição pode sofrer alterações com o tempo? Como criar um serviço inteligente? Quais os pré-requisitos necessários?
- **Interação usuário-serviço.** O problema é relacionado a como um serviço deve interagir com um usuário e quando ele deve fazer isso. Algumas das questões relacionadas são: Como solicitar uma informação do usuário? Quando devo informar o usuário sobre um determinado acontecimento? O usuário tem alguma preferência ou necessidade para a forma como acontece a interação?
- **Portabilidade.** Aqui o problema está relacionado aos dispositivos utilizados pelos usuários para acesso ao ambiente. Como o ambiente é um sítio web os seus usuários podem ter acesso não apenas por navegadores instalados em suas estações de trabalho

como também através de smartphones e tablets, surgem questões como: Os serviços devem ser providos independentes do dispositivo utilizado para acesso ao ambiente? Deve-se personalizar a experiência de uso do usuário de acordo com o dispositivo? Como tratar essas diferenças?

- **Disponibilidade.** Se os serviços a serem oferecidos estiverem em repositórios distribuídos, como a camada de serviços inteligentes deve tratar a questão da indisponibilidade de um serviço? Pode existir um serviço distribuído que faça o mesmo trabalho que outro? Podemos ter várias instâncias do mesmo serviço?
- **Flexibilidade.** Esse problema é sobre a questão dos serviços adaptarem as suas funcionalidades às necessidades dos seus usuários. Se possível, eles devem fazer isso de maneira dinâmica e flexível para poder atender uma variedade maior de usuários.
- **Segurança.** A solução deste problema é de natureza vertical pois deve englobar as várias etapas do desenvolvimento de uma solução. Também é vertical no sentido que cada camada da aplicação final deve ter os seus requisitos de segurança atingidos.
- **Resolução de conflitos.** O problema aqui refere-se aos possíveis conflitos que vão ocorrer durante o funcionamento de uma solução. Por exemplo, como escolher entre dois serviços diferentes mas que entregam o mesmo resultado? O que fazer quando dois usuários diferentes solicitam um mesmo serviço ao mesmo tempo?
- **Interoperabilidade.** Aqui a preocupação é em como desenvolver uma solução que mantenha todas as suas partes funcionando e interagindo entre si. Como garantir que novas funcionalidades não interfiram na execução das antigas?

Como já mencionado, não é do escopo deste trabalho resolver todos os subproblemas que podem existir ao se criar uma camada de serviços inteligentes.

O que se pretende obter neste trabalho é uma discussão exploratória do tema e ao final uma proposta de um meio padronizado para o desenvolvimento de uma camada de serviços inteligentes para ambientes colaborativos flexíveis na Web. De tal forma que os desenvolvedores ou projetistas de sistemas tenham um ponto de apoio na construção de suas próprias soluções.

Precisamos esclarecer os pontos e questões que foram ou não abordados, ou ainda em qual nível de detalhamento foram tratados nesse trabalho.

A comunicação entre os usuários e a camada de serviços inteligentes apesar de importante não foi discutida em profundidade neste trabalho. Optamos por utilizar protocolos já conhecidos e amplamente utilizados tanto no mercado como na academia, como por exemplo o HTTP.

A questão da adaptação de um software para todos os dispositivos onde ele possa ser utilizado é importante, porém não fez parte do escopo deste trabalho, assim, não discutimos como a solução proposta poderia ser adaptada aos dispositivos usados para acessar o ambiente. No entanto, de forma a validar os modelos descritos neste trabalho, apresenta-se um protótipo que demonstra situações de interação com um usuário. Porém, como a solução apresentada ao final deste trabalho utiliza-se de padrões bem conhecidos no mercado acredita-se que ela poderia ser facilmente adaptada para uso por outros dispositivos além dos que foram usados durante o seu desenvolvimento.

Detalhar procedimentos relacionados a segurança da informação acabaria por aumentar o tempo do projeto e estender seu escopo para além do necessário, consideramos então que a questão poderia ser tratada com recursos tradicionais como a utilização de antivírus, firewalls ou criptografia (por exemplo, utilizando o protocolo HTTPS).

Quanto a resolução dos conflitos, nesse trabalho optamos por utilizar padrões, métodos ou tecnologias abertos que já implementam uma solução.

Resumidamente, podemos dizer que o objetivo de uma camada de serviços inteligentes é prover serviços para os usuários do ambiente ao qual a camada foi incorporada e que esses serviços sejam percebidos como inteligentes pelos usuários do ambiente. Portanto, buscou-se construir e prover uma referência, que seja aberta e interoperável, para o desenvolvimento de uma camada de serviços inteligentes. Para tanto, uma instância de implementação é apresentada e alguns serviços foram especificamente implementados para o ambiente onde a camada foi incorporada.

## 1.5 Originalidade

Nesta seção buscamos detalhar o valor da proposta e apresentamos os aspectos inéditos quando no contexto de ambientes colaborativos flexíveis na Web.

Com base na revisão sistemática efetuada (vide [Capítulo 3](#)), os seguintes aspectos são apontados como contribuições originais:

- **Um modelo de referência para camadas de serviços inteligentes.** Não foram encontradas referências para a criação de um modelo, metodologia ou *framework* para criação de serviços inteligentes, nesse sentido, um modelo de referência é relevante para auxiliar futuras discussões e pesquisas.
- **Uma arquitetura de referência para a construção de serviços inteligentes.** O que permitiria que várias instâncias de implementações baseadas nesta arquitetura pudessem compartilhar os serviços inteligentes criados.

- **Independência da Camada de Serviços Inteligentes.** A camada de inteligência foi desenvolvida de modo que o ambiente não dependa totalmente dela para seu funcionamento normal. Assim, caso a camada venha por qualquer motivo a falhar em seu funcionamento, o ambiente ainda poderia ser utilizado (mesmo que de forma menos eficiente).

Outras características originais deste trabalho já foram mencionados nas seções anteriores, e são evidenciadas na revisão sistemática feita no [Capítulo 3](#) e em outros pontos do texto.

## 1.6 Adequação às Linhas de Pesquisa da Pós-graduação

Levando em consideração os objetivos apontados para este trabalho podemos afirmar que o nosso trabalho se encaixa nas linhas de pesquisa do Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas (UFAM) na área de Inteligência Artificial nos tópicos de pesquisa em Ambientes Colaborativos Flexíveis na Web, Organização do Conhecimento e Sistemas Multiagente.

## 1.7 Integração a Outros Projetos no Tema

Este trabalho teve como inspiração a necessidade de serviços inteligentes para o projeto MOrFEU. Em ([MENEZES et al., 2008](#)) o MOrFEU é descrito como uma proposta para espaços virtuais, seu nome sendo um acrônimo para Multi-Organizador Flexível para Espaços virtuais, que propõe um paradigma diferenciado para o desenvolvimento de ambientes virtuais flexíveis e extensíveis, que visem apoiar trabalho e aprendizagem colaborativos, e que tem como foco a produção intelectual de indivíduos e grupos, bem como a marcação, organização e publicação dos artefatos resultantes de tal produção. Associado às características de flexibilidade e extensibilidade, o MOrFEU busca agregar um conjunto de ferramentas semânticas provendo uma “camada de inteligência” à sua estrutura. As ferramentas para registro e organização das interações e produção de artefatos, já foram implementadas num ambiente de testes desenvolvido conforme o arcabouço teórico do MOrFEU, bem como as ferramentas semânticas para apoio automático à colaboração, ainda em estruturação conceitual, situam-se no escopo da configuração da Web Semântico-social, que emerge como tema catalisador para a pesquisa na área e de modo geral, para as próximas gerações da Web. Neste sentido o MOrFEU ainda necessitava, para ser utilizado em sua plenitude, de serviços complementares que incorporassem funcionalidades usualmente associados com o comportamento inteligente.

## 1.8 Elementos de Pesquisa

Para que pudéssemos iniciar a resolução do problema central deste trabalho foi necessária uma pesquisa bibliográfica das áreas relativas ao problema ou a ele correlatas, sempre considerando as questões elencadas na [seção 1.2](#). Para tanto foi feita uma revisão sistemática inicial tendo como objetivo responder às seguintes perguntas:

- **Quais Serviços Inteligentes estão em uso atualmente nos ambientes colaborativos?** Ao observarmos quais as necessidades atendidas pelos serviços e como eles resolvem os problemas, pudemos investigar os elementos e técnicas comuns aos trabalhos que foram utilizados para a construção dos serviços. Assim, uma lista de elementos e conceitos unificadores que podem ser reaproveitados foi estabelecida.
- **Quais técnicas de Inteligência Artificial são citadas nesses trabalhos?** Ao observarmos quais técnicas de IA estão sendo aplicadas nas soluções dos problemas pudemos identificar quais situações são melhor resolvidas por quais técnicas, além de verificarmos quais técnicas poderiam ser utilizadas em mais de uma situação. Assim buscamos aumentar a flexibilidade das soluções propostas.

Para a realização da revisão sistemática e de outras pesquisas bibliográficas realizadas durante o projeto, os seguintes sítios web de indexação de artigos científicos foram utilizados para pesquisa e obtenção do material científico em formato digital:

- IEEE Xplore <<http://ieeexplore.ieee.org>>;
- Google Acadêmico <<http://scholar.google.com.br/>>;
- Portal de Periódicos CAPES/MEC <<http://www.periodicos.capes.gov.br/>>.

## 1.9 Enquadramento Metodológico

A seguir descrevemos a metodologia adotada e o método seguido, onde metodologia é o conjunto de processos empregados na investigação e método é a ordem em que os processos foram usados com o objetivo de conseguir os resultados desejados. Para o enquadramento metodológico foi consultado o trabalho descrito em ([FILIPPO; PIMENTEL; WAINER, 2012](#)), a seguir apresentamos o resultado:

- **Quanto ao Método de Pesquisa:** Hipotético-dedutivo. Onde o problema percebido é a necessidade da criação de uma camada de serviços inteligentes para que possamos utilizar o ambiente em todo o seu potencial. Assim, buscou-se a comprovação da hipótese de ser possível conceber um padrão de desenvolvimento para uma

camada de serviços inteligentes através da construção de deduções lógicas sobre a sua utilização.

- **Quanto à Abordagem de Pesquisa:** Principalmente qualitativa. O processo de desenvolvimento do *framework* não foi acompanhado através de medidas exclusivamente quantitativas mas sim através do uso de modelos e arquiteturas consideradas aderentes e úteis ao processo. A abordagem portanto é qualitativa em relação a implementação de uma instância do *framework* objeto da pesquisa, porém questões pontuais da solução do problema usaram uma abordagem quantitativa.
- **Quanto a Posição Epistemológica:** Positivista. As variáveis observadas dentro do contexto do problema foram objetivas e independentes da interpretação dos observadores. Acreditou-se que o problema poderia ser solucionado através do uso e controle dessas variáveis bem como através da descoberta de padrões e/ou relações causais entre elas.
- **Quanto a Natureza da Pesquisa:** Aplicada. Tivemos um objetivo bem definido com aplicação prática: a modelagem e utilização de uma Arquitetura de Referência para criação de uma camada de serviços inteligentes.
- **Quanto a Finalidade da Pesquisa:** Exploratória. Buscamos melhorar a descrição e definição do que vem a ser serviços inteligentes em ambientes colaborativos flexíveis na Web além de encontrar e descrever relações de causa e efeito que ocorrem no contexto do problema.
- **Quanto aos Procedimentos de Pesquisa:** Pesquisa bibliográfica; concepção e validação de modelos; formalização de processos; definição de protocolos e prova de conceito.
- **Quanto ao Tempo da Pesquisa:** Transversal em momentos distintos. As informações e análises necessárias a construção dos modelos foram coletadas e executadas mais de uma vez em diferentes momentos no decorrer do trabalho aqui descrito.
- **Quanto ao Tipo de Observação na Pesquisa:** Não-estruturada, não-participante, individual e sistemática. As observações apesar de serem feitas de forma controlada e com objetivo de realizar provas de conceito, não foram medidas com ferramentas definidas previamente.

Este enquadramento foi seguido durante todas as etapas do trabalho aqui descrito.

## 1.10 Etapas de Execução do Trabalho

As principais etapas realizadas durante a execução do trabalho foram:

1. Delimitação do tema;
2. Revisão Sistemática sobre serviços inteligentes nos atuais ambientes colaborativos;
3. Definição e desenvolvimento da Arquitetura de Referência;
  - a) Desenvolvimento de um Modelo de Referência;
  - b) Desenvolvimento de uma Arquitetura de Referência;
  - c) Implementação de uma instância da arquitetura;
4. Avaliação da proposta;
5. Elaboração da tese;
6. Relato e socialização dos resultados.

## 1.11 Organização do Documento

O restante deste trabalho está organizado da seguinte forma:

- No Capítulo 2 são apresentados os conceitos básicos utilizados no restante do texto;
- No Capítulo 3 são apresentados e discutidos os trabalhos relacionados;
- No Capítulo 4 é apresentada a discussão sobre o desenvolvimento do Modelo de Referência para uma proposta de solução ao problema;
- No Capítulo 5, usando o Modelo de Referência é descrita a criação de uma Arquitetura de Referência para uma proposta de solução ao problema;
- No Capítulo 6 descreve-se a implementação de uma solução para servir como prova de conceito;
- No Capítulo 7 apresentamos uma discussão sobre os resultados alcançados e como podemos utilizar a solução desenvolvida para a criação de serviços inteligentes;
- No Capítulo 8 apresentamos as conclusões da tese.

## Parte II

### Referenciais teóricos

## 2 Fundamentação Teórica

Para uma adequada contextualização deste trabalho são necessárias algumas considerações sobre conceitos-chaves e temas abordados. Um dos objetivos deste capítulo é explicar em mais detalhes dois pontos importantes para o objeto desta Tese: O que é um serviço inteligente no contexto aqui adotado e o que são ambientes colaborativos flexíveis na *Web*. O outro objetivo é fazer uma ambientação sobre termos utilizados nas descrições e discussões.

Assim, abordamos a questão da definição do que vem a ser um serviço inteligente no nosso contexto, as principais técnicas de desenvolvimento de aplicações inteligentes que inspiraram os trabalhos realizados nesta tese, a descrição do modelo 3C de colaboração, um resumo sobre CSCW, a definição de Ambientes Colaborativos Flexíveis na *Web* e um resumo sobre o paradigma MOrFEU.

### 2.1 Serviço Inteligente

Todo sistema de informação fornece um ou mais serviços aos seus usuários, desde serviços simples (inserção de registros num banco de dados) até serviços mais complexos (busca por padrões interessantes em bancos de dados gigantescos). Alguns desses serviços podem ou não ser considerados inteligentes. A classificação em serviços inteligentes e não inteligentes pode ser um trabalho bastante complexo, devido principalmente a múltiplas interpretações do que vem a ser inteligência, inclusive porque tal questão apresenta aspectos subjetivos. Para a definição do termo “serviço inteligente” optamos por primeiro explicitar o significado que atribuímos a cada subtermo para depois descrever o termo completo.

#### 2.1.1 Definição do termo “serviço”

Para a definição de “serviço” emprestamos a definição usada no modelo de referência SOA ([MACKENZIE et al., 2006](#)): “Um serviço é um mecanismo que habilita o acesso a uma ou mais capacidades”. No caso, capacidade se refere a uma tarefa que é executada por algum sistema computacional. Outro ponto a ser observado é que segundo ([MACKENZIE et al., 2006](#)) o serviço se caracteriza por ter um provedor de serviço (quem realmente faz a tarefa) e um consumidor do serviço (quem precisa que a tarefa seja feita).

E para a definição de “tarefa” utilizamos a mesma definição utilizada por ([GHARSELLAOUI; BELLIK; JACQUET, 2012](#)), que diz: “Tarefas podem ser vistas como ações que devem ser executadas para se alcançar um objetivo em uma aplicação específica de um domínio”.

Para exemplificar as diferenças, apresentamos na Tabela 1 um exemplo do que seria uma “ação”, uma “tarefa” e um “serviço” no cenário de um corretor ortográfico.

Tabela 1 – Exemplos de ação, tarefa e serviço

Termo	Exemplo
Ação	Corrigir a ortografia de uma palavra.
Tarefa	Corrigir a ortografia de um documento.
Serviço	Oferecer para alguém a tarefa de corrigir a ortografia de um documento.

Fonte: Autor.

### 2.1.2 Definição do termo “inteligente”

Como não temos uma definição única sobre inteligência amplamente aceita na comunidade acadêmica e por precisamos de uma definição do que seria uma tarefa inteligente no contexto do trabalho aqui apresentado, utilizamos uma das definições de inteligência apresentada em (RUSSEL; NORVIG, 2013) que diz que “é inteligente se o comportamento é semelhante ao esperado de um comportamento humano ao desempenhar uma tarefa similar”.

Temos também definições para “Inteligência Artificial” que ajudam a entender o significado pretendido:

- “A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas.” (Kurzweil, 1990) apud (RUSSEL; NORVIG, 2013);
- “O estudo de como os computadores podem fazer tarefas que hoje são melhor desempenhadas pelas pessoas.” (Rich e Knight, 1881) apud (RUSSEL; NORVIG, 2013).

(STERLING; TAVETER, 2009) ao comentar sobre o atributo “inteligência” dos modernos ambientes computacionais, cita exemplos que ilustram bem o sentido de inteligência que queremos enfatizar:

- Quando um comercial de um produto afirma que seu produto é mais inteligente que os da concorrência, ele geralmente quer dizer que o seu produto tem mais funções do que o da concorrência;
- É inteligente quando o relógio de um sistema computacional se ajusta automaticamente ao período de horário de verão;
- É inteligente quando um *pendrive* ou uma câmera digital funciona sem problemas em uma variedade grande de computadores e sistemas operacionais.

Assim, aplicamos o adjetivo inteligente no mesmo sentido em que Turing o aplicou quando propôs o seu famoso teste de inteligência comportamental. Para passar no Teste de Turing um computador não precisaria ter inteligência verdadeira, porém ele precisaria parecer inteligente. Resumidamente, podemos afirmar que o trabalho realizado nesta tese segue a perspectiva conhecida como Inteligência Artificial “Fraca”.

### 2.1.3 Definição do termo “serviço inteligente”

Juntando as duas definições anteriores podemos trabalhar na definição do que consideramos como um serviço inteligente. Neste sentido, precisamos primeiro estabelecer o que é uma tarefa inteligente.

Tarefas inteligentes no contexto deste trabalho são: Ações executadas por um sistema computacional de forma semelhante ao que seria feito por um humano em igual contexto ao tentar alcançar um objetivo em uma aplicação específica de um domínio.

Portanto, serviço inteligente é a instanciação de uma tarefa inteligente dentro de um sistema computacional de forma a ser oferecido o seu acesso a outros sistemas computacionais.

## 2.2 Desenvolvimento de aplicações inteligentes

Para demonstrar a utilidade das ideias desenvolvidas neste trabalho é necessário fazer uma descrição de algumas das principais técnicas utilizadas no desenvolvimento de aplicações dentro da área da Inteligência Artificial (IA) que serviram de orientação para o desenvolvimento dos nossos trabalhos.

Segundo (RUSSEL; NORVIG, 2013) tem-se historicamente trabalhado com quatro estratégias para a criação de aplicações no campo da IA:

- **Pensar como um humano.** A estratégia é tentar replicar o funcionamento da mente humana.
- **Agir como um humano.** Aqui, tenta-se replicar o comportamento humano.
- **Pensar racionalmente.** A estratégia é utiliza-se da notação lógica para a resolução dos problemas.
- **Agir racionalmente.** Onde é utilizado a estratégia de criar e usar agentes racionais ou comunidades de agentes racionais para a resolução dos problemas.

A abordagem das ideias e conceitos apresentados neste trabalho se encaixa na classificação de “Agir como um humano” apesar de também utilizar conceitos e estratégias da classificação “Agir racionalmente”.

A seguir, descrevemos a Teoria de Agentes e a *Subsumption Architecture* pois foram as duas técnicas que mais inspiraram as ideias utilizadas no desenvolvimento da camada de serviços inteligentes.

### 2.2.1 Teoria de Agentes

Segundo (STERLING; TAVETER, 2009) o conceito de Agente em computação vem sendo discutido na área de IA desde os primeiros dias. Ele define um agente como “uma entidade que executa uma atividade específica dentro de um ambiente do qual ela tem consciência e que pode responder às mudanças nesse ambiente” e para ser considerado um agente inteligente ele deve ter as seguintes características:

- **Ser reativo.** Um agente deve perceber seu ambiente e responder em um tempo apropriado as mudanças que ocorrem nele.
- **Ser proativo.** Um agente deve não apenas responder às mudanças do ambiente como também poder agir oportunamente para atingir um objetivo ao tomar a iniciativa da ação.
- **Ser social.** Um agente deve interagir, quando apropriado, com outros agentes ou humanos com objetivo de resolver seus próprios problemas ou auxiliar outros em suas atividades.

(WOOLDRIDGE; JENNINGS, 1995) demonstra a importância da Teoria de Agentes dentro da IA ao informar que “É interessante notar que umas das formas de definir IA é como o subcampo da ciência da computação que objetiva construir agentes que exibam aspectos de comportamento inteligente”. Ele explica que a linha de pesquisa em agentes consegue criar aplicações inteligentes utilizando formalismos que capturam os relacionamentos entre os vários elementos que compõem o estado cognitivo dos agentes. Os agentes por sua vez, precisam ser conceitualizados ou implementados utilizando conceitos usualmente aplicados aos humanos tais como noções mentalistas (conhecimento, crença, intenção, obrigação, etc...) ou emocionais além de apresentarem as seguintes propriedades: autonomia, habilidades sociais, reatividade e pró-atividade.

Em (JENNINGS; WOOLDRIDGE, 1999) um agente é definido como “um sistema computacional situado em algum ambiente que seja capaz de ação autônoma para atingir seus objetivos”, sendo autonomia definida como a capacidade de agir sem a intervenção de humanos (ou outros agentes) juntamente com o controle das suas próprias ações e estado interno. E para ser considerado um agente inteligente, a autonomia do agente deve ser flexível, para isso o agente precisa ser: responsivo (reativo), pró-ativo e social.

Baseado nesses três trabalhos pode-se concluir que o desenvolvimento de um serviço inteligente poderia ser visto como o resultado do comportamento racional de um ou vários

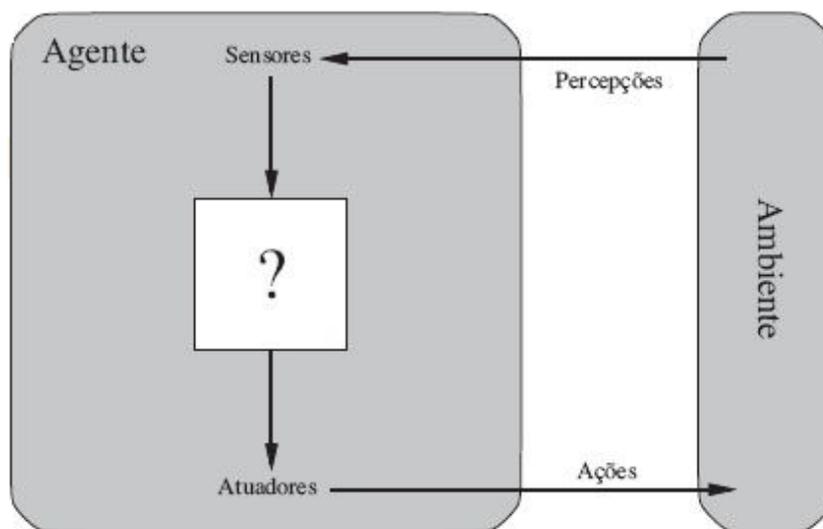
agentes em execução, em um determinado domínio. A desvantagem é que os projetistas das aplicações precisam ser treinados na utilização da técnica de agentes e caso seja utilizado para a criação de serviços inteligentes em sistemas que já estejam em uso será necessário a “agentificação” de vários processos já existentes no sistema.

### 2.2.1.1 Agentes

“Um agente é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores” (RUSSEL; NORVIG, 2013).

Um agente é “uma entidade que realiza uma atividade específica em um ambiente sobre o qual ele tem uma percepção e sobre o qual ele pode atuar em resposta a mudanças no mesmo.” (STERLING; TAVETER, 2009).

Figura 1 – Funcionamento de um Agente.



Fonte: (RUSSEL; NORVIG, 2013).

A Figura 1 ilustra a ideia de um Agente. Um Agente interage com o ambiente recebendo informações através dos seus sensores. Ele então processa a informação recebida dos sensores e decide uma ação ou série de ações a serem executadas no ambiente. Essas ações são então executadas pelos seus atuadores e elas podem modificar o ambiente ou modificar o funcionamento do ambiente.

Como exemplo de sensores de um agente de software para o recebimento de informações temos: leitura de uma base de dados, leitura dos registros de eventos de um ambiente, leitura dos dados de uma página *Web* e receber mensagens de outros agentes.

Como exemplo de atuadores de um agente de software para realizar ações em um ambiente podemos citar: escrita de um arquivo, registrar um evento no registro de eventos do ambiente e enviar mensagens para outros Agentes.

### 2.2.1.2 Sistemas Multiagente

De acordo com (STERLING; TAVETER, 2009), podemos dizer que em linhas gerais um sistema é um conjunto de entidades conectadas constituindo uma entidade complexa ou que desempenham uma função complexa e se vários ou todas as entidades forem agentes então nós temos um Sistema Multiagente (SMA).

Em um SMA o trabalho dos agentes segue a linha de que o todo é maior do que a soma das suas partes. (STERLING; TAVETER, 2009) afirma que vários agentes interagindo podem produzir um comportamento interessante que não era previsível.

A principal vantagem do uso de um SMA é dividir um problema complexo em várias subpartes independentes mas que ainda assim podem se relacionar de forma coordenada para a solução de um problema.

### 2.2.2 *Subsumption Architecture*

Uma técnica para a criação de aplicações inteligentes que inspirou as ideias apresentadas neste trabalho é a mostrada em (BROOKS, 1986) chamada *Subsumption Architecture* (SA). Ela foi desenvolvida inicialmente para o uso na construção de sistemas de controle de robôs, porém pode ser adaptada para outros contextos desde que o objetivo seja a construção de um comportamento inteligente. Nela a execução do serviço é feita através de três passos: criação dos níveis de competência, criação das camadas de controle e implementação das camadas.

Os níveis de competência são definições descrevendo classes de comportamentos esperados que a solução apresente (o robô, no caso de Brooks). Quanto maior o nível de competência mais específica deve ser a classe de comportamento. De forma que cada nível seja um subconjunto dos comportamentos do próximo nível superior. Para exemplificar, no caso de (BROOKS, 1986) ele descreveu os seguintes níveis para um robô autônomo inteligente:

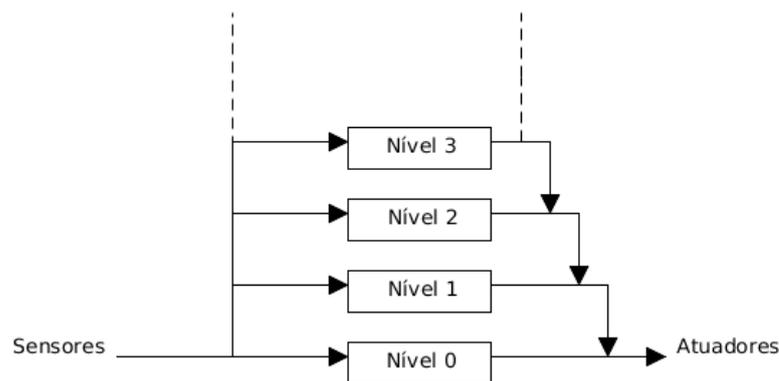
- **Nível 0:** Evitar entrar em contato com objetos (sejam objetos que se movem ou não)
- **Nível 1:** Andar aleatoriamente pelo ambiente evitando encostar em outros objetos.
- **Nível 2:** Explorar o ambiente, procurando locais distantes mas que sejam alcançáveis.
- **Nível 3:** Construir um mapa do ambiente e planejar rotas entre os locais
- **Nível 4:** Detectar mudanças de local em objetos que estavam parados
- **Nível 5:** Raciocinar sobre o ambiente em termos de objetos identificáveis e executar tarefas em objetos específicos.

- **Nível 6:** Formular e executar planos que envolvem mudar o estado do ambiente para um estado mais favorável.
- **Nível 7:** Raciocinar sobre o comportamento dos objetos do ambiente e modificar os planos de acordo.

Para cada nível de competência é então construída uma camada de controle. Todas as camadas recebem os mesmos dados de entrada e cada camada processa apenas os dados que precisa. Como cada camada pode alterar o funcionamento dos atuadores da camada abaixo dela (alterando ou suprimindo os comportamentos da camada abaixo), esta arquitetura acaba por apresentar as seguintes vantagens:

- Cada camada pode trabalhar paralela e independentemente, e portanto:
  - Aumenta a resiliência da solução, pois se uma camada falhar os comportamentos das outras camadas continuam funcionando;
  - Aumenta o desempenho da solução, pois todas as camadas podem começar o seu funcionamento sem esperar pelas camadas inferiores.
- Os comportamentos das camadas superiores tem sempre prioridade em relação as camadas inferiores.

Figura 2 – Subsumption Architecture.



Fonte: (BROOKS, 1986). Traduzido pelo autor.

A Figura 2 apresenta a ideia do funcionamento das camadas de controle. Os sensores enviam informações para todos os níveis ao mesmo tempo, cada nível processa as informações que forem relevantes para o seu funcionamento e envia o resultado para os atuadores, sendo que cada nível pode alterar o resultado que foi enviado pelos níveis abaixo do seu, alterando assim o resultado final.

Voltando então ao robô exemplo usado por Brooks, temos que o robô ao ser colocado em um novo ambiente vai inicialmente ficar parado para evitar colisões com outros objetos do ambiente, depois vai começar a andar aleatoriamente pelo ambiente com o objetivo de explorar e criar um mapa do ambiente, durante o seu deslocamento ele vai continuar evitando entrar em contato com outros objetos a menos que isso seja necessário para atingir um objetivo de uma camada maior (exemplo: passar por um corredor apertado). Após isso, ele vai começar a registrar as mudanças no ambiente e verificar se ele pode alterar o ambiente para melhorar seu próprio desempenho.

A implementação sugerida em (BROOKS, 1986) é que cada camada implementa uma máquina de estados finitos com a habilidade de guardar estruturas de dados. Assim conforme vai-se colocando as camadas de controle umas em cima das outras, as máquinas de estados finitos das camadas eram ligadas umas nas outras.

As principais vantagens da Subsumption Architecture são:

1. Um desenvolvimento iterativo da solução resolvendo os problemas usando a estratégia *bottom-up*;
2. Cada camada somente precisa processar os dados percebidos que são de seu interesse;
3. Um controle dos atuadores de forma distribuída e paralela.

Para o trabalho aqui descrito é interessante conhecer que os argumentos apresentados por Brooks para a criação da SA, conforme sumarizado em (WOOLDRIDGE; JENNINGS, 1995), foram:

1. Comportamento inteligente pode ser gerado sem uma representação explícita e sem o uso de raciocínio abstrato;
2. Inteligência é uma propriedade que emerge em certos sistemas complexos.

Por fim, a *Subsumption Architecture* ainda continua sendo usada nos dias atuais para a construção de soluções como demonstra o trabalho de (TURNER; GIVIGI; BEAULIEU, 2013) no controle de robôs em tempo real e o trabalho de (OLAND; ANDERSEN; KRISTIANSEN, 2016) na aplicação da SA no controle de voos de drones.

## 2.3 Modelo de Colaboração 3C

O modelo de colaboração 3C foi criado por (ELLIS; GIBBS; REIN, 1991) ao enfatizar a importância da comunicação, da colaboração e da coordenação para o sucesso de um *groupware*. Ele explica que a comunicação mediada por computador ainda não está

bem integrada com outras formas de comunicação e que isto faz com que existam lacunas entre as comunicações síncronas e as comunicações assíncronas, porém com o avanço da integração da telecomunicação com o processamento computacional essas lacunas devem diminuir ou sumir.

([ELLIS; GIBBS; REIN, 1991](#)) explica ainda que a colaboração, assim como a comunicação, é essencial para a realização das atividades em grupo. Indo mais além ele informa que para uma efetiva colaboração as pessoas precisam compartilhar informações entre si e que idealmente o ambiente (compartilhado pelo grupo) deveria atualizar as informações do contexto do grupo automaticamente além de notificar os usuários sobre as ações dos outros usuário, sempre que esta notificação for relevante.

Por fim, ([ELLIS; GIBBS; REIN, 1991](#)) explica que a coordenação tem a capacidade de aumentar a eficiência da comunicação e da colaboração e que a coordenação é uma necessidade quando várias partes estão envolvidas na execução de uma mesma tarefa.

O modelo de colaboração 3C foi aprimorado por ([FUKS et al., 2007](#)) que troca “colaboração” no modelo original por “cooperação” e explica que o suporte computacional na colaboração acontece através de ciclos iterativos entre os 3Cs do modelo (comunicação, coordenação e cooperação). Ou seja, para ocorrer a colaboração em um grupo é necessário que os membros do grupo participem de atividades de comunicação, coordenação e cooperação uns com os outros, emergindo desse conjunto de atividades a colaboração. Tal exemplo do modelo aplicado ao trabalho em grupo foi utilizado por ([FUKS et al., 2007](#)) para a construção da Figura 3, que representa as interações entre os diversos conceitos envolvidos na obtenção da colaboração.

Figura 3 – Representação do Modelo 3C.



Fonte: ([FUKS et al., 2007](#)).

Neste contexto, enquanto os membros do grupo se coordenam eles resolvem os conflitos e organizam as atividades de forma a prevenir uma interrupção seja na comunicação seja na cooperação dentro do grupo. A cooperação ocorre quando os membros do grupo compartilham o espaço e as ferramentas durante a execução das suas tarefas. No entanto o surgimento de situações não previstas vai demandar um novo ciclo de comunicação que

por sua vez vai requerer um novo ciclo de coordenação para reorganizar as tarefas que serão executadas durante o ciclo de cooperação.

Portanto, para (FUKS et al., 2007):

- Comunicação está relacionado com a troca de mensagens e informações entre as pessoas;
- Coordenação está relacionado com o gerenciamento das pessoas, suas atividades e os recursos disponíveis;
- Cooperação ocorre quando as pessoas trabalham em um mesmo ambiente compartilhado entre elas;
- Colaboração acontece como o resultado da aplicação dos três conceitos anteriores num mesmo ambiente.

### 2.3.1 Os termos cooperação e colaboração neste trabalho

Em alguns trabalhos como o de (FUKS et al., 2007), é feita uma distinção clara entre o significado dos termos colaboração e cooperação, enquanto em vários outros eles acabam sendo utilizados de forma intercambiável, principalmente devido à dificuldade em classificar se o que ocorreu foi cooperação ou colaboração. Tal problema é bem descrito no trabalho apresentado em (KEMCZINSKI et al., 2008) que discute a pertinência, concorrência e/ou complementaridade entre as diferentes visões do que vem a ser colaboração e cooperação.

No restante desse trabalho não nos preocupamos na distinção entre os termos, mas apenas em qualifica-los no sentido mais abrangente, aquele da palavra “colaboração” do modelo de colaboração 3C (FUKS et al., 2007).

## 2.4 CSCW

A complexidade e o tamanho das tarefas do mundo atual ressaltou a necessidade das pessoas interagirem entre si para a execução daquelas tarefas. O aumento da interação entre as pessoas levou então a necessidade de melhores ferramentas para facilitar a colaboração entre as pessoas de um mesmo grupo, dando origem a área de CSCW.

CSCW é a sigla para “*Computer Supported Cooperative Work*” que em português podemos traduzir como “Trabalho Cooperativo Auxiliado por Computador” ou “Sistema de Trabalho Cooperativo apoiado por Computador”. É uma área multidisciplinar de pesquisa sobre a utilização de ferramentas computacionais para auxiliar um grupo (de pessoas) a realizar as suas tarefas de forma a atingir um objetivo. Avanços em CSCW são interessantes e vêm sendo recorrentemente buscados por indivíduos e grupos.

Nesta seção vamos apresentar um resumo de seu histórico e principais conceitos.

### 2.4.1 Histórico do CSCW

Segundo (BORGES; CAVALCANTI; CAMPOS, 1995) em meados dos anos 70 surgiu a área precursora do CSCW, a Automação de Escritório, que buscava transformar aplicações mono-usuário, principalmente processadores de texto e planilhas eletrônicas, em aplicações que poderiam ser utilizadas simultaneamente por um grupo de usuários.

Com o avanço dos estudos, outras áreas de pesquisa (tais como: sociologia, psicologia, antropologia e educação) foram incluídas na tentativa de melhorar o suporte ao trabalho colaborativo. Assim, o termo Automação de Escritório caiu em desuso e o termo CSCW passou a ser amplamente utilizado (GRUDIN; POLTROCK, 2012), sendo criada uma conferência com o mesmo nome em 1986 organizada pela ACM. Desde então vários eventos e periódicos vêm sendo organizadas em torno da área.

### 2.4.2 CSCW e Groupware

Apesar de por vezes serem mencionadas como sinônimos, temos que groupware são as tecnologias (tanto de hardware como de software) geradas pela pesquisa em CSCW (BORGES; CAVALCANTI; CAMPOS, 1995; ELLIS; WAINER, 1999; PENICHER et al., 2007). Assim, por exemplo, sistemas de correio-eletrônico, sistemas de suporte à decisão e editores de texto colaborativos podem ser classificados como groupware, enquanto a pesquisa realizada neles com objetivo de melhorar a comunicação entre os membros de um grupo pertence a área de CSCW.

Portanto apesar dos termos “*groupware*” e “CSCW” serem utilizados dependendo do enfoque que se queira dar (artefato X pesquisa), no decorrer deste trabalho optamos por utilizar o termo CSCW sempre que possível.

### 2.4.3 Características de um CSCW

De acordo com a literatura (ELLIS; GIBBS; REIN, 1991; ELLIS; WAINER, 1999; BORGES; CAVALCANTI; CAMPOS, 1995; GRUDIN; POLTROCK, 2012) podemos elencar as seguintes características dos CSCW:

- Prover um contexto compartilhado entre os usuários do ambiente;
- Classificar os usuários em papéis dependendo da sua função;
- Facilitar a cooperação sem impor uma forma única de trabalho;
- Facilitar a comunicação entre os participantes;

- Gerenciar os dados e informações sobre os trabalhos executados.

#### 2.4.4 Taxonomia dos CSCW

A clássica classificação utilizada para as aplicações de CSCW é a matriz de espaço-tempo proposta por Johansen em 1988 (PENICHER *et al.*, 2007) mostrada na Tabela 2, que classifica CSCW de acordo com o uso deles no espaço-tempo.

Tabela 2 – Matriz de espaço-tempo proposta por Johansen.

	<b>Mesmo Tempo (Síncrono)</b>	<b>Diferentes Tempos (Assíncrono)</b>
<b>Mesmo Espaço (Presencial)</b>	Interações Face-a-Face (Ex: Sala de Aula)	Interação Assíncrona (Ex: No carro, indo/voltando do trabalho)
<b>Diferentes Espaços (Remoto)</b>	Interação Síncrona Remota (Ex: Desktop Remoto)	Interação Assíncrona Distribuída (Ex: Email e Newsgroups)

Fonte: (PENICHER *et al.*, 2007). Traduzido pelo autor.

No entanto com a passagem do tempo as aplicações CSCW ficaram mais complexas e novas formas de classificação surgiram. Por exemplo, uma extensão da matriz proposta por Johansen foi feita por Grudin em 1991 (GRUDIN; POLTROCK, 2012) ao subdividir as categorias “Diferentes Espaços” e “Diferente Tempos” em categorias conforme mostrado na Tabela 3.

Tabela 3 – Extensão da matriz de espaço-tempo proposta por Grudin.

	<b>Mesmo Tempo</b>	<b>Tempo Diferente, Mas Previsível</b>	<b>Tempo Diferente e Imprevisível</b>
<b>Mesmo Local</b>	Sala de Aula	No carro, indo/voltando do trabalho.	Salas de Reunião
<b>Local Diferente, Mas Previsível</b>	Desktop Remoto	Email	Escrita Colaborativa
<b>Local Diferente e Imprevisível</b>	Seminários Interativos	Newsgroup	Workflow

Fonte: (GRUDIN; POLTROCK, 2012). Traduzido pelo autor.

Outras extensões foram propostas, porém percebeu-se o esgotamento da matriz de espaço-tempo, pois os CSCW passaram a englobar vários tipos de funcionalidades ao mesmo tempo. Por exemplo, um CSCW ao implementar funções de chat e email no mesmo sistema vai poder realizar tanto iterações síncronas como assíncronas.

Assim, surgiram classificações baseadas nas funcionalidades do CSCW, (PENICHER *et al.*, 2007) propôs uma classificação baseada nas características de compartilhamento de informação, comunicação e coordenação do CSCW além das possibilidades de espaço-tempo conforme sumarizado na Tabela 4, da qual retiramos as combinações que não poderiam ser aplicadas.

Tabela 4 – Classificação de CSCW proposta por Penichet.

Tipo	Características do CSCW			Tipo	Tempo		Espaço	
	Compartilhamento de Informação	Comunicação	Coordenação		Síncrono	Assíncrono	Mesmo	Diferente
<b>A</b>	Não	Não	Sim	<b>5</b>	Não	Sim	Não	Sim
<b>B</b>	Não	Sim	Não	<b>6</b>	Não	Sim	Sim	Não
<b>C</b>	Não	Sim	Sim	<b>7</b>	Não	Sim	Sim	Sim
<b>D</b>	Sim	Não	Não	<b>9</b>	Sim	Não	Não	Sim
<b>E</b>	Sim	Não	Sim	<b>10</b>	Sim	Não	Sim	Não
<b>F</b>	Sim	Sim	Não	<b>11</b>	Sim	Não	Sim	Sim
<b>G</b>	Sim	Sim	Sim	<b>13</b>	Sim	Sim	Não	Sim
				<b>14</b>	Sim	Sim	Sim	Não
				<b>15</b>	Sim	Sim	Sim	Sim

Fonte: (GRUDIN; POLTROCK, 2012). Traduzido pelo autor.

Assim, a classificação B-5 se refere a um sistema com apenas a característica de comunicação ocorrendo em tempo assíncrono e espaços diferentes (Ex: Email). Enquanto um sistema G-15 apresenta todas as características e elas ainda podem ocorrer de forma síncrona ou assíncrona, sendo ou não no mesmo espaço.

Resumos de outros exemplos de classificações que foram usadas na literatura podem ser encontrados em (GRUDIN; POLTROCK, 2012).

#### 2.4.5 Uso de IA em CSCW

Segundo (GRUDIN; POLTROCK, 2012), a utilização de Inteligência Artificial (IA) em CSCW começou em 1988 tendo como ênfase o uso de sistemas multiagente e mais tarde passando a ênfase para sistemas de recomendação.

Em 1991 (ELLIS; GIBBS; REIN, 1991) descreveu a IA como uma das principais disciplinas responsável pelo sucesso da área de CSCW, afirmando que o relacionamento entre IA e CSCW era mutuamente benéfico, pois enquanto por um lado a IA melhora o entendimento e funcionamento dos ambientes CSCW por outro lado a IA se beneficia com os desafios e tópicos de pesquisa que a CSCW fornece que de outra forma talvez nunca fossem explorados. (ELLIS; GIBBS; REIN, 1991) afirma ainda que a IA poderia no futuro prover as mais significativas contribuições para a área de CSCW.

Para (ELLIS; WAINER, 1999) a Teoria de Agentes é útil para a criação de funcionalidades subjacentes do CSCW ao usar componentes de software inteligentes ou semi-inteligentes para a construção de funções especializadas ou para apoiar a dinâmica de um grupo. Ele afirma que frequentemente é necessário a construção de módulos para a execução de subtarefas que requerem conhecimento especializado de um domínio específico. Como exemplo, ele aponta um “mediador social” usado em um encontro virtual. Ele classifica ainda os tipos de agentes criados para ambiente CSCW em três tipos: agente autônomo (trabalham normalmente sozinhos em uma subtarefa independente), agente de

usuário (que interagem e trabalham para apenas um usuário do grupo) e agentes de grupo (que interagem e trabalham com vários usuários do grupo).

#### 2.4.6 A Web e CSCW

Com o surgimento da *Web* e a sua posterior disseminação, criando a *Web Social* e a *Web Semântica* as investigações na área de CSCW passaram cada vez mais da infraestrutura de rede para os ambientes colaborativos virtuais e para os ambientes colaborativos na web (CORREIA; FONSECA; PAREDES, 2011; GRUDIN; POLTROCK, 2012).

A *Web Social* vem fazendo a “Internet criar novas construções sociais, comunidades de utilizadores com características distintas que agem em torno de um objetivo comum” (CORREIA; FONSECA; PAREDES, 2011). Tais comunidades demonstraram o potencial da “inteligência coletiva” que estava sendo criada através da Internet. Ao mesmo tempo a *Web Semântica* vem permitindo aos computadores e humanos trabalharem em cooperação.

Portanto podemos afirmar que os avanços da *Web* abriram todo um novo leque de oportunidades de pesquisas para a área de CSCW e que as novas ferramentas desenvolvidas já contam com a *Web* como um bloco de construção já concluído.

#### 2.4.7 CSCL e CSCW

Aprendizagem Colaborativa Apoiada por Computador (*Computer Supported Collaborative Learning* - CSCL) é um campo de pesquisa multidisciplinar que estuda como acontece e como pode melhorar o processo de ensino-aprendizagem que ocorre com grupos de pessoas dentro de um ambiente computacional (STAHL; KOSCHMANN; SUTHERS, 2006).

Segundo (BORGES; CAVALCANTI; CAMPOS, 1995) a interdisciplinaridade de CSCW juntamente com a grande quantidade e variedade dos trabalhos desenvolvidos acaba por fazer com que apareçam novas vertentes, como seria o caso da área de CSCL. Assim, os ambientes CSCL seriam uma subdivisão dos ambientes CSCW pois tem as mesmas características básicas de trabalho cooperativo, porém em CSCL temos uma intencionalidade na aprendizagem e na tutoria.

(CARNEIRO; GELLER; TAROUÇO, 2002) explica que com a ampliação da Internet e a implantação de Intranets os conceitos e ferramentas desenvolvidos em CSCW passaram também a ser aplicados na área educacional. No começo os primeiros ambientes eram usados apenas como repositórios para o material educacional produzido pelos professores, passando depois a focar na concepção, desenvolvimento e utilização de ferramentas para apoiar o processo de ensino-aprendizagem. Assim a área de CSCW incorporou características desenvolvidas em campos como a Educação a Distância e deu origem ao CSCL. (STAHL; KOSCHMANN; SUTHERS, 2006) corrobora esta ideia ao afirmar que os

ambientes CSCL tiveram um crescimento nos anos 1990 devido a sua contraposição aos softwares que forçavam os alunos a aprenderem de forma isolada dos outros alunos.

Para (ARRUDA; PIMENTEL, 2001) os ambientes CSCL tem elementos motivadores para o processo de ensino-aprendizagem, dentre os quais ele destaca:

- Maior participação dos alunos pois no ambiente online, mesmo os alunos que são tímidos conseguem expor as suas ideias;
- Melhor acompanhamento dos alunos pelos professores baseado nos dados armazenados no ambiente;
- Maior interação dos alunos em uma interação conjunta na construção do conhecimento.

Em português os CSCL são também representados por Ambientes Virtuais de Aprendizagem (AVA) que tenham suporte a colaboração.

Para maiores detalhes sobre o histórico de CSCL recomendamos a leitura do artigo (STAHL; KOSCHMANN; SUTHERS, 2006) onde é feito um levantamento histórico, descrição do surgimento de CSCL a partir dos trabalhos em CSCW, como foi a incorporação de técnicas de uso do computador para a educação dentro do CSCL, mudança de foco do indivíduo para o grupo e mudança de representação mental para construção colaborativa do conhecimento.

Durante o desenvolvimento deste projeto foram utilizadas pesquisas tanto em CSCW como em AVA/CSCL.

## 2.5 Ambientes Colaborativos na *Web*

Quando o termo CSCW foi criado, ele unificou uma quantidade significativa de outros termos, dentre os quais o de Ambientes Colaborativos (GRUDIN; POLTROCK, 2012). Assim podemos utilizar Ambientes Colaborativos como um sinônimo de CSCW, desde que façamos algumas ressalvas. No restante deste texto, o termo Ambientes Colaborativos se refere aos ambientes colaborativos onde o processo de colaboração acontece conforme descrito no Modelo 3C, com a utilização de recursos computacionais.

No entanto, para os trabalhos descritos nesta tese não estamos interessados em todos os tipos de Ambientes Colaborativos, mas somente naqueles em que os recursos computacionais são as tecnologias presentes e disponíveis na *Web*.

## 2.6 Ambientes Colaborativos Flexíveis na *Web*

De acordo com (MENEZES et al., 2008; SANTOS; CASTRO; MENEZES, 2012; SANTOS; CASTRO; MENEZES, 2010) os Ambientes Colaborativos Flexíveis surgiram a partir da necessidade dos usuários de Ambientes Colaborativos adaptarem as ferramentas disponibilizadas nos mesmos para as suas necessidades. Muitos dos pedidos de alteração de ferramentas feitos pelos usuários, normalmente requerem a intervenção de um programador e várias horas de serviço para entrega de cada pedido. Tal problema acaba por obrigar os usuários dos Ambientes Colaborativos a utilizarem as ferramentas do ambiente apenas da forma como foram planejadas inicialmente, além do que apenas através de ciclos demorados de desenvolvimento, se poderia personalizar o ambiente às suas necessidades, ou seja, podemos afirmar que os ambientes tradicionais limitam as possibilidades dos seus usuários ao previamente planejado.

(MENEZES et al., 2008) afirma que várias das ferramentas presentes nos ambientes colaborativos são inadequadas ao uso em ambientes de aprendizagem colaborativa devido principalmente a pouca flexibilidade apresentada por essas ferramentas. Como exemplo, ele descreve a dificuldade de conseguir apoiar um método pedagógico chamado “Controvérsia Acadêmica” dentro de um ambiente de aprendizagem quando o ambiente não foi inicialmente planejado para utilizar tal método.

Neste sentido, surgiram os Ambientes Colaborativos Flexíveis na *Web* que possuem como principal característica a capacidade de permitir alterações em tempo de execução das suas estruturas, funcionalidades e interfaces para melhor atender ao usuário. Outras características notadas também são:

- Favorecem o protagonismo e a autoria individual e coletiva através da oferta de várias formas de organizar as interações e produções feitas pelos seus usuários;
- Favorecem o autoacompanhamento através de uma visão de totalidade das produções individuais;
- Favorecem a autoria coletiva através da interação e agilidade no acesso às produções de todos os participantes do ambiente;
- Se pautam pelos seguintes aspectos: plasticidade, ergonomia, redução da repetição do trabalho e redução da sobrecarga cognitiva.

Ambientes que apresentam essas características permitem então que as ações executadas pelos seus usuários não necessariamente sigam a “ordem lógica”, ou seja, as ações não teriam entre elas uma “barreira ou parede virtual” (MENEZES et al., 2008). Assim esses ambientes permitem uma maior liberdade em sua utilização.

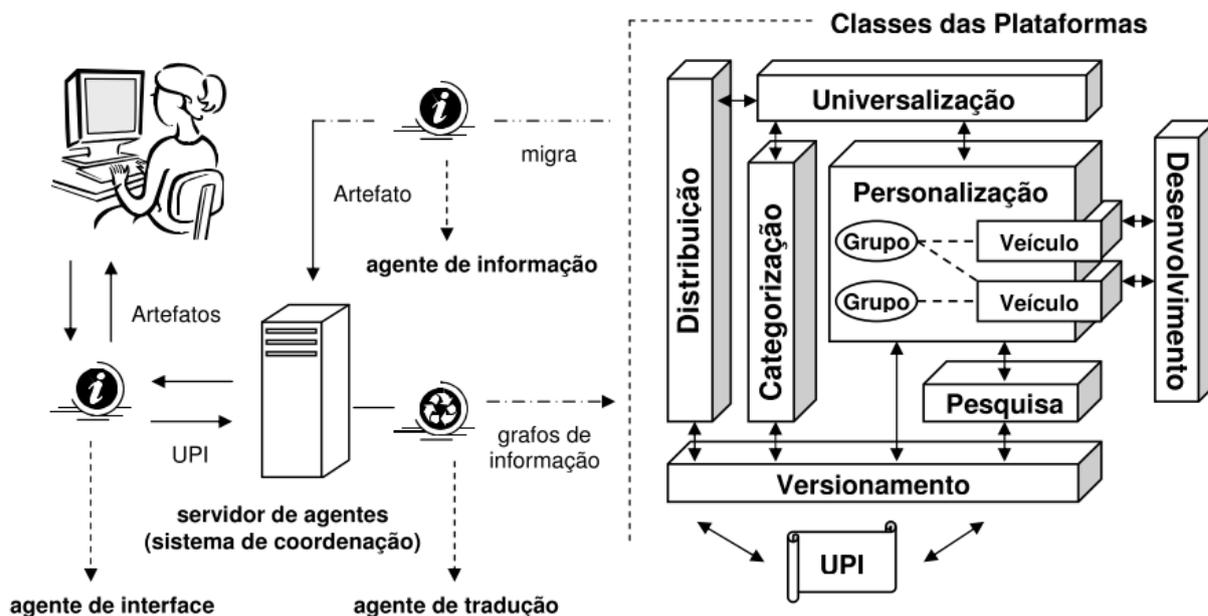
No trabalho aqui proposto é utilizado o MORFEU como uma instância específica de um Ambiente Colaborativo Flexível na *Web*.

## 2.7 MORFEU

O MORFEU (MENEZES et al., 2008) é um ambiente pautado pelos aspectos de: plasticidade, ergonomia, redução da repetição de trabalho e redução da sobrecarga cognitiva. Um de seus objetivos é fazer com que a apropriação do ambiente pelos seus usuários ocorra de forma mais rápida e efetiva. MORFEU é um acrônimo para **M**ulti-**O**rganizador **F**lexível para **E**spaços Virtuais.

Segundo (BELTRAME; MONTEIRO; CURY, 2008) o MORFEU tem como um de seus objetivos desenvolver um ambiente *Web* que gerencie os espaços virtuais coletivos, não limitando a criatividade dos seus usuários à percepção inicial dos programadores. Para isso o MORFEU constrói um novo paradigma para a criação de espaços virtuais e socialização do conhecimento, sendo sua principal característica a flexibilidade (na apresentação dos elementos de interface, na criação de novas ferramentas, e nas listas de acesso).

Figura 4 – Esboço informal da arquitetura do MORFEU.



Fonte: (BELTRAME; MONTEIRO; CURY, 2008).

Conforme o esboço da arquitetura mostrado na Figura 4 as características do MORFEU podem ser organizadas em diferentes aspectos, conforme segue:

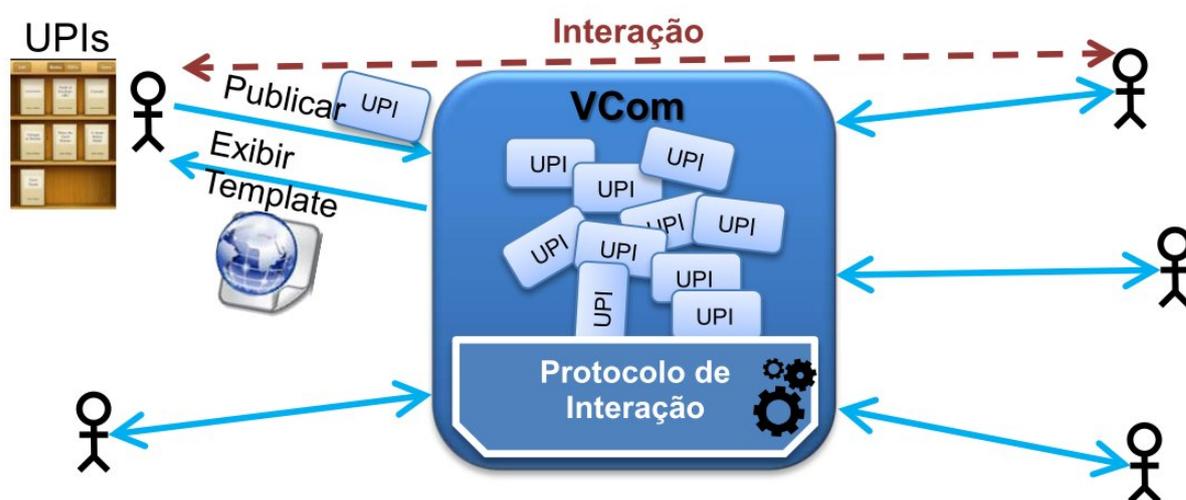
- **Desenvolvimento:** Criação de novos ambientes virtuais coletivos e flexibilização da sua organização. Os espaços criados são chamados de Veículos de Comunicação.

- **Personalização:** Criação de grupos, organização dos veículos de publicação nos grupos e criação de perfis coletivos.
- **Pesquisa:** Busca e recuperação de Unidades de Produção Intelectual (UPI).
- **Categorização:** Indexação e criação de índices remissivos sobre as UPIs.
- **Universalização:** Promoção de agentes inteligentes para tradução, mecanismos de flexibilização estrutural, apoio a inter-semiótica.
- **Distribuição:** Canais de distribuição das UPIs.
- **Versionamento:** Gerenciamento de versões para as UPIs.

Tal organização garante a facilidade de integração dos diversos elementos da infraestrutura além da reutilização de componentes (BELTRAME; MONTEIRO; CURY, 2008).

### 2.7.1 Visão Geral do MOrFEU

Figura 5 – Funcionamento de um VCom no MOrFEU.



Fonte: (SANTOS, 2013).

Na Figura 5 temos ilustrado o funcionamento conceitual do MOrFEU. A Unidade de Produção Intelectual (UPI) é a unidade básica e o seu ponto central. As UPI são organizadas em Documentos que são instâncias de Veículos de Comunicação (VCom), podendo ter UPIs de mais de um usuário. O ato de incluir uma UPI em um documento é chamado de Publicar a UPI. Ao conjunto de regras que explicitam como um documento pode ser gerado, damos o nome de Protocolo de Interação. Finalmente, a visualização dos VCom é feita através de Templates que definem quais partes do documento serão geradas.

A seguir descrevemos os principais elementos do MOrFEU.

## 2.7.2 Unidade de Produção Intelectual

Um dos elementos básicos do MORFEU é a Unidade de Produção Intelectual (UPI). Ela é utilizada para armazenar os trabalhos realizados pelos usuários. Segundo (SANTOS; CASTRO; MENEZES, 2010) a UPI é composta por três elementos: autor, título e conteúdo da UPI. Sendo o conteúdo básico de uma UPI formado de textos escritos em HTML assim, uma UPI pode usar várias mídias e fazer uso de links para referenciar outras UPIs ou URLs. Outros conteúdos possíveis são: imagens, vídeos, documentos XML, códigos-fonte, etc.

Cada UPI também é versionada pelo próprio ambiente conforme é alterada pelo usuário. A principal vantagem da UPI é que ela pode ser reutilizada em vários locais e não apenas no seu local de criação. Elas são associadas ao usuário autor da UPI de forma que mesmo que o local onde uma UPI tenha sido publicada seja apagado a UPI em si não é apagada.

O processo de publicação de uma UPI é ilustrado na Figura 6 e ocorre quando um usuário fornece os dados da UPI enquanto o sistema informa o esquema de publicação da UPI e um “XPath” para o processador de publicação, o esquema de publicação (código XML) é então inserido no Documento do VCom no local indicado pelo XPath e a UPI é registrada no banco de dados.

Figura 6 – Processo de publicação de um UPI.



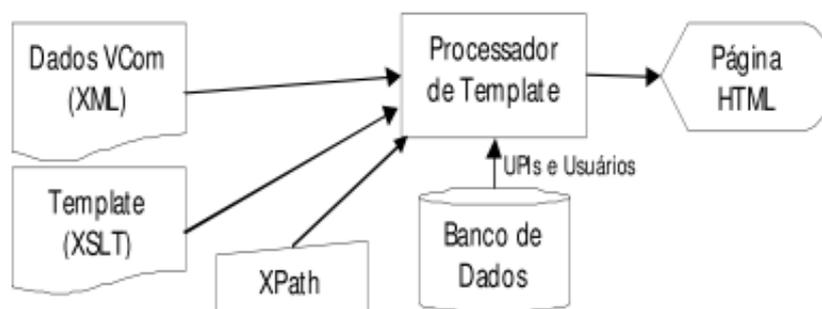
Fonte: (SANTOS; CASTRO; MENEZES, 2010).

## 2.7.3 Veículo de Comunicação

Os Veículos de Comunicação (VCom) podem ter vários tipos e cada tipo possui sua própria diretriz de composição. Fórum, chat e blog são exemplos comuns de VCom que podem existir em um ambiente. Os usuários podem criar novos tipos de VCom com características e estruturação próprias. Isto torna o MORFEU bastante flexível e portanto extensível às necessidades dos seus usuários.

No MORFEU a visualização dos dados de um VCom é feita usando XML juntamente com Templates usando XSLT para processar o XML e gerar uma página HTML.

Figura 7 – Processo de visualização de um VCom.



Fonte: (SANTOS; CASTRO; MENEZES, 2010).

Na Figura 7 temos uma ilustração de como ocorre o processo de visualização de um VCom no MORFEU. No início do processo, o XML que representa os dados do VCom é carregado juntamente com o XSLT que representa o Template além de um XPath. Após esses carregamentos os dados da UPI e do usuário são recuperados do banco de dados. Por fim, o XSLT do Template é processado e uma página HTML é retornada como resultado.

#### 2.7.4 Template

Segundo (SANTOS; CASTRO; MENEZES, 2010) os Templates podem ser modificados sempre que necessário para mudar a forma de visualização da página ou adicionar novas funcionalidades ao VCom. Caso necessário, mais de um Template pode ser associado ao mesmo VCom como também um Template pode incluir o uso de outros Templates.

#### 2.7.5 Documento

Segundo (SANTOS, 2013) as UPI podem ser agrupadas em um artefato criado coletivamente chamado Documento, que é uma instância de um VCom. Exemplificando, temos que enquanto o chat é um VCom uma sessão específica de um chat seria um Documento. A formação do Documento segue um conjunto de regras definidas pelo Protocolo de Interação. Por fim, para gerar as visualizações do Documento são utilizados os Templates associados ao VCom.

### 2.8 Conclusão do Capítulo

Neste capítulo, foi explicado que para um serviço fornecido ser considerado inteligente é necessário que a tarefa que este serviço vai executada também seja inteligente. E para ser inteligente a tarefa a ser executada deve ser uma que exige inteligência na sua execução quando ela é realizada por um humano.

Assim, para a construção de uma camada de serviços inteligentes o trabalho desenvolvido nesta tese de doutorado utilizou duas das principais técnicas de Inteligência Artificial: A Teoria de Agentes e a *Subsumption Architecture*. A Teoria de Agentes faz uso de agentes inteligentes que não apenas respondem em tempo apropriado as mudanças ocorridas em seu ambiente como também podem tomar a iniciativa de uma ação necessária para conseguir obter os resultados esperados, como por exemplo, entrar em contato com outros agentes do ambiente com o objetivo de conseguir negociar soluções conjuntas para os seus problemas. A *Subsumption Architecture*, por sua vez, para criar sistemas inteligentes segue três passos: primeiro defini-se os níveis de competência esperados para o sistema, depois especifica-se as camadas de controle para cada nível de competência e por fim implementa-se as camadas. Através da utilização dos níveis de competência a *Subsumption Architecture* consegue construir comportamentos que são considerados inteligentes.

Também foi apresentado o Modelo de Colaboração 3C que explica que a colaboração é o resultado de um processo iterativo que envolve atividades de comunicação, coordenação e cooperação entre os membros de um grupo. O Modelo de Colaboração 3C é utilizado no restante desta tese para realizar a desambiguação entre cooperação e colaboração, ou seja, apesar do uso dessas palavras serem intercambiáveis no restante da tese o sentido que se deve atribuir para a palavra é o mesmo que o sentido da palavra colaboração tem no Modelo de Colaboração 3C.

Outro ponto importante descrito neste capítulo é sobre os ambientes CSCW. Neste capítulo, apresentamos um resumo da história do CSCW pontuando que ele surgiu devido a necessidade dos usuários por uma interação melhor com outros usuários para a realização de trabalhos em grupo. Foram apresentadas as principais taxonomias utilizadas para classificar ambientes CSCW, e foram descritos os relacionamentos existentes entre CSCW, Inteligência Artificial e a *Web*. Também foi mostrado o relacionamento existente entre ambientes CSCW e CSCL, onde pode-se afirmar que o conjunto dos ambientes CSCL é um subconjunto dos ambientes CSCW.

Por último, foi descrito a classe de ambientes que é o objeto de estudo desta tese de doutorado: O Ambiente Colaborativo Flexível na *Web* (ACFW). Os ACFW podem ser classificados como um subconjunto dos CSCW que tem uma interseção com os CSCL. A flexibilidade desses ambientes é no sentido que eles podem alterar tanto o seu funcionamento como as suas interfaces e relacionamentos em tempo de execução. Essa flexibilidade tem por objetivo dar uma maior liberdade para que o usuário possa usar o ambiente da melhor forma. Ao final, descreveu-se o MORFEU que é uma instância de um ACFW. O MORFEU faz uso dos conceitos de Veículo de Comunicação (VCom) e Unidade de Produção Intelectual (UPI) para poder implementar as idéias e conceitos referentes a classe dos Ambientes Colaborativos Flexíveis na *Web*.

## 3 Trabalhos Relacionados

A ideia de adicionar serviços inteligentes aos ambientes colaborativos na *Web* não é nova, pois como será demonstrado já existem trabalhos sobre essa temática, no entanto nestes trabalhos a preocupação não é com a forma em si ou com o processo de criação de um serviço inteligente mas sim com a resolução do problema objeto do trabalho. Portanto, denota-se a necessidade de pesquisar e trabalhar para encontrar uma forma de simplificar a criação de serviços inteligentes em um ambiente colaborativo na *Web* e especificamente em Ambientes Colaborativos Flexíveis na *Web* (ACFW), que é o objeto de estudo deste projeto.

Como primeiro passo para a criação de um *framework* para prover serviços inteligentes em ACFW buscou-se na literatura por relatos de resolução deste tipo de problema ou problemas correlatos. No caso dos ACFW, além de não existir uma grande quantidade de relatos no tema, também não foram encontrados relatos de outras pesquisas envolvidas com a construção de serviços inteligentes em tal contexto.

Assim, descrevemos a seguir os resultados das investigações realizadas.

### 3.1 Problemas Correlatos

Nesta seção fazemos uma descrição de trabalhos abordando a temática da criação de serviço inteligente em ambientes colaborativos na *Web* e outros trabalhos relacionados com a área.

#### 3.1.1 Serviços Inteligentes

Percebe-se na literatura um esforço para a criação de ferramentas fornecedoras de serviços inteligentes aos usuários de ambientes CSCW e CSCL com o objetivo de aumentar a eficácia dos ambientes onde essas ferramentas estão inseridas.

Como um primeiro exemplo temos o TutorICC, apresentado em (PÍCCOLO *et al.*, 2011), e o seu uso no ensino de programação. O ambiente onde o TutorICC está inserido é um AVA específico para o ensino de programação e o TutorICC funciona como um Sistema Tutor Inteligente (STI) auxiliando o aluno no processo de aprendizagem de programação.

Outro exemplo é a ferramenta STCEQ.Net, demonstrado em (SANTOS; REIS, 2005), onde agentes inteligentes foram utilizados para implementar um AVA e um STI para funcionar como um Sistema Especialista na recomendação de soluções dos problemas apresentados aos alunos durante o aprendizado de Controle Estatístico de Qualidade.

Um caso de um serviço inteligente inovador é o apresentado em (BITTENCOURT; COSTA, 2011) onde é descrito a utilização da ferramenta Risys para gerência de riscos em projetos executados em ambientes colaborativos. O Risys utiliza um Sistema Especialista para recomendar ações para gerenciamento de risco em projetos executados em ambientes colaborativos. A inovação está em trazer para um ambiente colaborativo um serviço disponível em ambientes não colaborativos.

Outra ferramenta relacionada a execução de projetos é a apresentada em (DANTAS et al., 2008) onde temos a descrição do funcionamento dela no auxílio na execução de projetos através de um Ambiente Colaborativo na *Web*. A ferramenta avalia as contribuições de membros dentro do ambiente e ajuda ao acompanhar os membros do grupo quanto a sua contribuição no projeto, podendo avisar ao coordenador em caso de desvios.

Já em (FILHO; QUARTO; FRANCA, 2010) é descrito uma metodologia para utilização em AVAs na criação de grupos socioafetivos com o objetivo de auxiliar o mediador. O serviço inteligente é a formação de grupos de alunos de forma inteligente afim de melhorar o funcionamento do AVA.

Outra ferramenta proposta na literatura é a MAE (agente Mediador da Autoeficácia, descrita em (BICA; VERDIN; VICARI, 2006)) com o objetivo de melhorar o desempenho dos STIs utilizados em AVAs.

Um outro esforço para melhorar o desempenho dos AVAs é descrito em (KUNZEL et al., 2011) onde foi desenvolvido um assistente pessoal com emoções para AVAs chamado de Agente Companheiro. Tal abordagem se difere de um STI ao se apresentar como um colega virtual do estudante e não como um tutor.

Ainda sobre STIs, temos em (MAFFON et al., 2013) um STI para simular o comportamento de um ser humano e se adequar ao estilos de aprendizagem dos alunos.

Temos ainda em (FREDDO et al., 2009) o desenvolvimento de uma ferramenta de apoio ao processo colaborativo através da documentação do próprio processo. Essa ferramenta poderia então ser utilizada em qualquer CSCW e funcionaria como um assistente pessoal.

Mudando de abordagem, em (BARBOZA; MOTTA; SANTORO, 2008) é demonstrado a junção de técnicas de Sistemas de Hipermídia Adaptativa com CSCL, onde é descrita uma metodologia para construir um assistente para auxiliar o tutor na montagem do conteúdo a ser apresentado em um AVA. Portanto a ferramenta vai funcionar como um assistente para autoria de conteúdos auxiliando o tutor no seu trabalho.

Outra técnica relacionada é descrita em (LUGO; TACLA, 2012) onde são utilizadas Ontologias e a Técnica de Contexto para a criação de um Assistente Pessoal com capacidade para apresentar sugestões de ações a um membro do Ambiente Colaborativo. A ferramenta proposta identifica ações prováveis de atores humanos num processo de desenvolvimento

de software e funciona como um assistente pessoal para o coordenador do projeto.

Por fim em (WILGES et al., 2012) é demonstrada a utilização de técnicas de BDI (*Belief-Desire-Intention*) e Lógica *Fuzzy* para a criação de um modelo a ser utilizado por um STI para a avaliação e acompanhamento de alunos em um AVA.

### 3.1.2 Ambientes Inteligentes Colaborativos na Web

A preocupação em criar ambientes inteligentes para ajudar os seus usuários é uma preocupação demonstrada em vários trabalhos na comunidade acadêmica conforme nota-se em (AZEVEDO; TAVARES, 2001), onde é realizada uma discussão sobre vários ambientes inteligentes para aprendizagem colaborativa existentes na literatura além de fazer um contraponto ao ambiente proposto no próprio trabalho.

Já em (THIRY; BARCIA, 1999) é apresentado um Ambiente Colaborativo construído com o uso de um Sistema Multiagente (SMA) para utilização em Ensino a Distância, ou seja um CSCL. É sugerida a existência de uma relação entre sistemas de ensino com sistemas de apoio à tomada de decisão e os serviços inteligentes fornecidos pelo ambiente são: assistente pessoal inteligente (um Sistema Especialista para tomada de decisão) e agentes realizando tarefas no lugar do usuário.

Outro ambiente é a ferramenta EADOnline, descrita em (CONCEIÇÃO, 2000), utilizada para realização de cursos de Ensino a Distância (EaD). Nela são utilizados agentes tutores inteligentes para fornecer os serviços inteligentes: agente de agendamento de horários e agente tutor responsável por responder perguntas dos alunos.

### 3.1.3 Frameworks

Além de ambientes e ferramentas prontas para uso também existe na literatura um esforço para criação de *frameworks* com o objetivo de auxiliar no desenvolvimento de ambientes colaborativos na *Web*.

Um exemplo é o *framework* FamCorA, descrito em (PESSOA; NETTO; MENEZES, 2002). Ele pode ser utilizado para a criação de aplicações CSCL (um subgrupo das aplicações CSCW). Ele prevê serviços inteligentes para o CSCL tais como: busca inteligente de informações, sistema de recomendação, sistema de tradução de língua e assistentes inteligentes. No entanto, ele não considera a criação de outros serviços inteligentes além dos previstos e também não é possível aplicar diretamente este *framework* para ambientes em produção na *Web*.

Outro *framework* existente na literatura é o apresentado em (ZAMBIASI; RABELO, 2008) para a construção de Assistentes de Software Virtuais (ASV) em um cenário de computação ubíqua, com o objetivo de os assistentes auxiliarem as pessoas em determinadas tarefas. Nele é descrita uma modelagem genérica para a criação dos assistentes e

como identificar quais tarefas em ambientes colaborativos podem ser gerenciadas por um ASV. Os serviços inteligentes possíveis de serem criados seriam: substituição do usuário em determinadas tarefas, gerenciamento da agenda e atividades do usuário, procura de informações em repositórios, auxílio ao usuário na compreensão de tarefas a serem executadas, execução de algumas tarefas do usuário e filtro e gerenciamento das informações providas de redes ubíquas.

E temos ainda um *framework* descrito em (MARONI et al., 2011) para a construção de STIs utilizando simuladores virtuais inteligentes para serem utilizados por um AVA.

### 3.1.4 Modelos

Na literatura também encontramos modelos propostos para criação de ferramentas inteligentes para serem utilizadas dentro de um de CSCW ou para a construção do próprio ambiente.

Por exemplo, em (OLIVEIRA; TEDESCO, 2010) é proposto o modelo I-Collaboration para criação de CVAs (Companheiro Virtual de Aprendizado) em AVAs. Os CVAs se diferenciam dos STIs pois trabalham como um assistente pessoal (como um colega ou companheiro) para os alunos e não como um tutor. A proposta é a utilização dos CVAs em qualquer ambiente de Ensino a Distância.

Outro modelo existente na literatura é o MASSAYO, apresentado em (BITTENCOURT; COSTA, 2011), utilizado para a construção AVAs através de uma metodologia para a criação de Ambientes Educacionais Adaptativos e Semânticos. Aqui também é tida uma preocupação com a criação de serviços inteligentes tais como: armazenamento e recuperação de informações efetivas, agentes para aumentar o aprendizado e informações recuperadas pelos humanos e suporte, extensão e expansão das capacidades de comunicação dos humanos através de múltiplos formatos. No entanto a mesma crítica feita ao FamCorA se aplica aqui.

E temos também em (MELO et al., 2012) um modelo para organização de conteúdos didáticos em Sistemas Tutores Inteligentes conexionistas para uso em AVAs. Onde se descreve uma maneira de reestruturar e formalizar um conteúdo em diferentes níveis de apresentação do mesmo. O STI deve então escolher qual provavelmente será melhor para o aluno.

### 3.1.5 Arquiteturas

Dentre as propostas de arquiteturas encontradas na literatura para a criação de ambientes inteligentes destaca-se uma preferência por utilizar Sistemas Multiagentes.

Como exemplo, temos em (TOBAR et al., 2001) a proposta e descrição de um modelo de arquitetura a ser seguida na construção de um Ambiente Colaborativo para o

aprendizado de programação, um CSCL.

Outro exemplo é apresentado em (FILHO; QUARTO; FRANCA, 2010) onde se descreve uma arquitetura de Sistema Multiagentes para um Ambiente Colaborativo com ênfase em jogos com serviços inteligentes de assistente pessoal tanto para o tutor como para o aluno.

## 3.2 Resumo da Revisão Sistemática

Fizemos uma revisão sistemática para buscar saber quais serviços, considerados inteligentes pelos usuários, são oferecidos em ambientes CSCW e a partir dessa informação apoiar a construção de um *framework* desse tipo de serviço para a subclasse de Ambientes Colaborativos Flexíveis na *Web*. Optou-se por fazer o *framework* para uma subclasse e não para a classe mãe em razão de os Ambientes Colaborativos Flexíveis na *Web* já terem funcionalidades que facilitam a criação de serviços mais elaborados, como a capacidade de permitir alterações em tempo de execução das suas estruturas e interfaces.

Durante a execução da revisão sistemática, verificou-se que a ideia de adicionar serviços inteligentes aos Ambientes Colaborativos na *Web* não é nova, pois já existem trabalhos sobre essa temática, no entanto nos trabalhos encontrados a preocupação não é com a forma em si ou com o processo de criação de um serviço inteligente mas sim com a resolução de um problema muito específico do domínio onde o Ambiente Colaborativo é utilizado, o que evidencia a necessidade de desenvolver uma abordagem mais flexível e simples para a criação de serviços inteligentes em um Ambiente Colaborativo na *Web* e em especial nos ACFW por serem esses últimos o foco deste trabalho.

O processo de revisão sistemática está descrito no [Apêndice A](#) deste trabalho. Nesta seção nos concentramos apenas nos resultados encontrados que foram utilizados no desenvolvimento de nossa proposta.

Após a leitura e análise inicial dos artigos foram feitas algumas categorizações sobre os mesmos, com o objetivo de melhorar o estudo dos resultados. Assim os artigos foram classificados em grupos, descritos a seguir.

### 3.2.1 Classificando por Tipo de Ambiente

Nesta classificação os artigos foram separados dependendo se o contexto do artigo é um sistema de trabalho (CSCW) ou de aprendizagem (CSCL) colaborativo apoiado por computador. O resultado pode ser visto na Tabela 5

Um fato que se destacou foi a concentração do desenvolvimento de Serviços Inteligentes em CSCL. Tal fato chama a atenção para a possibilidade de analisar que a aplicação dos serviços desenvolvidos especificamente para CSCL sejam tornados genéricos e

Tabela 5 – Artigos classificados pelo tipo de ambiente

Tipo	Artigos	Qtd	Pct
CSCW	(CORDEIRO; BASTOS, 2006; DANTAS et al., 2009; DANTAS et al., 2008; FREDDO et al., 2009; GATIUS; GONZ, 2011; LIMA; DAVID; DANTAS, 2010; LUGO; TACLA, 2012; TONG; LU, 2002; YUAN; CHEN, 2008; ZAMBIASI; RABELO, 2008)	10	27,03%
CSCL	(ANAYA; BOTICARIO, 2010; AZEVEDO; TAVARES, 2001; BICA; VERDIN; VICARI, 2006; BITTENCOURT; COSTA, 2011; CHEN; ZHANG, 2002; CONCEIÇÃO, 2000; DONG; LI, 2005; CARO; JUMÉNEZ; PATERNINA, 2012; FIGUEIRA et al., 2013; FILHO; QUARTO; FRANCA, 2010; GOULART; GIRAFFA, 2001; GUTIÉRREZ-CARREÓN; DARADOUMIS; JORBA, 2008; IEPSEN; BERCHT; REATEGUI, 2013; KUNZEL et al., 2011; LI; MEI; WANG, 2012; FONTES et al., 2013; MELO et al., 2013; MAFFON et al., 2013; MARONI et al., 2011; MELO et al., 2012; OLIVEIRA; TEDESCO, 2010; PESSOA; NETTO; MENEZES, 2002; PRIMO; VICARI, 2011; THIRY; BARCIA, 1999; TOBAR et al., 2001; VICENTE; VASCONCELOS; RESTIVO, 2011; WILGES et al., 2012)	27	72,97%

Fonte: Autor.

aplicados aos CSCW, uma vez que segundo (BORGES; CAVALCANTI; CAMPOS, 1995), o primeiro é um subconjunto do último.

### 3.2.2 Classificando pelas Técnicas de IA Utilizadas

As técnicas de IA implementadas nos artigos da revisão sistemática estão elencadas na Tabela 6.

Tabela 6 – Classificação pelas técnicas de IA utilizadas

Técnica	Artigos	Qtd	Pct
Sistemas Multiagente	(AZEVEDO; TAVARES, 2001; BICA; VERDIN; VICARI, 2006; BITTENCOURT; COSTA, 2011; CHEN; ZHANG, 2002; CONCEIÇÃO, 2000; CORDEIRO; BASTOS, 2006; FREDDO et al., 2009; GOULART; GIRAFFA, 2001; KUNZEL et al., 2011; LUGO; TACLA, 2012; FONTES et al., 2013; MARONI et al., 2011; OLIVEIRA; TEDESCO, 2010; PESSOA; NETTO; MENEZES, 2002; THIRY; BARCIA, 1999; TONG; LU, 2002; VICENTE; VASCONCELOS; RESTIVO, 2011; WILGES et al., 2012; YUAN; CHEN, 2008; ZAMBIASI; RABELO, 2008)	20	54,40%
Ontologias	(BITTENCOURT; COSTA, 2011; CORDEIRO; BASTOS, 2006; DONG; LI, 2005; GATIUS; GONZ, 2011; GUTIÉRREZ-CARREÓN; DARADOUMIS; JORBA, 2008; LI; MEI; WANG, 2012; LIMA; DAVID; DANTAS, 2010; LUGO; TACLA, 2012; MAFFON et al., 2013; PRIMO; VICARI, 2011; TOBAR et al., 2001; YUAN; CHEN, 2008)	12	32,43%

continua

Tabela 6 – Classificação pelas técnicas de IA utilizadas

continuação

<b>Técnica</b>	<b>Artigos</b>	<b>Qtd</b>	<b>Pct</b>
Afetividade Computacional	(BICA; VERDIN; VICARI, 2006; CHEN; ZHANG, 2002; FILHO; QUARTO; FRANCA, 2010; IEPSSEN; BERCHT; REATEGUI, 2013; KUNZEL et al., 2011; FONTES et al., 2013; OLIVEIRA; TEDESCO, 2010)	7	18,92%
Clusterização	(CHEN; ZHANG, 2002; CARO; JUMÉNEZ; PATERNINA, 2012; FILHO; QUARTO; FRANCA, 2010)	3	8,11%
Contexto Computacional	(LIMA; DAVID; DANTAS, 2010; LUGO; TACLA, 2012; OLIVEIRA; TEDESCO, 2010)	3	8,11%
Lógica Fuzzy	(BICA; VERDIN; VICARI, 2006; MELO et al., 2013; WILGES et al., 2012)	3	8,11%
Mineração de Dados	(BITTENCOURT; COSTA, 2011; IEPSSEN; BERCHT; REATEGUI, 2013; MELO et al., 2013)	3	8,11%
Processamento de Linguagem Natural	(FIGUEIRA et al., 2013; GATIUS; GONZ, 2011; TOBAR et al., 2001)	3	8,11%
Raciocínio Baseado em Casos	(BITTENCOURT; COSTA, 2011; CARO; JUMÉNEZ; PATERNINA, 2012; THIRY; BARCIA, 1999)	3	8,11%
Rede Neural Artificial	(MAFFON et al., 2013; MELO et al., 2012; TOBAR et al., 2001)	3	8,11%
Aprendizagem por Reforço	(GOULART; GIRAFFA, 2001; YUAN; CHEN, 2008)	2	5,40%
Filtragem Colaborativa	(CARO; JUMÉNEZ; PATERNINA, 2012; PRIMO; VICARI, 2011)	2	5,40%
Lógica de Primeira Ordem	(DANTAS et al., 2009; DANTAS et al., 2008)	2	5,40%
N-Gramas	(FIGUEIRA et al., 2013; MELO et al., 2013)	2	5,40%
Semântica Latente	(PESSOA; NETTO; MENEZES, 2002; TOBAR et al., 2001)	2	5,40%
Serious Game	(MARONI et al., 2011; VICENTE; VASCONCELOS; RESTIVO, 2011)	2	5,40%
Algoritmo Genético	(VICENTE; VASCONCELOS; RESTIVO, 2011)	1	2,70%
Árvores de Decisão	(ANAYA; BOTICARIO, 2010)	1	2,70%
Metacognição	(CARO; JUMÉNEZ; PATERNINA, 2012)	1	2,70%
Raciocínio Baseado em Memória	(PESSOA; NETTO; MENEZES, 2002)	1	2,70%
Rede Bayesiana	(MARONI et al., 2011)	1	2,70%
Rede Neuro-Fuzzy	(CORDEIRO; BASTOS, 2006)	1	2,70%
Rede Semântica	(DANTAS et al., 2008)	1	2,70%
Regras de Produção	(THIRY; BARCIA, 1999)	1	2,70%
Sistema Hipermedia Adaptativo	(MAFFON et al., 2013)	1	2,70%
Web Semântica	(CORDEIRO; BASTOS, 2006)	1	2,70%

Fonte: Autor.

Aqui a concentração de pesquisas realizadas em Sistemas Multiagente (SMAs) pode ser explicada devido à diversidade e complexidade dos problemas tratados, pois com a utilização de agentes para modelar a resolução do problema é possível dividir o mesmo em vários problemas menores. Estes problemas podem então ser resolvidos individualmente sem afetar a resolução dos outros.

Além disso, é importante notar que em muitos casos os SMAs foram utilizados em conjunto com outras técnicas de IA para resolução do problema.

### 3.2.3 Classificando por Tipo de Serviço Inteligente

Durante a análise dos artigos da revisão sistemática foi percebida uma preferência por acrescentar Serviços Inteligentes ao ambiente na forma de uma melhoria de alguma atividade executada por algum usuário do ambiente. Em outras palavras, as melhorias são intervenções pontuais visando aumento de produtividade na interação entre o ambiente e os seus usuários.

Outro ponto notado foi a preferência por implementar essas melhorias nos ambientes como algum tipo de Software Assistente Virtual (SAV) executando uma função específica, podendo interagir com o usuário ou, em alguns casos, até mesmo chegando a substituir os usuários na execução de algumas tarefas.

O potencial de tarefas possíveis de serem executadas em um ambiente CSCW é grande e fortemente dependente dos objetivos do ambiente. Por exemplo, uma boa parte das tarefas executadas em um CSCW com objetivo de se escreverem livros didáticos são diferentes das tarefas a serem executadas em um CSCW com objetivo de gerenciar o andamento de projetos. No entanto, ainda assim existem algumas tarefas comuns em todos os ambientes CSCW, como por exemplo tarefas relacionadas ao gerenciamento de agendas e negociação de reuniões.

De modo a realizar uma análise apropriada, foi necessário adotar ou, na impossibilidade de tal, conceber uma forma de classificar os serviços em diferentes tipos genéricos de Serviços Inteligentes fornecidos.

#### 3.2.3.1 Classes de Serviços Inteligentes em Ambientes Colaborativos na Web

Para a criação de classes genéricas de serviços inteligentes foi utilizado como ponto de partida a lista de seis classes de tarefas realizadas por um SAV em um ambiente colaborativo conforme citado em (ZAMBIASI; RABELO, 2008): classe 1 - substituição do usuário em determinadas tarefas; classe 2 - gerenciamento da agenda e atividades do usuário; classe 3 - procura de informações em repositórios; classe 4 - auxílio ao usuário a compreender que tarefas precisam ser executadas e como elas seriam executadas; classe 5 -

execução de algumas tarefas do usuário; e classe 6 - filtro e gerenciamento das informações provindas de redes ubíquas.

Em uma tentativa preliminar de classificar os serviços dos artigos da revisão sistemática nessas seis classes, foram encontradas dificuldades no enquadramento de algumas tarefas. Por exemplo, em alguns casos nem sempre as informações provêm de redes ubíquas e, para serviços responsáveis por personalizar o ambiente para o seu usuário, não era possível classificá-los nas seis classes existentes. Tais problemas foram superados com a alteração da descrição da classe 6, que passou a ser: classe 6 – filtro e gerenciamento das informações provindas de fontes externas. Criamos também uma nova classe: classe 7 – personalização automática do ambiente ao usuário.

Assim, na Tabela 7 são mostradas as 7 classes de Serviço Inteligentes bem como a porcentagem de artigos da revisão sistemática onde é percebido um exemplo do Serviço Inteligente daquela categoria. O somatório das porcentagens é maior que 100% pois muitos trabalhos apresentaram mais de um tipo de serviço inteligente.

Tabela 7 – Distribuição dos Tipos de Serviços Inteligentes na Revisão Sistemática

Classe	%
Auxílio ao usuário a compreender que tarefas precisam ser executadas e como	83,78%
Procura de informações em repositórios	27,03%
Personalização automática do ambiente ao usuário	27,03%
Substituição do usuário em determinadas tarefas	18,91%
Execução de algumas tarefas do usuário	18,91%
Filtro e gerenciamento das informações provindas de fontes externas	8,11%
Gerenciamento da agenda e atividades do usuário	5,40%

Fonte: Autor.

Percebe-se uma forte concentração nas pesquisas onde o usuário é auxiliado na compreensão da escolha de quais tarefas devem ser executadas e como elas devem ser executadas. Ou seja, a concentração é na criação de Serviços Inteligentes para auxiliar no trabalho realizado no ambiente em questão seja ele um CSCW ou um CSCL.

Essa conclusão parece ser natural pois podemos ver como um objetivo tanto dos CSCW como dos CSCL fazer com que os seus usuários executem tarefas dentro do ambiente para se atingir algum objetivo previamente estipulado.

### 3.2.4 Classificando por Formas de Auxílio ao Usuário

Como mostrado anteriormente, os artigos da revisão sistemática tiveram uma grande concentração na área de serviços para auxiliar os usuários na identificação de tarefas a serem executadas e em como executar essas tarefas.

Com o objetivo de conhecer melhor essa classe de serviços inteligentes, foi verificado quais eram os tipos de sistemas utilizados nos artigos para se implementar tais serviços.

Revisando os artigos foi possível notar a utilização de três tipos de sistemas para implementar os serviços e um quarto tipo que é uma especialização de um dos tipos já existentes:

- **Sistema de Recomendação.** Quando o serviço apresenta ao usuário recomendações de ações a serem feitas ou opções possíveis para alguma atividade, porém ele não informa uma ordem de preferência ou ranqueamento. Ou seja, o usuário não recebe uma explicação sobre por que escolher uma opção em detrimento de outra.
- **Sistema de Apoio a Decisão.** Quando o serviço apresenta ao usuário ações a serem feitas além de apresentar uma ordem de preferência ou ranqueamento. Aqui o usuário além de ter uma explicação ou pelo menos uma indicação de por que escolher uma opção em vez de outra ele continua tendo a liberdade de não escolher a melhor sugestão.
- **Sistema Especialista.** Quando o serviço oferece uma função ou opção no ambiente a qual normalmente seria preciso a intervenção de um especialista para poder ocorrer.
  - **Sistema Tutor Inteligente (STI).** Um Sistema Especialista com objetivo de ensinar ao usuário um assunto ou matéria específica. Deve apresentar os módulos: estudante (conhecimento sobre o estudante), domínio (conhecimento sobre o assunto), pedagógico (metodologias de ensino), especialista (acompanha o desenvolvimento dos alunos e ajusta o processo de aprendizagem) e comunicação.

Assim na Tabela 8 temos a distribuição das formas de auxílio ao usuário através de serviços inteligentes que foram encontrados durante a revisão sistemática.

Tabela 8 – Distribuição das Formas de Auxílio ao Usuário com Serviços Inteligentes

Tipo	%
Sistema de Recomendação	13,89%
Sistema de Apoio à Decisão	13,89%
Sistema Especialista - Sistema Tutor Inteligente	38,89%
Sistema Especialista - Outros Tipos	33,33%

Fonte: Autor.

Percebe-se uma tendência para implementação desse tipo de serviço (com objetivo de auxiliar os usuários na execução de tarefas no ambiente CSCW) através da captura do conhecimento de um especialista em um assunto importante para as atividades a serem exercidas no ambiente específico em questão.

As informações sobre classes de serviços inteligentes e as formas mais utilizadas para auxílio ao usuário serviram de inspiração para a tarefa de planejar um *framework* que fornece serviços inteligentes em Ambientes Colaborativos Flexíveis. A proposta deste *framework* é apresentada no próximo capítulo.

### 3.3 Validação da Classificação de Serviço Inteligente

Nesta seção vamos explicar o processo de validação que realizamos para verificar se as classes de serviços inteligentes utilizadas (vide [subseção 3.2.3.1](#)) são apropriadas ao nosso contexto. Para tanto, conforme discutimos na Seção 2.1 estamos seguindo a premissa de que se um humano realiza um serviço X em um ambiente computacional para um outro humano (seu contratante) então esse serviço X seria considerado “inteligente” caso fosse feito por um sistema computacional.

Precisamos, portanto, de um contexto onde um humano forneça serviços para outros humanos, ambos dentro de um ambiente computacional. Tal contexto é justamente o caso do trabalho de um “Assistente Humano Virtual”.

Assim, vamos descrever o serviço de um Assistente Humano Virtual e como isto nos ajudou a verificar a nossa classificação de serviços inteligentes.

#### 3.3.1 Assistente Humano Virtual

A expansão do volume de serviços disponíveis na Internet, uma expansão das ferramentas para comunicação na Internet (principalmente VOIP, videoconferência e webconferência) juntamente com a terceirização dos serviços não-essenciais por parte das empresas fez com que aumentasse o volume de pessoas trabalhando como Assistente Virtual (em inglês *Virtual Assistant*). Para não confundir com sistemas de software vamos chamá-lo de Assistente Humano Virtual (AHV).

Um AHV é uma pessoa ou uma empresa (consistindo de vários AHV) cujo trabalho é a realização de serviços profissionais de assistência a distância (principalmente via Internet). Tais serviços podem ser de ordem administrativa, técnica ou criativa. Temos portanto, que os trabalhos possíveis de serem feitos por AHVs tem uma grande variedade seja em produtos ou em serviços prestados para quem os contrata.

Quem normalmente utiliza esse tipo de serviço são pessoas com muitas atribuições ou responsabilidades porém sem o tempo necessário para trabalhá-las em todos os detalhes (por exemplo: CEOs, Diretores, Gerentes, etc) ou negócios que gostariam de ter acesso a um profissional específico mas que não podem ou não têm necessidade desse profissional em tempo integral.

Toda a interação entre cliente e fornecedor se dá a distância e uma boa parte dessa interação acontece por meios digitais, não sendo difícil imaginar casos onde o cliente e o fornecedor jamais tenham se encontrado pessoalmente.

Devemos observar que o AHV não faz o trabalho do seu cliente, mas sim as tarefas relacionadas ao trabalho do cliente de tal forma que o cliente possa dedicar mais tempo ao seu trabalho. Em tal cenário é razoável considerar que de forma análoga, as pessoas

poderiam terceirizar para os sistemas computacionais as tarefas auxiliares na execução do trabalho que precisa ser feito.

### 3.3.2 Avaliação Realizada

Como forma exploratória de investigar se essa abordagem poderia ser utilizada, foi feito o experimento de a partir da classificação de serviços inteligentes adotada neste trabalho classificarmos tarefas normalmente associadas com terceirização de serviços para um AHV.

Portanto, a etapa de verificação realizada pode ser resumida na seguinte pergunta: as tarefas normalmente delegadas para um AHV caso fossem feitas por um sistema computacional poderiam ser classificadas em uma das classes de serviços inteligentes? Caso não seja possível fazer essa classificação então teríamos uma inadequação nas nossas categorias. Seja porque as classes propostas são insuficientes ou porque algumas tarefas não podem ser feitas via um sistema computacional ou por outro motivo.

As principais tarefas terceirizadas para um AHV geralmente são do campo da administração, no entanto existem AHV especialistas em muitos campos tais como: marketing, desenvolvimento de sítios *web*, turismo, etc. Desse modo, uma lista genérica de tarefas sugeridas para serem terceirizadas para um AHV deve englobar tanto tarefas genéricas (fazem sentido para qualquer contexto de trabalho) como específicas (fazem sentido apenas em alguns contextos de trabalho). Esta lista de tarefas, bem como o processo para a sua confecção e a classificação de todas as tarefas da lista nas classes de serviço inteligente estão descritos no [Apêndice B](#) desta proposta.

### 3.3.3 Teste e Descrição das Classes

Aqui vamos descrever cada uma das classes de serviço, exemplificando-as com algumas das tarefas obtidas da lista de tarefas do [Apêndice B](#).

#### 3.3.3.1 Substituição do Usuário

Aqui entram os serviços que substituem os usuários na execução de alguma tarefa. Os serviços devem não apenas realizar certas tarefas, mas também personificar o usuário, assim caso durante a execução do serviço seja necessário interagir com outros usuários, estes poderiam não saber se no momento em que a tarefa é realizada é o próprio usuário ou o sistema quem está interagindo com ele.

Fazendo uma comparação com o comportamento de um AHV, nesta classe teríamos as tarefas que são repassadas para o AHV e ele as executa com as credenciais da pessoa que terceirizou o serviço. Como exemplo de tarefas de um AHV nessa classe temos:

- Moderação de fóruns na internet;
- Reserva de hotéis e marcação de voos;
- Realização de pagamentos e transferências bancárias.

### 3.3.3.2 Gerenciamento da Agenda

Aqui ficam os serviços responsáveis pela criação e manutenção da agenda do usuário e/ou do grupo incluindo tarefas tais como: agendamento de tarefas, reuniões, lembretes para o usuário e gerenciamento de prazos. O próprio ambiente pode realizar agendamentos automáticos ou sugerir reuniões e horários aos usuários. Ele pode por exemplo facilitar a negociação de horários entre os envolvidos.

Comparando com o comportamento de um AHV aqui teríamos as tarefas terceirizadas para o AHV que envolvam a agenda do seu cliente, bem como lembretes de prazos e eventos que o AHV julgue interessante ou necessário para o seu cliente. Como exemplo de tarefas de um AHV que podemos classificar nesta classe temos:

- Criação e atualização de eventos importantes no calendário;
- Agendar a execução de postagens em mídias sociais;
- Criar lembretes para prazos e eventos importantes.

### 3.3.3.3 Busca de Informações em Repositórios

Aqui ficam os serviços que precisam interagir com repositórios de informações externos ao ambiente para obter informações necessárias ao seu funcionamento ou devido a solicitação do usuário. É desejável que os serviços tenham uma certa iniciativa de quando e onde devam buscar informações, não sendo sempre inicializados devido a demanda explícita do usuário.

Na comparação com as tarefas terceirizadas para um AHV aqui ficariam as tarefas que precisam que o AHV realize pesquisa em uma ou várias fontes de informação para que possam resolver o problema. Como exemplo de tarefas de um AHV que podemos classificar nesta classe temos:

- Realizar uma pesquisa de palavras-chaves sobre um assunto determinado;
- Encontrar locais apropriados e com alto tráfego para publicação de propaganda;
- Procurar e contactar especialistas da indústria para participarem em um *podcast*.

### 3.3.3.4 Auxílio ao Usuário

Aqui ficam os serviços responsáveis por auxiliarem os usuários na execução das suas tarefas. Eles auxiliam o usuário a compreender que tarefas precisam ser executadas e como elas devem ser executadas. Esse auxílio pode ocorrer: fornecendo ferramentas para facilitar o trabalho do usuário; elucidando dúvidas do usuário sobre a tarefa; ou sugerindo atividades a serem feitas.

Comparando com tarefas feitas por um AHV teríamos nesta classe as tarefas onde o AHV realiza a tarefa em conjunto com o cliente ou realiza tarefas acessórias que auxiliam o cliente na execução da tarefa principal. Como exemplo de tarefas de um AHV que podemos classificar nesta classe temos:

- Juntar documentos necessários e preencher a declaração de imposto de renda;
- Criar documentos a partir de rascunhos feitos à mão ou enviados por fax ou ditados em uma gravação;
- Revisão e correção de textos escritos

### 3.3.3.5 Execução de Tarefas dos Usuários

Aqui ficam os serviços que assumem a responsabilidade pela execução de tarefas que antes eram executadas pelo usuário. Assim como na classe “Substituição do Usuário” aqui também o serviço é feito com as credenciais do usuário, a diferença é que na classe “Substituição do Usuário” há uma delegação temporária da tarefa do usuário para o sistema, enquanto que aqui a delegação é por tempo indeterminado. Serviços desta classe são geralmente construídos através de automatização de tarefas do usuário.

Como exemplo de tarefas de um AHV nesta classe temos:

- Envio de cartões de visita para potenciais clientes;
- Realização de backups regulares para prevenir perda de dados;
- Edição de arquivos de áudio para remoção de ruído.

### 3.3.3.6 Filtro e Gerenciamento das Informações Provindas de Fontes Externas

Aqui ficam os serviços que precisam receber ou interceptar a comunicação de fontes externas endereçadas ao usuário para poderem realizar as suas tarefas. De posse da comunicação eles podem filtrar ou encaminhar as mensagens conforme for necessário para a execução de suas tarefas.

Como exemplo de tarefas de um AHV nesta classe temos:

- Verificar e-mail e administrar o SPAM;
- Filtrar comentários no seu blog;
- Atender ligações e registrar recados.

### 3.3.3.7 Personalização Automática do Ambiente ao Usuário

Aqui ficam os serviços que visam adaptar automaticamente o ambiente às necessidades do usuário. Os serviços devem detectar as características do usuário e aproveitar cenários de sucesso em outros perfis semelhantes.

Na relação de exemplo de tarefas de um AHV não foi possível encontrar exemplos para essa classe. No entanto, acreditamos que isso aconteceu devido a esse tipo de serviço ser básico e por isso passar despercebido durante a elaboração da lista de tarefas. É natural que um AHV humano ao entregar o produto de algum serviço executado faça uma adaptação de acordo com as necessidades do usuário. Por exemplo, caso o usuário tenha uma deficiência visual o AHV vai entregar relatórios com uma fonte maior para facilitar a leitura ou sem usar algumas cores (no caso de clientes daltônicos). Outro exemplo é o caso em que o cliente tem preferência ou necessidade de usar algum padrão específico (de arquivo, de apresentação, etc) para atender uma conformidade com padrões de governança, leis locais ou requisitos de mercado.

## 3.4 Conclusão do Capítulo

Neste capítulo, foi verificado que adicionar serviços inteligentes aos ambientes colaborativos é uma ideia bem difundida pois já existem vários relatos de serviços inteligentes, ambientes inteligentes colaborativos na web, *frameworks*, modelos e arquiteturas que tratam desta questão. No entanto, tais estudos além de não serem específicos para Ambientes Colaborativos Flexíveis na *Web* também tem um foco excessivo no ambiente computacional onde o serviço é oferecido, dificultando assim a reutilização de soluções já desenvolvidas para um ambiente para outro ambiente. Além disso, os *frameworks*, modelos e arquiteturas apresentados nesta seção não consideram o desenvolvimento de serviços inteligentes para ambientes que já estão em produção, e portanto, já sendo utilizado pelos seus usuários.

Também foi apresentado um resumo da revisão sistemática realizada, bem como os resultados encontrados, especialmente: o tipo de ambiente colaborativo na *Web* onde a maioria dos serviços inteligentes em pesquisa se encontra, quais as técnicas de Inteligência Artificial que foram mais utilizadas nesses casos, quais são as classes de serviços inteligentes e quais as principais formas de auxílio ao usuário que os serviços inteligentes encontrados forneceram. Sendo que o principal resultado da revisão sistemática foi a criação de uma

classificação para os tipos de serviços inteligentes (para a camada de serviços inteligentes, objeto desta tese):

- Auxílio ao usuário a compreender que tarefas precisam ser executadas e como;
- Procura de informações em repositórios;
- Personalização automática do ambiente ao usuário;
- Substituição do usuário em determinadas tarefas;
- Execução de algumas tarefas do usuário;
- Filtro e gerenciamento das informações provindas de fontes externas;
- Gerenciamento da agenda e atividades do usuário.

Para se chegar a esta classificação partiu-se de uma proposta já existente e foram feitos aprimoramentos nela de forma a para atender as necessidades do trabalho realizado neste doutorado. Para validar essa classificação foi feito e descrito um processo de avaliação onde foi concluído que essas classes de serviço inteligente são não somente válidas como também úteis para o contexto do trabalho dessa tese.

Parte III

Resultados

## 4 Um Modelo de Referência para Serviços Inteligentes na Web

Conforme apresentado no primeiro capítulo, neste trabalho foi desenvolvido um *framework* para a criação de serviços inteligentes em Ambientes Colaborativos Flexíveis na Web. Como um primeiro passo para a construção do *framework* foi criado um modelo de referência para o desenvolvimento de uma solução do problema abordado. Este modelo foi então utilizado para guiar a construção de uma arquitetura de referência, sendo esta arquitetura o *framework* objeto deste trabalho. Por fim, foi feita uma implementação da mesma para servir como prova de conceito do *framework*.

No restante deste capítulo vamos descrever a analogia utilizada como inspiração para a criação do modelo de referência. Ela vai ser utilizada como uma introdução para os conceitos, ideias e relacionamentos apresentados no decorrer deste capítulo. Após a analogia vamos descrever o modelo de referência e os seus conceitos e relacionamentos.

### 4.1 A Analogia do Assistente Humano Virtual

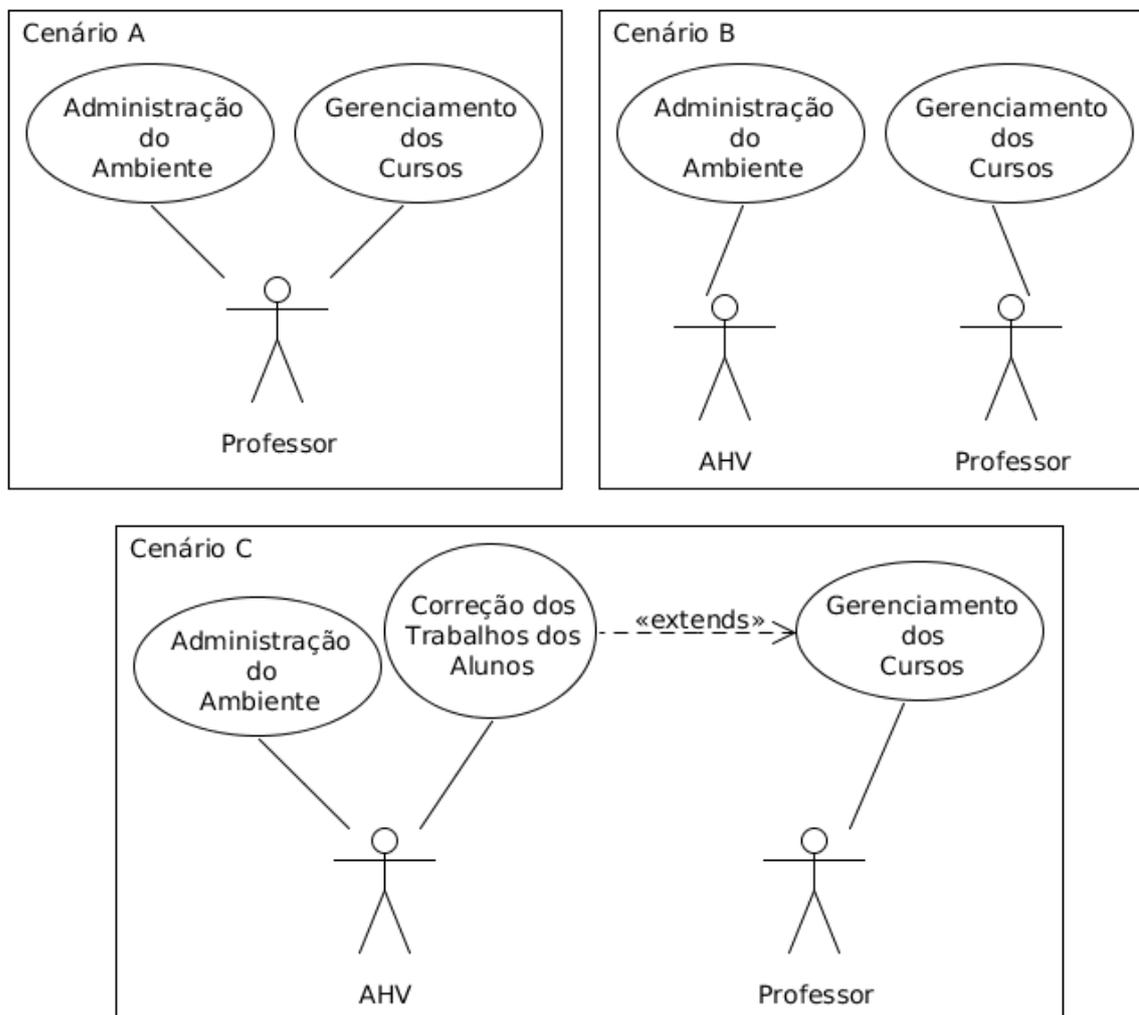
Para poder explicar como o AHV serviu de inspiração para a criação de um modelo de referência para uma camada de serviços inteligentes vamos descrever um cenário exemplo utilizando um AHV e depois discutir sobre a analogia de como os sistemas computacionais podem ser vistos como assistentes e não somente como executores de solicitações e em como essa mudança de visão ajudou a planejar uma camada de serviços inteligentes.

#### 4.1.1 Um Cenário de Exemplo

Inicialmente um professor começa um empreendimento de cursos online. No começo ele sozinho executa todas as tarefas necessárias ao andamento do curso (criação de conteúdo, correção de trabalhos, tirar dúvidas dos alunos, etc..) juntamente com as tarefas necessárias a administração do Ambiente Virtual de Aprendizagem (atualizações, backups, criação de contas, etc..) conforme cenário A da Figura 8.

Com o passar do tempo e o crescimento do seu empreendimento (mais alunos e mais turmas) ele resolve contratar ajuda e assim terceiriza a administração do ambiente virtual para um AHV, ficando então ele livre para executar as tarefas necessárias ao andamento do curso (cenário B da Figura 8). Idealmente o AHV contratado já deveria saber realizar as tarefas que serão terceirizadas, mas o professor pode ensinar ele como desenvolve-las. O AHV então passa a ser responsável pela execução das tarefas de administração do

Figura 8 – Cenário de Exemplo.



Fonte: Autor.

AVA, consultando o professor sempre que algum imprevisto ocorra. Ele também envia ao professor alguns relatórios sobre o ambiente sempre que solicitado. Passado mais algum tempo, o sucesso do ambiente virtual é tão grande que agora o professor não tem tido tempo para executar todas as tarefas necessárias para corrigir os trabalhos enviados pelos alunos, visto que aumentou muito a quantidade de alunos. O professor então expande o contrato de terceirização do AHV para incluir as tarefas de correção dos trabalhos dos alunos além das tarefas que o AHV já executava (cenário C da Figura 8).

#### 4.1.2 Discussão sobre AHV e Camada de Serviços Inteligentes

Como o relacionamento entre um AHV e o seu cliente acontece inteiramente a distância, principalmente usando a Internet, não é difícil imaginar casos onde o cliente e o AHV jamais tenham se encontrado pessoalmente. Some-se a isso o fato de o AHV não

fazer o trabalho do seu cliente mas sim as tarefas relacionadas ao trabalho do cliente de tal forma que o cliente possa dedicar mais tempo ao seu trabalho, podemos concluir que se o AHV fosse substituído por um sistema computacional inteligente (por exemplo, um que fosse capaz de passar no Teste de Turing) que nem o cliente nem outras pessoas que dependessem do trabalho do cliente perceberiam a mudança.

Então imaginemos que no cenário exemplo atuasse um sistema computacional suficientemente inteligente se passando por um AHV. Neste caso, podemos concluir que não teríamos desvantagens para o cliente. No entanto, teríamos vantagens adicionais tais como: serviço disponível a qualquer hora, capacidade maior de solicitar tarefas em paralelo e atendimento sempre cordial e amigável.

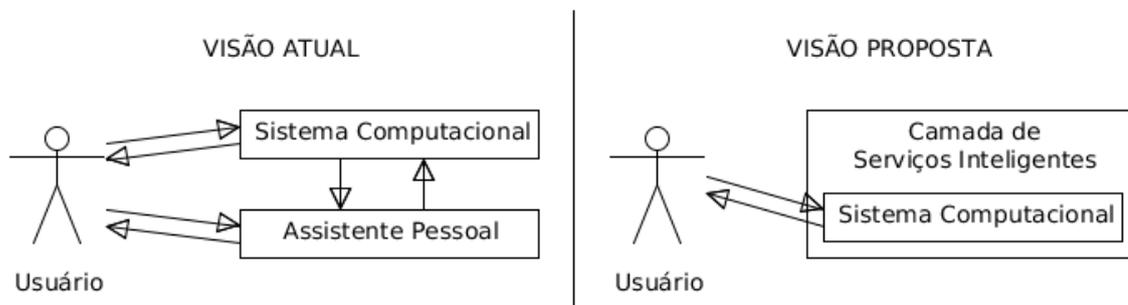
Infelizmente ainda estamos longe de tal cenário se tornar uma realidade, no entanto podemos utilizar esse cenário como uma meta a ser atingida ou como inspiração para o comportamento esperado de um sistema computacional, objetivando a criação de novos serviços ou melhoramentos dos serviços atualmente existentes em um sistema computacional. A idéia é fazer com que o sistema computacional (que no nosso caso é um Ambiente Colaborativo Flexível na Web) se comporte de forma mais parecido com um AHV.

Muitas das qualidades que apreciamos em um bom AHV também seriam apreciadas em um sistema computacional. No cenário de exemplo, poderíamos dizer que o AHV faz um bom trabalho se ele percebe que o professor pede um relatório específico a cada dois meses e com isso já prepara o relatório quando a data estivesse perto das próximas solicitações, ou se ele ao corrigir as provas de uma turma informa ao professor quais os tópicos que os alunos estão com dificuldade. Tais ações se efetuados por um humano (o AHV) seriam consideradas inteligentes, por isso consideramos que elas também seriam inteligentes dentro de um sistema computacional.

Como fazer então para que um sistema computacional se comporte de forma parecida com um AHV? Nós não visualizamos a criação de um assistente pessoal dentro da aplicação fazendo interface com o usuário e auxiliando-o diretamente. Outros trabalhos já propuseram essa linha de pesquisa (HUHNS; SINGH, 1998; POZNA; FOLDESI; KOVACS, 2013). O que estamos propondo é que o próprio Ambiente Colaborativo Flexível na Web se comporte como um AHV. Nossa visão é exemplificada na Figura 9, onde é apresentada a diferença de visões, de um lado tem-se um usuário interagindo separadamente um assistente pessoal e com um sistema computacional, do outro lado temos o usuário interagindo apenas com o sistema computacional enquanto a camada de serviços inteligentes realizada o seu trabalho de forma transparente para o usuário.

Fazendo um paralelo com o cenário exemplo, um cliente seria um usuário do ambiente, o AHV seria então o próprio ambiente, a criação de serviços inteligentes seria uma forma de fazer com que as tarefas do usuário fossem repassadas para o ambiente.

Figura 9 – Nossa proposta de visão para serviços inteligentes.



Fonte: Autor.

Assim como no cenário exemplo, o ambiente vai fazer com que a produtividade dos seus usuário aumente. No exemplo, o professor está atendendo em cada fase uma quantidade maior de alunos portanto do ponto de vista do professor houve um aumento da sua produtividade.

Ou seja, o ambiente deveria, sempre que possível, realizar as tarefas correlatas a execução da tarefa sendo feita pelo usuário. Quanto mais tarefas ele fizer pelo usuário mais sofisticado é o ambiente e maior é a produtividade do usuário. Por exemplo, se o trabalho do usuário é escrever um livro, o sistema poderia fazer a correção ortográfica e a diagramação do texto de tal forma que o usuário se concentra apenas na escrita do texto o que significaria que o usuário poderia escrever mais e melhor neste tipo de ambiente do que em outro que não fornece esses serviços extras.

Para se obter esse tipo facilidade e organizar a criação de novos serviços propomos que se deve utilizar uma Camada de Serviços Inteligentes. O objetivo de tal camada é servir de meio para que o sistema computacional comece a se comportar como como um AHV fazendo com que o usuário consiga realizar o seu trabalho com um esforço cognitivo menor do que antes.

## 4.2 Definição de Modelo de Referência

Um modelo de referência, segundo (MACKENZIE et al., 2006), é um *framework* abstrato que descreve de forma satisfatória as principais entidades e os principais relacionamentos entre as entidades de um determinado ambiente. Ele deve permitir a criação de padrões e especificações consistentes que podem ser utilizadas para se fazer referências específicas durante o desenvolvimento da solução ou para a criação de arquiteturas mais concretas (tais como as arquiteturas de referência). Assim, um modelo de referência deve

atender aos seguintes requisitos:

- Ser abstrato. Ele deve possuir um conjunto mínimo de conceitos unificadores que descrevem os tipos de entidades existentes dentro do domínio do problema. Ele não deve descrever entidades específicas;
- Possuir um conjunto mínimo de axiomas do domínio do problema;
- Descrever um conjunto mínimo de relacionamentos existentes entre os conceitos do domínio do problema. Ele deve descrever como os conceitos se conectam e interagem uns com os outros e se eles tem ou não propriedades em comum;
- Ser independente de padrões, tecnologias e implementações específicas ou de outros detalhes mais concretos.

Como exemplos de Modelos de Referência podemos citar:

- Arquitetura de von Neumann;
- Modelo OSI de redes de computadores;
- *Agent systems reference model*<sup>1</sup>;
- O Modelo de Referência para uma Arquitetura Orientada a Serviços ([MACKENZIE et al., 2006](#)).

Assim, podemos concluir que o objetivo de um modelo de referência é ser utilizado como um local comum onde várias implementações podem ser ancoradas. Portanto, um modelo de referência pode ser utilizado para comparar entre si várias implementações, estudar o domínio do problema ou explicar o funcionamento de uma implementação específica.

### 4.3 Convenções Utilizadas na Descrição

As palavras chaves NECESSITA (MUST), NÃO PODE (MUST NOT), DEVE (SHOULD), NÃO DEVE (SHOULD NOT) e PODE (MAY) devem ser interpretadas conforme descrito na RFC2119<sup>2</sup>.

Referências a outros conceitos dentro da descrição de um conceito serão escritos em **negrito**.

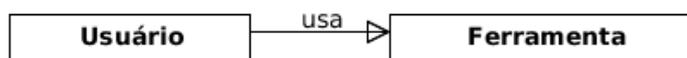
<sup>1</sup> <[http://www.fipa.org/docs/ACIN-reference\\_model-v1a.pdf](http://www.fipa.org/docs/ACIN-reference_model-v1a.pdf)>

<sup>2</sup> <<https://www.ietf.org/rfc/rfc2119.txt>>

Para a representação dos relacionamentos existentes entre as ideias ou conceitos do modelo de referência aqui apresentado foi escolhido utilizar-se de mapas conceituais por serem uma ferramenta adequada a essa finalidade (NOVAK; CAÑAS, 2008).

Nos mapas conceituais utilizados o sentido das setas indica a direção em que os conceitos devem ser lidos, por exemplo na Figura 10 devemos ler “Usuário usa Ferramenta”.

Figura 10 – Mapa conceitual de exemplo.



Fonte: Autor.

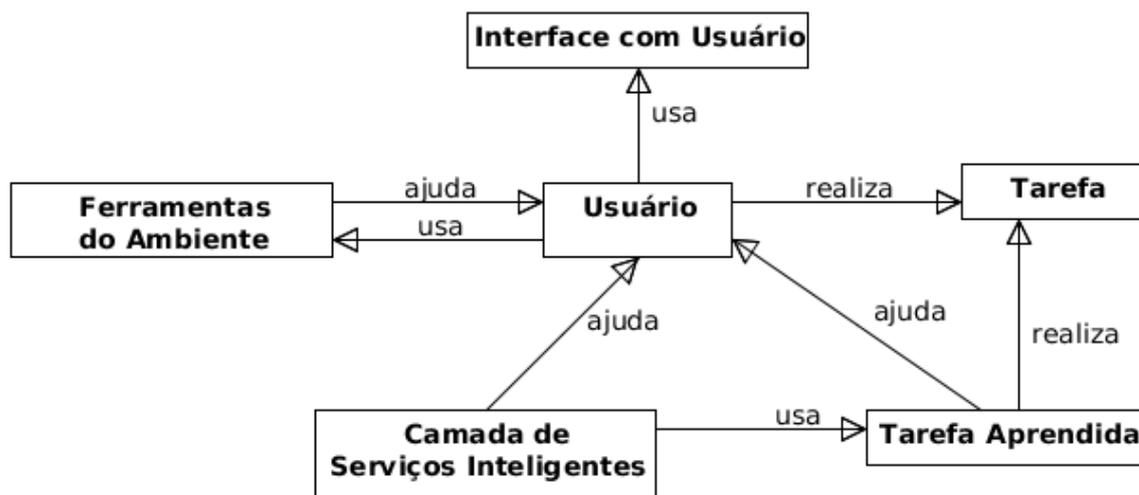
Os possíveis verbos utilizados como rótulos em cima das setas e as suas definições são as seguintes:

- **Usa.** No sentido de utilizar, realizar uma ação com ou trabalhar com.
- **Ajuda.** Usado no sentido de auxiliar no trabalho, auxiliar durante a execução, apoiar o trabalho. Indica que mesmo que a relação não existisse a ação ainda poderia ser feita, porém com um maior grau de dificuldade e esforço.
- **Ler.** Utilizado no sentido de ter acesso para leitura dos dados. Não necessariamente indica uma capacidade de alteração ou gravação dos dados.
- **Realiza.** No sentido de fazer o trabalho, realizar a ação, ser responsável por fazer.
- **Tem.** Usado no sentido de ter uma subparte, de ser dividido em partes, de um está contido dentro de outro.
- **Armazena.** Utilizado no sentido de guardar dados, de registrar uma informação, de manter registros.
- **Cria.** No sentido de construir, fabricar ou desenvolver.

#### 4.4 O Modelo de Referência

Para a apresentação do modelo de referência optamos pela descrição dos conceitos em uma ordem escolhida que, acredita-se, facilita o entendimento dos mesmos.

Figura 11 – Relações do conceito Usuário.



Fonte: Autor.

#### 4.4.1 Usuário

A Figura 11 apresenta o conceito usuário e os seus relacionamentos com outros conceitos. Definimos um usuário, no nosso contexto, como sendo um humano que utiliza o ambiente.

A **camada de serviços inteligentes** visa justamente auxiliar esses humanos na realização das suas atividades dentro do ambiente, não importando se é uma atividade de trabalho, aprendizagem ou algum outro tipo.

Os usuários acessam o sistema computacional para poder utilizar as **ferramentas do ambiente**. Essas ferramentas, por sua vez, ajudam os usuários na resolução dos problemas. Um mesmo usuário pode utilizar mais de uma ferramenta para concluir o seu trabalho dentro do **ambiente**.

Apesar de outras entidades poderem utilizar a **interface com usuário** a principal entidade a fazer uso dela é o **usuário**. Além disso, ele é também o principal destinatário da maioria das ações que o **ambiente** executa. Assim, devido a importância do usuário para os serviços inteligentes, pois é ele quem os consome, qualquer solução adotada DEVE representá-lo formalmente dentro da sua estrutura. Pois assim será possível levar em consideração as características intrínsecas dos usuários durante a execução dos serviços.

Inicialmente, antes da incorporação de uma **camada de serviços inteligentes**, são os usuários que realizam as **tarefas** dentro do **ambiente** com a ajuda das **ferramentas do ambiente**. Conforme novas **tarefas aprendidas** são criadas dentro da **camada de serviços inteligentes** o **ambiente** vai ficando mais refinado pois reduz a carga cognitiva do usuário. Essa redução acontece porque vai diminuindo o número de **tarefas** realizadas

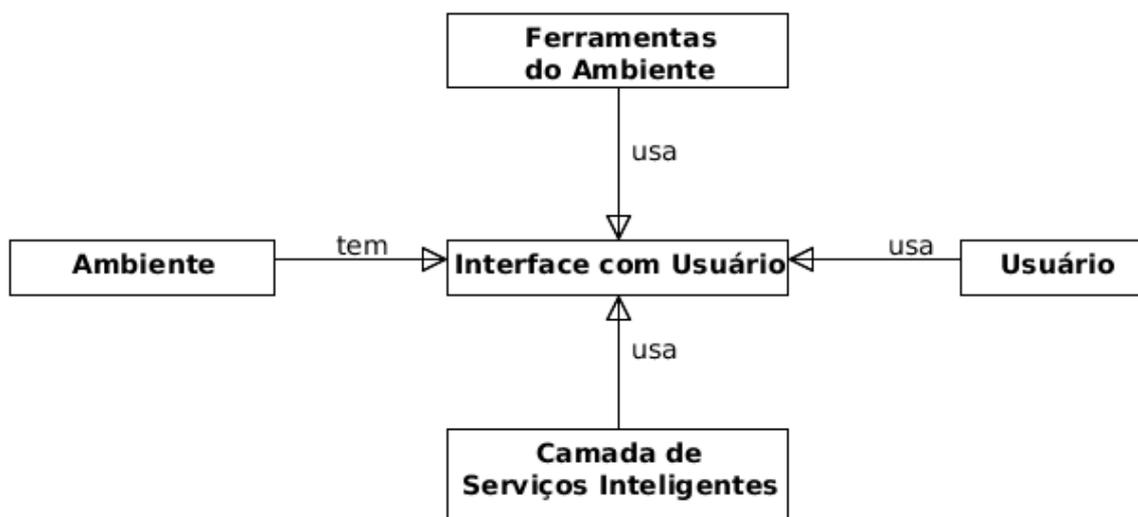
pelo usuário à medida que o **ambiente** incorpora as antigas **tarefas** como funcionalidades do sistema (criando novas **tarefas aprendidas**).

A principal percepção que um usuário deveria ter da aplicação de uma **camada de serviços inteligentes** no **ambiente** é no sentido de que ela aumenta a sua produtividade ao diminuir a carga cognitiva dele. Como para realizar as **tarefas** dentro do **ambiente** os usuários utilizam as **ferramentas do ambiente** (seja para realizar interações com outros participantes seja para executar as ações necessárias à realização das suas **tarefas**) então podemos afirmar que com mais **tarefas** sendo realizadas pelo próprio **ambiente** menos necessidade vai haver do usuário interagir com as ferramentas (seja reduzindo o tempo total de uso de uma ferramenta ou a quantidade de ferramentas necessárias a execução de uma **tarefa** pelo usuário. Portanto:

**Axioma 1.** *Para o usuário o ambiente mais inteligente é aquele onde ele menos precisa interagir para concluir uma tarefa.* —

#### 4.4.2 Interface com Usuário

Figura 12 – Relações do conceito Interface com Usuário.



Fonte: Autor.

O conceito de interface com usuário e os seus relacionamentos no modelo de referência são apresentados na Figura 12. No modelo a interface com usuário é o local e o meio por onde ocorrem as interações entre os **usuários** e o sistema computacional. Ela é o local onde os **usuários** acessam os serviços oferecidos pelo **ambiente** para a realização das **tarefas**. As **ferramentas do ambiente** usam a **interface com usuário** tanto para receberem os dados como para receberem as solicitações dos **usuários**. Como a realização das **tarefas** pode ser totalmente dentro do **ambiente** ou pode ser apenas uma parte da

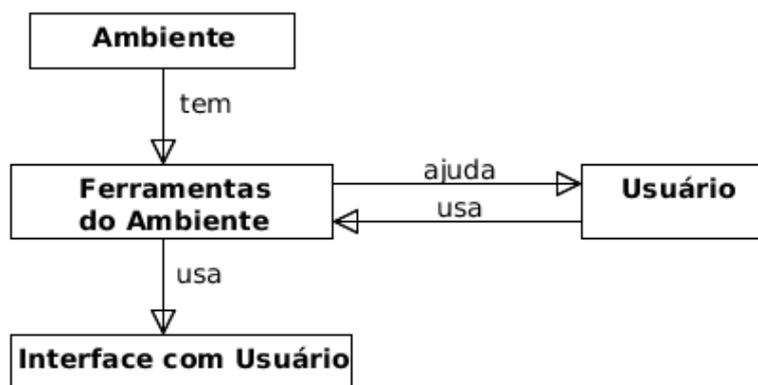
**tarefa** (nos casos onde parte da **tarefa** precisa ser feita fora do **ambiente**). Portanto, a interface com o usuário é o local onde o **usuário** percebe as interações que ele realiza.

A **camada de serviços inteligentes** DEVE ter a capacidade de receber e/ou obter informações da interface com usuário para poder não somente observar as ações do **usuário** como também receber algum dado diretamente do **usuário**.

A **camada de serviços inteligentes** DEVE poder usar a interface com usuário para realizar interações que sejam necessárias com o **usuário** ou para executar alguma **tarefa** em nome do **usuário**.

#### 4.4.3 Ferramentas do Ambiente

Figura 13 – Relações do conceito Ferramentas do Ambiente.



Fonte: Autor.

Ferramentas do ambiente, conforme conceito representado na Figura 13, é definido como um conjunto das funcionalidades ou serviços criados no desenvolvimento do **ambiente** e oferecidos pelo **ambiente** para que os **usuários** possam realizar as **tarefas** dentro do **ambiente**. Elas podem ser divididas em dois tipos: ferramentas de comunicação e ferramentas de trabalho.

- **Ferramentas de comunicação.** são os serviços oferecidos pelo **ambiente** para auxiliar no processo de comunicação entre os **usuários** para a realização das **tarefas**. Como exemplo desse tipo de ferramenta podemos citar: correio eletrônico (e-mail), fax, videoconferência, chamada telefônica via Internet (VOIP), salas de bate-papo (chat) e outras.
- **Ferramentas de trabalho.** são os serviços oferecidos pelo **ambiente** para auxiliar os **usuários** durante a execução de suas **tarefas**. Como exemplo desse tipo de ferramenta podemos citar: agenda eletrônica, editores colaborativos (de texto, imagem, som, etc...), gráfico de Gantt, sistema de arquivo compartilhado e outras.

#### 4.4.4 Tarefa

Figura 14 – Relações do conceito Tarefa.



Fonte: Autor.

A Figura 14 apresenta o conceito de tarefa e os seus relacionamentos sendo que para a definição de tarefa, utilizaremos a mesma definição utilizada por (GHARSELLAOU; BELLIK; JACQUET, 2012) que diz “Tarefas podem ser vistas como ações que devem ser executadas para se alcançar um objetivo em uma aplicação específica de um domínio”.

Portanto, tarefas são as atividades que os **usuários** precisam realizar dentro do **ambiente** para que os seus objetivos sejam atingidos. Tarefas podem ser divididas em várias ações. Por exemplo, durante a execução de uma tarefa de criar um artigo é necessário realizar várias ações:

- criar um documento vazio;
- inserir, alterar e apagar textos no documento;
- salvar o estado atual do texto.

**Axioma 2.** *Uma tarefa é um conjunto de ações.* —

Uma tarefa PODE ser unida a outra tarefa com o objetivo de gerar uma tarefa mais complexa. Por exemplo, as tarefas de criar e revisar um artigo podem ser unidas e resultam na tarefa criar artigo revisado.

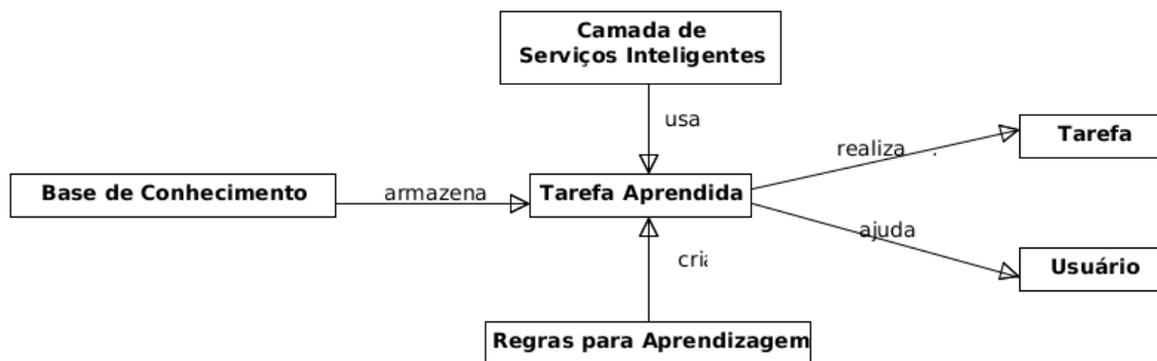
**Axioma 3.** *Uma tarefa pode ser composta por outras tarefas.* —

#### 4.4.5 Tarefa Aprendida

Na Figura 15 apresentamos o conceito de tarefa aprendida e os seus relacionamentos com outros conceitos. As tarefas aprendidas ficam armazenadas na **base de conhecimento** e são usadas pela **camada de serviços inteligentes** para realizar **tarefas** do **usuário** ou ajudar o **usuário** na execução de outras **tarefas**.

Uma tarefa aprendida PODE ser derivada de uma **tarefa** existente ou que existiu no **ambiente**. Quando uma **tarefa** é incorporada ao conjunto de funcionalidades do

Figura 15 – Relações do conceito Tarefa Aprendida.



Fonte: Autor.

sistema uma nova tarefa aprendida é criada e a antiga **tarefa** desaparece (**usuário** não precisa mais realizá-la) ou tem seu contexto reduzido (quando o sistema não realiza toda a **tarefa** mas ainda assim ajuda na sua realização).

Para a criação das tarefas aprendidas a **camada de serviços inteligentes** DEVE fazer uso de um conjunto de **regras para aprendizagem** que vão especificar como acontece o processo de criação das tarefas aprendidas dentro de um **ambiente** específico.

**Axioma 4.** *Uma tarefa aprendida é uma tarefa que é executada pelo sistema computacional.*

O processo de criação das tarefas aprendidas pode se dar pelas três formas que foram apresentadas em (GHARSELLAUI; BELLIK; JACQUET, 2012): Programada, Treinada ou Deduzida.

Uma tarefa aprendida programada é uma **tarefa** que foi ensinada diretamente para o **ambiente** pelo programador, projetista do sistema ou **usuário**. É o tipo mais comum que se apresenta hoje em dia nos ambientes que são considerados inteligentes. Como exemplo desse tipo de tarefa aprendida temos o conjunto de funcionalidades existente na criação do **ambiente** e o conjunto de funcionalidades que são acrescentados em cada nova versão do **ambiente**.

Uma tarefa aprendida treinada é criada quando a **camada de serviços inteligentes** percebe um padrão repetitivo de invocação de uma ou mais **tarefas** pelo **usuário** e passa então a aprender e antecipar as ações ou necessidades do **usuário**. O ponto principal aqui é que o ambiente PODE ter um mecanismo para a detecção dos padrões de interação de um **usuário** e a sua conversão em tarefas aprendidas.

Uma tarefa aprendida deduzida é criada quando a **camada de serviços inteligentes** inclui algumas regras que permitem algumas **tarefas** serem iniciadas pelo sistema

de forma a satisfazer necessidades especiais do **usuário** baseado no seu perfil. Diferente do caso das tarefas treinadas o ponto principal aqui é que o ambiente PODE ter um mecanismo para a detecção de padrões de interação por grupos de **usuários** e a sua convensão em tarefas aprendidas.

**Axioma 5.** *As tarefas aprendidas são criadas através de um processo de programação, treinamento ou dedução de tarefas executadas pelos usuários.* —

Quanto a forma como as tarefas aprendidas podem ser iniciadas também adotamos a classificação usada em (GHARSELLAOUI; BELLIK; JACQUET, 2012): Automática, Implicitamente Interativa e Explicitamente Interativa.

Uma tarefa aprendida automática é uma **tarefa** que é executada automaticamente pelo **ambiente** sem que o **usuário** precise realizar alguma ação para ela ocorrer.

Uma tarefa aprendida implicitamente interativa é uma **tarefa** que é executada sempre que um determinado gatilho ocorre. O gatilho pode ser uma ação do **usuário**, uma condição interna do **ambiente** ou uma combinação de ambos.

Uma tarefa aprendida explicitamente interativa é uma **tarefa** onde o **usuário** solicita diretamente a execução da **tarefa** pelo **ambiente**.

(GHARSELLAOUI; BELLIK; JACQUET, 2012) classifica ainda um quarto tipo que seriam as **tarefas** executadas diretamente pelo **usuário**. Aqui não usamos essa classificação pois esse tipo de **tarefa** seria justamente as **tarefas** que ainda não foram aprendidas pelo ambiente.

**Axioma 6.** *As tarefas aprendidas são executadas das seguintes formas: automaticamente, explicitamente interativa com o usuário ou implicitamente interativa com o usuário.* —

Na Tabela 9 apresentamos exemplos de possíveis tarefas aprendidas segundo a definição aqui utilizada.

#### 4.4.6 Base de Conhecimento

A base de conhecimento, conforme mostrado na Figura 16, é um local onde a **camada de serviços inteligentes** armazena tanto o conhecimento estruturado como o não-estruturado obtido e gerenciado durante a vida útil do sistema computacional. No trabalho aqui apresentado o conceito de base de conhecimento é utilizada de forma mais abrangente pois o conhecimento guardado inclui não apenas as **tarefas aprendidas** como também outras partes necessárias a definição e execução da **tarefa aprendida**, tais como: ontologias, sistemas de inferência, regras de associação e outros.

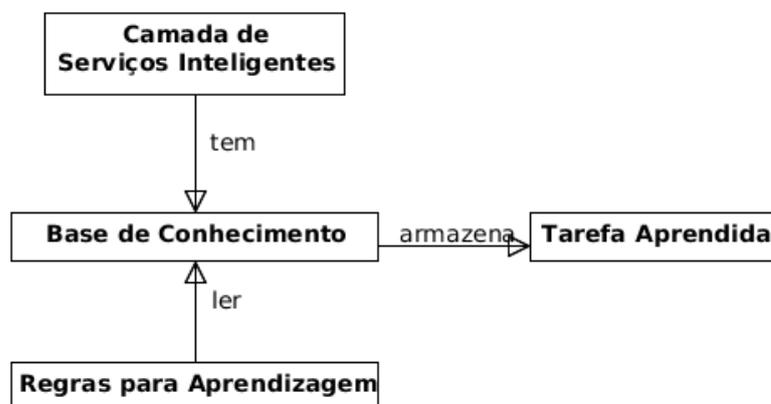
A base de conhecimento de uma **camada de serviços inteligentes** DEVE ter como parte dela uma base de dados.

Tabela 9 – Exemplos de possíveis tarefas aprendidas.

	<b>Automática</b>	<b>Implicitamente Interativa</b>	<b>Explicitamente Interativa</b>
<b>Programada</b>	Envio de resumo de postagens em fóruns em um horário programado pelo usuário.	Sempre que o usuário entra no ambiente mostra uma lista de mensagens não lidas.	O usuário solicita ao sistema para mostrar um lembrete a ele num determinado horário.
<b>Treinada</b>	Aviso de novas postagens em fóruns, aprendida observando os hábitos de leitura de fóruns do usuário.	Sempre que o usuário entra no ambiente mostra uma lista de mensagens não lidas pois usuário fez isso tantas vezes que o ambiente aprendeu.	Conforme o usuário vai postando em fóruns o ambiente aprende quais posts devem ser publicados na hora ou quais devem ser atrasados pois o usuário acaba reeditando alguns posts
<b>Deduzida</b>	Aviso sobre a criação de um novo fórum pois ele é similar a outros fóruns que o usuário acompanha.	Redireciona novas mensagens para o celular do usuário (deduzida do fato do GPS/IP do usuário indicar que ele não está nem no campus nem na sua casa)	Quando o usuário começa a assistir um vídeo o ambiente atrasa o recebimento de mensagens pelo usuário para que ele não tenha distrações enquanto está vendo o vídeo.

Fonte: Autor.

Figura 16 – Relações do conceito Base de Conhecimento.

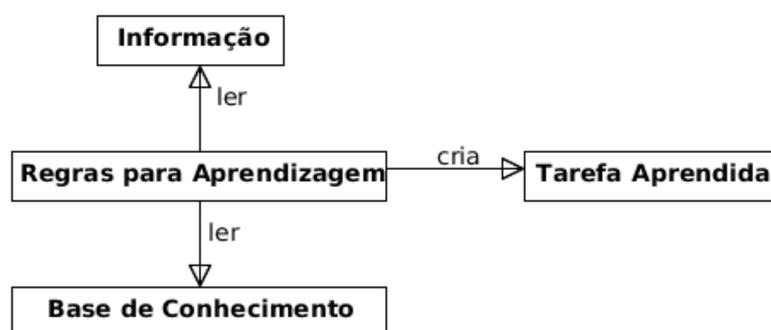


Fonte: Autor.

A base de conhecimento cumpre dois objetivos dentro da **camada de serviços inteligentes**: o primeiro é armazenar as **tarefas aprendidas** e o segundo é servir de entrada para os processos das **regras de aprendizagem** para a geração de novas tarefas aprendidas.

#### 4.4.7 Regras para Aprendizagem

Figura 17 – Relações do conceito Regras para Aprendizagem.



Fonte: Autor.

Na Figura 17 apresentamos o conceito de regras para aprendizagem que são um conjunto geral de regras e definições utilizadas para a criação de novas **tarefas aprendidas** dentro da **camada de serviços inteligentes**.

As regras para aprendizagem PODE ter:

- Um conjunto de protocolos descrevendo casos e condições para criação de novas **tarefas aprendidas**;
- Um conjunto de algoritmos para serem utilizados por novas **tarefas aprendidas**.

Para o seu funcionamento as regras para aprendizagem NECESSITA ter acesso ao conteúdo da **base de conhecimento** da **camada de serviços inteligentes** e das **informações** do ambiente.

#### 4.4.8 Informação

Um dos conceitos principais do nosso modelo de referência é o conceito de Informação apresentado na Figura 18. Para a definição de Informação no contexto deste trabalho foi utilizado o *Knowledge Continuum* conforme utilizado por (BAKER, 2007) apud (ELIAS, 2011) e descrito em (ROWLEY, 2007).

Figura 18 – Relações do conceito Informação.



Fonte: Autor.

**Axioma 7.** *Dado é um fato observável dentro do Ambiente Virtual. Ele pode ser gerado tanto pelo Usuário como pelo próprio ambiente. Exemplo: Acessar um fórum, ter nota 5 num trabalho, ser aprovado num curso, etc.* —

**Axioma 8.** *Uma Informação é o resultado da fusão de um Dado com o seu significado. Para isso é necessário responder a pelo menos um dos seguintes tipos de perguntas: Quem? O que? Quando? Onde? e Quantos?. Exemplo: João acessou o fórum duas vezes, José tem nota 5 no trabalho e Joaquim foi aprovado no curso.* —

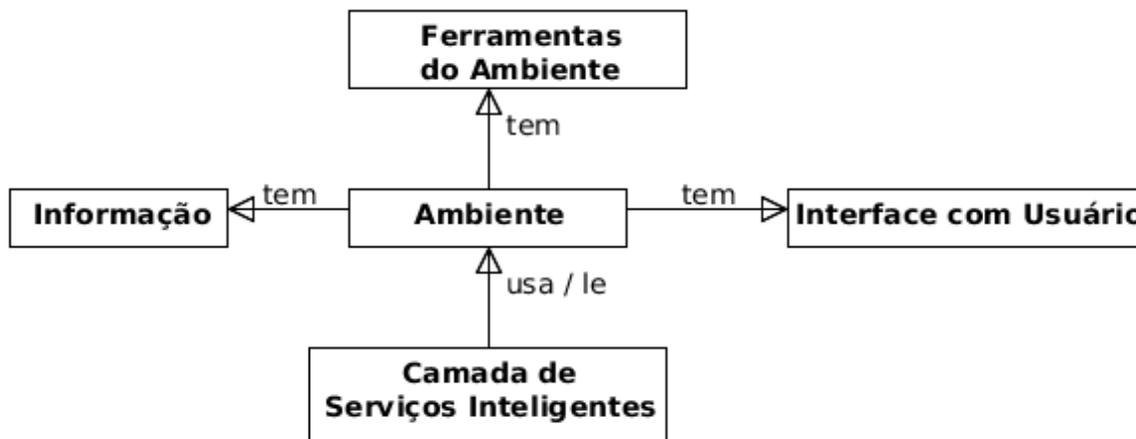
As Informações são geradas tanto pela interação dos **usuários** com o **ambiente** como pelo próprio **ambiente**. O local onde as informações são armazenadas pode ainda ser dividido em duas partes:

- **Base de Dados do Ambiente.** Local onde são armazenadas os dados e informações necessários ao intrínseco funcionamento do **ambiente**. Dados podem ser removidos, acrescentados ou alterados a qualquer momento. As **ferramentas do ambiente** armazenam os seus dados neste local. Como exemplo dos tipos de dados e informações armazenadas podemos citar: Informações cadastrais dos usuários, documentos gerados pelos usuários e itens de configuração do ambiente.
- **Registro de Eventos do Ambiente.** Local onde se tem um histórico das atividades dentro do **ambiente**, sejam essas atividades executadas pelo próprio **ambiente** ou pelo **usuário**. As **ferramentas do ambiente** geram e registram eventos neste local. O registro de eventos tem a característica de estar sempre crescendo em volume pois as informações são apenas acrescentadas, nunca sendo retiradas ou alteradas. Como exemplos de eventos podemos citar: começar a utilizar uma ferramenta, parar de utilizar uma ferramenta, se autenticar no ambiente e enviar uma mensagem para outro usuário.

#### 4.4.9 Ambiente

A Figura 19 apresenta o conceito de ambiente no contexto da nossa proposta. O Ambiente engloba não apenas o local onde acontecem as interações entre os **usuários** e entre

Figura 19 – Relações do conceito Ambiente.



Fonte: Autor.

o **usuário** e o ambiente. Ele engloba também outras áreas necessárias ao funcionamento do sistema computacional tais como sistemas de arquivo, bases de dados e código-fonte.

Do ponto de vista do **usuário** uma definição para ambiente bastante útil é a que foi utilizada por (SANTOS, 2013) para a definição de ambiente virtual:

“pode-se caracterizar ambiente virtual como um mundo virtual no qual o explorador tem a sensação de presença, um espaço de interação por proximidade dentro do qual o explorador pode controlar diretamente um representante de si mesmo.” (SANTOS, 2013)

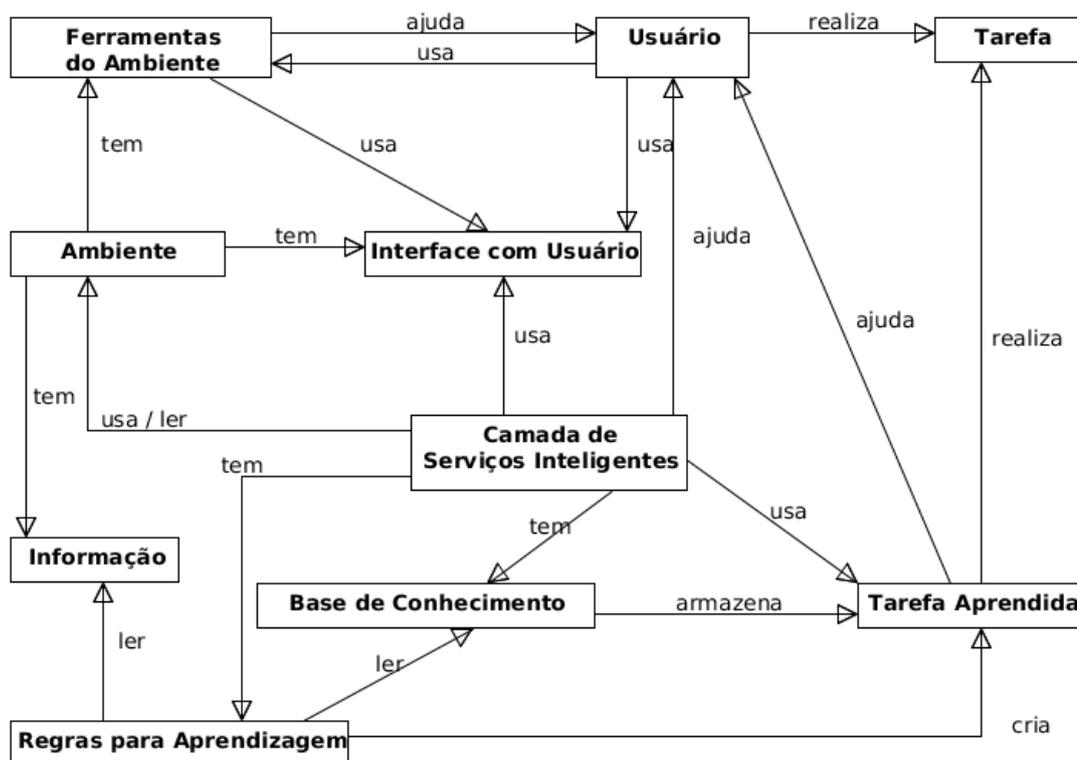
#### 4.4.10 Camada de Serviços Inteligentes

Com a junção dos outros conceitos podemos agora apresentar na Figura 20 o mapa conceitual para o conceito de camada de serviços inteligentes.

Uma camada de serviços inteligentes usa as **tarefas aprendidas** para ajudar o **usuário** a realizar as **tarefas** ou para realizar ela mesmo as **tarefas** diretamente. Para isso a camada de serviços inteligentes tem uma **base de conhecimento** e **regras para aprendizagem**. A camada aplica as regras nas informações existentes do **ambiente** e na sua própria **base de conhecimento**, gerando assim novas **tarefas aprendidas**. Para executar as **tarefas aprendidas** a camada de serviços inteligentes pode usar a **interface com usuário** ou um acesso interno ao **ambiente**, com isso ela tem acesso às **ferramentas do ambiente**. Ferramentas essas que são as mesmas que o **usuário** utilizaria ao realizar ele mesmo as **tarefas**.

Conforme explicitado no modelo de referência, a criação das **tarefas aprendidas** é um processo contínuo e que se retroalimenta. Isto acontece pois para prover os serviços inteligentes para um **ambiente** a camada de serviços inteligentes usa as **regras para**

Figura 20 – Modelo de referência para uma Camada de Serviços Inteligentes.



Fonte: Autor.

**aprendizagem** para ler o conteúdo da sua **base de conhecimento** e o conteúdo das informações do **ambiente** para criar as **tarefas aprendidas**, que são então integradas a sua própria **base de conhecimento** podendo ser utilizadas na construção de novas **tarefas aprendidas**, conforme o processo de criação de **tarefas aprendidas** mostrado na Figura 21.

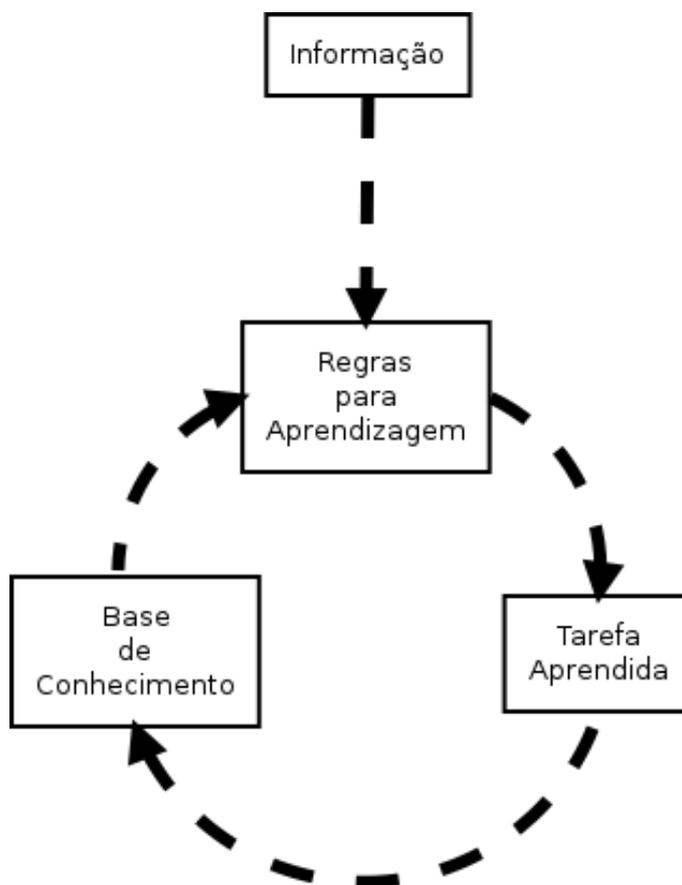
Portanto podemos dividir o trabalho da camada de serviços inteligentes em duas áreas distintas:

- A capacidade de executar as **tarefas aprendidas**;
- A capacidade de gerar novas **tarefas aprendidas**.

Se não tiver a capacidade de executar as **tarefas aprendidas** o uso da camada fica limitado a apenas elencar requisitos a serem incorporados ao ambiente.

Se não tiver a capacidade de gerar novas **tarefas aprendidas** a camada vai ter um limite na capacidade de diminuir a sobrecarga cognitiva dos seus clientes e portanto, corre um risco maior de ficar obsoleta com a passagem do tempo.

Figura 21 – Processo de criação das tarefas aprendidas.



Fonte: Autor.

## 4.5 Usando o Modelo de Referência

Além do uso que fizemos do modelo de referência na criação de uma arquitetura de referência para a criação de uma camada de serviços inteligentes podemos utilizar o modelo para fazer comparações de ambientes que fornecem serviços inteligentes, como exemplo desse uso temos os seguintes comentários sobre alguns dos trabalhos presentes no capítulo de trabalhos relacionados:

- Em (ANAYA; BOTICARIO, 2010) teríamos um exemplo da convenção de uma tarefa para uma tarefa aprendida que poderia ser incorporada por sistemas onde seu uso fosse relevante;
- Em (AZEVEDO; TAVARES, 2001) temos a falta de regras de aprendizagem pois o ambiente foi idealizado com um conjunto fechado de tarefas aprendidas (as funcionalidades);
- Em (BICA; VERDIN; VICARI, 2006) temos outro exemplo de convenção de uma tarefa para uma tarefa aprendida, e além disso é demonstrado que a tarefa aprendida,

descrita no artigo, melhora a eficiência de um ambiente de ensino aprendizagem. O que vem ao encontro das ideias descritas neste capítulo.

Portanto acreditamos que o modelo de referência aqui descrito contribui nas pesquisas relacionadas com ambientes web inteligentes e mais especificamente com ambientes colaborativos flexíveis.

## 4.6 Conclusão do Capítulo

Neste capítulo foi apresentada a analogia do Assistente Humano Virtual (AHV) que serviu de inspiração para a definição de como uma camada de serviços inteligentes deveria se comportar. Nossa visão é que os sistemas computacionais serão cada vez mais sofisticados a medida em que o seu comportamento vai se assemelhando ao que esperamos de um bom assistente humano virtual.

Baseado no AHV foi construído um modelo de referência para a criação de uma camada de serviços inteligentes. Foram discutidos os principais relacionamentos dos principais conceitos do modelo proposto:

- **Usuário.** Um humano que utiliza o ambiente.
- **Interface com Usuário.** Local e o meio por onde ocorrem as interações entre os usuários e o sistema computacional.
- **Ferramentas do Ambiente.** Um conjunto de funcionalidades do ambiente que são oferecidos pelo ambiente para que os usuários possam realizar as tarefas dentro do ambiente.
- **Tarefa.** Atividades que os usuários precisam realizar dentro do ambiente para que os seus objetivos sejam atingidos.
- **Tarefa Aprendida.** Uma tarefa que é executada pelo sistema computacional.
- **Base de Conhecimento.** Local onde a camada de serviços inteligentes armazena tanto o conhecimento estruturado como o não-estruturado.
- **Regras para Aprendizagem.** Um conjunto geral de regras e definições utilizadas para a criação de novas tarefas aprendidas.
- **Informação.** É o resultado da fusão de um dado com o seu significado.
- **Ambiente.** Local onde acontecem as interações entre os usuários e entre o usuário e o ambiente e os locais necessários a existência de tais eventos.

A principal ideia do modelo é que um ambiente computacional deve aprender a executar as tarefas que os seus usuários realizam nele, essas tarefas aprendidas por sua vez iram fazer com que os usuários tenham um aumento de produtividade visto que eles poderam focar os seus esforços numa quantidade cada vez menor de tarefas. Tal modelo vai servir de guia para o restante deste trabalho seja na construção de uma arquitetura de referência como nas discussões necessárias para implantação de uma prova de conceito.

## 5 O *Framework* Proposto

Neste capítulo descreveremos a criação da arquitetura de referência usada para a criação da camada de serviços inteligentes. Começamos com uma definição do que é uma arquitetura de referência e em seguida a apresentamos e comentamos os seus componentes.

### 5.1 Definição de Arquitetura de Referência

Uma arquitetura de referência, segundo (BROWN et al., 2012), modela os elementos arquiteturais abstratos dentro do domínio de interesse do problema, no entanto, ainda o faz de forma independente das tecnologias, protocolos ou produtos a serem utilizados numa possível implementação da arquitetura. Ela se difere do modelo de referência pois mostra uma figura mais clara da solução ao modelar as entidades existentes no modelo, constituindo um *framework*, um “arcabouço conceitual” para a concepção e desenvolvimento de serviços de software.

Como exemplos de Arquiteturas de Referência podemos citar:

- *Java Platform, Enterprise Edition*<sup>1</sup>;
- A Arquitetura Orientada a Serviços (BROWN et al., 2012);
- AUTOSAR<sup>2</sup>.

Podemos afirmar que o objetivo de uma arquitetura de referência é o de identificar soluções abstratas para os problemas do domínio. A forma como ela consegue atingir esse objetivo é fazendo o mapeamento do modelo de referência para elementos de software. Portanto, uma mesma arquitetura de referência pode ser aplicada a toda uma classe de problemas.

### 5.2 Conversões Utilizadas na Descrição

Assim como no modelo de referência quando um conceito fizer referência para outro conceito dentro da sua descrição o conceito referenciado estará em **negrito**.

As setas usadas na Figura 22 que mostra a arquitetura de referência representam o fluxo de informação e o sentido das setas indica o sentido em que a informação flui. Caso a seta tenha duas direções indicam que o fluxo de informação ocorre nos dois sentidos.

<sup>1</sup> <<http://www.oracle.com/technetwork/java/javaee/overview/index.html>>

<sup>2</sup> <<https://www.autosar.org/>>

Caso a ligação entre os conceitos seja uma linha e não uma seta isto indica que entre os conceitos temos uma troca de informação que é mais complexa e que o fluxo da interação será melhor explicado pelo protocolo utilizado.

### 5.3 Arquitetura de Referência: Detalhamento

Neste trabalho foram identificadas sete classes de serviços inteligentes fornecidos em um ambiente CSCW. Acreditamos portanto que um *framework* para a criação desses serviços deve contemplar essas sete classes, bem como não impedir que serviços de uma classe usem serviços de outra, formando assim uma base para a resolução dos problemas utilizando as estratégias de divisão e conquista e *top-down*. No entanto, o foco do *framework* proposto é prover serviços para Ambientes Colaborativos Flexíveis na Web (ACFW) e não ambientes CSCW em geral. Tal escolha se deve principalmente a dois motivos: O primeiro é devido ao tempo disponível e a necessidade de dar um foco ao trabalho. O segundo é que acredita-se que ao escolher trabalhar com ACFW, tais como o MOrFEU, temos acesso aos dados mais estruturados e devidamente organizados que o ACFW utiliza para se adaptar às necessidades do usuário e esta característica facilitaria o desenvolvimento de serviços mais elaborados.

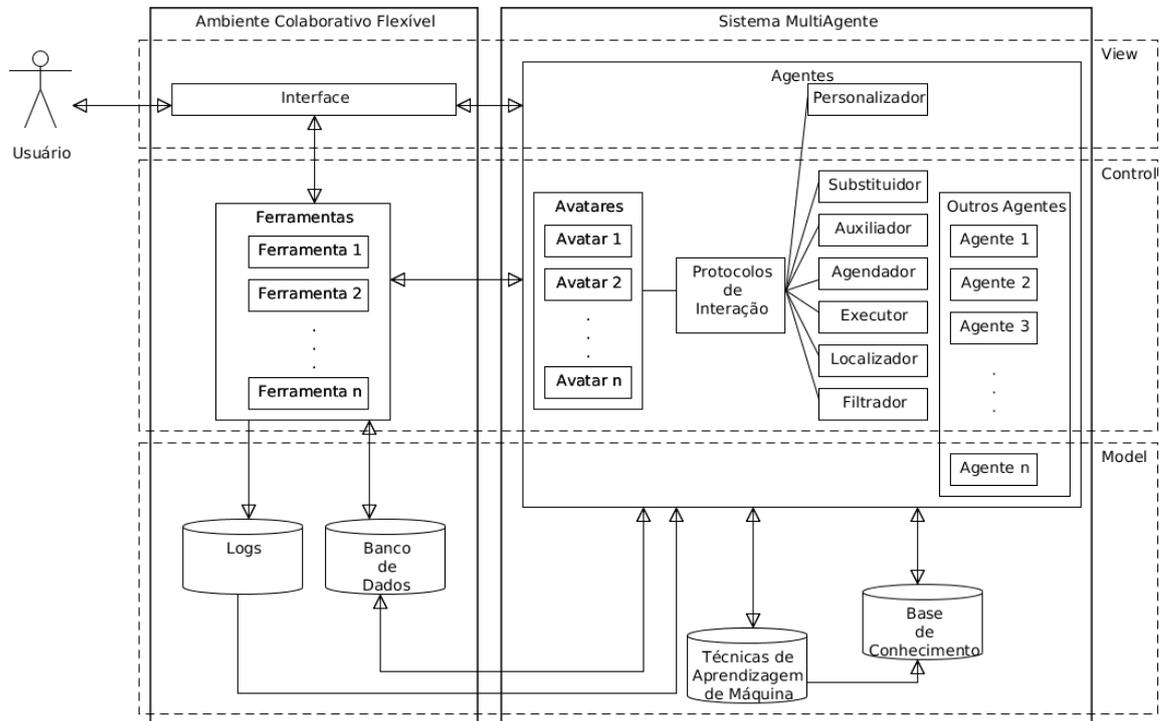
Devido à complexidade de tal *framework* é proposto a utilização de um Sistema Multiagente para a sua implementação. Levando em consideração que a maioria dos artigos da revisão sistemática utilizou-se de SMAs e obteve bons resultados, espera-se reproduzir esse sucesso no *framework* objeto desta proposta.

Apresentamos então na Figura 22 a arquitetura de referência proposta, que será utilizada para a implementação do *framework*.

Utilizados o *design pattern* MVC pois, como explica (MAHMOUD; MAAMAR, 2007), apesar de termos diferenças importantes entre objetos e agentes e isto dificultar e em muitos casos até mesmo impossibilitar o uso de ferramentas de análise e planejamento de soluções orientadas a objeto para soluções orientadas a agentes, não é o caso do uso do MVC. No nosso *framework*, o MVC ajudou a definir os tipos de agentes que seriam necessários à solução, além disso, uma comparação com o MVC utilizado no MOrFEU serviu de inspiração para a nossa solução.

No SMA cada usuário tem um agente avatar representando o usuário. A interação do usuário com o SMA será gerenciada através do seu avatar, o qual vai funcionar como um orquestrador para as operações do SMA para aquele usuário. Para o usuário, o seu avatar vai se comportar de forma parecida a um assistente pessoal, porém as respostas e interações com o seu usuário vão ser inseridas dentro da Interface do ambiente de modo a parecer que é o próprio ambiente quem está respondendo as interações ou requisitando informações.

Figura 22 – Arquitetura de referência utilizada.



Fonte: Autor.

Para cada uma das sete classes de serviços inteligentes para Ambientes Colaborativos opta-se por ter vários agentes daquela classe dentro do SMA, cada um deles oferecendo um serviço representante da classe. Assim, para a classe 1 temos agentes Substituidores, para a classe 2 temos o agentes Agendadores, para a classe 3 temos agentes Localizadores, para a classe 4 temos agentes Auxiliadores, para a classe 5 temos agentes Executores, para a classe 6 temos agentes Filtradores, e para a classe 7 temos agentes Personalizadores.

A opção de utilizar um agente orquestrador (o avatar do usuário) para solicitar serviços para outros agentes orquestradores (das sete classes de serviço) é possibilitar o desenvolvimento de um *framework* genérico onde possa ser possível expandi-lo ou escolher quais partes são interessantes a um Ambiente Colaborativo Flexíveis na Web específico através do uso dos outros tipos de agentes existentes no SMA. Assim poderiam ser especificados novos agentes para resolver problemas pontuais e eles poderiam ser integrados na solução através do Protocolo de Interação executado pelos agentes do SMA.

A seguir é feita uma descrição dos elementos da arquitetura de referência.

### 5.3.1 Usuário

Usuários são os utilizadores humanos do **ambiente colaborativo flexível**. Eles usam o ambiente através da **interface** de onde eles obtêm acesso as **ferramentas** neces-

sárias a realização das suas tarefas.

### 5.3.2 Interface

A Interface é o meio por onde o **usuário** interage com outros **usuários** ou com o **ambiente colaborativo flexível**. É o local onde as **ferramentas** exibem os seus resultados e requisitam ações do **usuário**. Por hospedar essas interações a Interface pode ser utilizada pelos agentes para conseguir informações relevantes à execução das tarefas aprendidas. No entanto, apenas o agente **Personalizador** deve sugerir alterações na forma como a Interface funciona.

### 5.3.3 Ferramentas

Ferramentas são os serviços oferecidos pelo **ambiente colaborativo flexível** para que o **usuário** consiga realizar as suas tarefas dentro do ambiente. Elas podem ajudar tanto na comunicação entre **usuários** (correio eletrônico, fax, videoconferência, chamada telefônica via Internet, salas de bate-papo, etc...) como podem ajudar na execução do trabalho a ser realizado pelo **usuário** (agenda eletrônica, editores de texto, compartilhamento de arquivo, etc...).

As Ferramentas podem ser utilizadas pelos agentes para a execução das tarefas aprendidas. As Ferramentas também registram os eventos em que elas tenham alguma participação nos **logs** e os dados necessários à configuração e execução das mesmas são armazenados no **banco de dados**, assim como os artefatos produzidos pelas Ferramentas.

### 5.3.4 Banco de Dados

O Banco de Dados é o local onde são armazenados os dados e informações necessários ao funcionamento do **ambiente colaborativo flexível**. As **ferramentas** armazenam os seus dados e opções de configuração neste local e recuperam as informações daqui quando é necessário. Em geral o Banco de Dados costuma ser disponibilizado através do uso de um SGBD (Sistema Gerenciador de Banco de Dados).

### 5.3.5 Logs

Os Logs são conjuntos de registros de eventos ocorridos no ambiente. As **ferramentas** geram e registram eventos neste local. Apesar dos Logs terem a característica de estarem sempre crescendo em volume e devido a restrições de espaço, em geral, apenas os registros de eventos mais recentes estão disponíveis, o restante é movido para um armazenamento terciário ou salvo em cópias de segurança. Portanto o **SMA** deve funcionar apenas com uma janela dos registros recentes dos eventos disponíveis.

### 5.3.6 Avatares

Os agentes Avatares tem dois papéis a serem realizados no ambiente: representar internamente os **usuários** e servir de interface entre o **SMA** e os **usuários**. Assim o Avatar de um **usuário** tem a responsabilidade de ser um auxiliar do **usuário** na utilização do **ambiente colaborativo flexível**. Ele pode desempenhar internamente vários papéis dependendo do ambiente: secretário, tutor, companheiro, etc. Esse agente vai utilizar com frequência outros agentes para melhorar o seu próprio desempenho, por exemplo: através de um agente Agendador ele pode sugerir reuniões para o usuário ou pode solicitar a execução de tarefas específicas para um agente Executor. O Avatar é responsável por orquestrar o funcionamento do **SMA** do ponto de vista do **usuário**.

### 5.3.7 Agendadores

Os agentes Agendadores são responsáveis pelo gerenciamento da agenda e atividades dos **usuários**, eles provém serviços tais como agendamento de prazos para as tarefas, compromissos e reuniões dos **usuários**. Um agente Agendador recebe através de um **avatar** informações do **usuário** tais como preferências de horários e tempos livres. Através do **banco de dados** ele pode ter as informações de todas as agendas e horários do ambiente além de outras informações de todos os **usuários**. Assim pode utilizar técnicas de Inteligência Artificial para não somente agendar os compromissos e metas dos **usuários** como também lembrar e incentivar o **usuário** a cumprir as tarefas e agendamentos.

### 5.3.8 Executores

Os agentes Executores se especializam em realizar tarefas no lugar do **usuário**. Os agentes devem ser capazes de utilizar as ferramentas do **ambiente colaborativo flexível** para atingirem os seus objetivos. O usuário pode indicar, através do seu Avatar, quais tarefas o agente pode fazer. Além disso, esse agente pode receber pedidos de outros agentes para realização de tarefas. Exemplo de tarefas projetadas para esse agente seriam: realizar uma postagem num fórum, baixar arquivos necessários ao trabalho do usuário, entregar mensagens para o usuário, etc.

### 5.3.9 Substituidores

Os agentes Substituidores são responsáveis por substituírem o **usuário** em determinadas tarefas. Ela se distingue da classe **Executor** pois assume a responsabilidade pela execução da tarefa. Tarefas repetitivas são boas candidatas a serem repassadas para o Substituidor. Exemplos de possíveis tarefas de responsabilidade desse agente: solicitar ao agente Executor para abrir uma sessão de chat em uma determinada hora todo dia na

semana ou reconfigurar em tempo real os parâmetros de conexão de uma videoconferência, etc.

### 5.3.10 Localizadores

Os agentes Localizadores são responsáveis por procurar informações para os **usuários** em repositórios internos e externos de informação. Eles podem fazer isso de forma preemptiva ou por demanda. Um exemplo de uso deste tipo de agente seria o **usuário** solicitar a um Localizador para procurar páginas *web* relevantes sobre um determinado tema. O agente Localizador após realizar pesquisas em máquinas de busca poderia solicitar ao agente **Executor** o envio de um e-mail com o resultado das suas pesquisas para o **usuário**.

### 5.3.11 Filtradores

Os agentes Filtradores são responsáveis pelo gerenciamento das informações providas de fontes externas. Eles se diferenciam dos **Localizadores** pois eles não vão atrás das informações mas sim funcionam como uma interface de entrada para outros sistemas enviarem informações para o **SMA**. Quando outros sistemas precisam enviar uma informação/mensagem para o **SMA** é um agente Filtrador quem recebe as mensagens e interage com esses sistemas. Um exemplo de funcionamento desse agente seria por exemplo converter mensagens de e-mails externos em mensagens internas para o **usuário** no ambiente, interceptando e descartando eventuais mensagens não-desejadas (SPAMs)

### 5.3.12 Personalizadores

Os agentes Personalizadores atuam na construção personalizada e de forma automática da **Interface**. Eles podem decidir não somente a disposição dos elementos como também aprender com o comportamento do **usuário**. Um agente Personalizador, por exemplo, poderia ter habilidades para detectar se o **usuário** tem alguma deficiência de visão e ajustar a **Interface** (cores, tamanhos de letras, fontes utilizadas) para melhor se adaptar ao **usuário**.

### 5.3.13 Auxiliadores

Os agente Auxiliadores são responsáveis por auxiliar o **usuário** a compreender quais tarefas precisam ser executadas ou como elas devem ser executadas. Ele deve apresentar sugestões de atividades ou soluções para problemas apresentados pelo **usuário**. Um exemplo de uso do Auxiliador seria ele realizar um tour virtual para novos usuários do Ambiente Flexível explicando e mostrando as principais características e ferramentas do ambiente.

### 5.3.14 Outros Agentes

No **SMA** além dos **avatares** e dos agentes especializados descritos anteriormente, temos outros agentes que cumprem um papel de ajudar os outros agentes dentro do sistema. Por exemplo, se o **SMA** tiver um banco de dados dele (diferente do ambiente) então pode ser interessante criar agentes que salvem dados, recuperem dados e otimizem o funcionamento desse banco, as informações armazenadas neste **banco de dados** poderiam inclusive ser utilizadas por mais de um dos agentes do **SMA**.

### 5.3.15 Protocolo de Interação

O Protocolo de Interação deve especificar quais são os procedimentos e processos que os agentes que fornecem os serviços inteligentes devem utilizar para localizar e utilizar outro agentes fornecedores de serviços inteligentes. Além disso, o protocolo deve também descrever como é inserção, alteração e remoção de agentes dentro desses procedimentos e processos.

Podemos dividir o Protocolo de Interação em duas partes:

- Descoberta de Serviços, que descreve como os agentes procuram e utilizam serviços fornecidos por outros agentes para realizarem a suas tarefas;
- Composição de Serviços, que descreve como o agente que está solicitando os serviços deve compor e encadear o uso dos outros agentes durante a realização da tarefa.

### 5.3.16 Base de Conhecimento

É a implementação da Base de Conhecimento do modelo de referência. Pode ser implementada através de um SGBD, um sistema de lógica de primeira ordem ou mesmo arquivos em um sistema de arquivos. O objetivo da Base de Conhecimento é salvar as informações que o *framework* tem no momento, tais como estados atuais do agentes ou informações sobre o ambiente. O **SMA** tem acesso a Base de Conhecimento e deve poder criar, apagar ou atualizar as informações salvas na mesma.

### 5.3.17 Técnicas de Aprendizagem de Máquina

Para a criação de novas tarefas aprendidas pelo *framework*, Técnicas de Aprendizagem de Máquina são utilizadas. Na arquitetura proposta este é um espaço onde ficam armazenados as informações e dados necessários para executar as técnicas. Ela não fica interna aos Agentes pois as técnicas podem ser pesquisadas e utilizadas e somente o resultado ser incorporado ao **SMA**. Esta incorporação pode ser direta quanto é feita através da implementação de novos agentes ou indireta quanto o resultado é salvo na Base de Conhecimento para que os agentes usem essa informação.

## 5.4 Do Modelo para a Arquitetura

Para ficar melhor entendimento da implementação do modelo de referência na arquitetura proposta foi construída a Tabela 10 para apontar quais partes da arquitetura são equivalentes a quais implementações do modelo.

Tabela 10 – Elementos : Modelo de Referência X Arquitetura de Referência

<b>Elemento do Modelo de Referência</b>	<b>Elemento da Arquitetura de Referência</b>
Usuário	Usuário
Interface com Usuário	Interface
Ferramentas do Ambiente	Ferramentas
Tarefa	Na interação entre Usuário e Ferramentas.
Tarefa Aprendida	Nos Agentes do SMA
Base de Conhecimento	Base de Conhecimento
Regras para Aprendizagem	Técnicas de Aprendizagem de Máquina
Informação	Banco de Dados e Logs
Ambiente	Ambiente Colaborativo Flexível
Camada de Serviços Inteligentes	Sistema Multiagente

Fonte: Autor.

## 5.5 Exemplos de Uso

Para ilustrar o funcionamento da camada de Serviços Inteligentes e a sua utilização por um Ambiente Colaborativo Flexível na Web é útil descrever possíveis cenários de uso.

### 5.5.1 Cenário Um – Apoiando um Professor em Ambiente Online

Para esse cenário vamos utilizar o contexto de um ambiente educacional e um professor montando uma tarefa de ensino no ambiente. Quando o professor informar ao ambiente que deseja montar uma nova tarefa um agente Auxiliador poderia perceber qual o tipo de tarefa que o professor está interessado e sugerir dicas (agente GeraDicas) ou exemplos de tarefas semelhantes (agente SugereSimilares) que tenham na base de dados do Ambiente. Outra possibilidade seria de um agente Auxiliador sugerir quais usuários do Ambiente poderiam ser tutores (agente SugereTutor) dessa tarefa baseado nos desempenhos dos alunos em tarefas semelhantes.

Um agente Personalizador poderia então sugerir alterações na interface de forma a apresentar ao usuário as dicas dos agentes Auxiliadores em local de destaque (agente InsereAvisos) ou caso note que o professor prefira ser informado de forma mais discreta (através de interação com o agente Avatar do professor), ele pode apenas colocar um informe avisando que tem maiores informações para o professor.

Após montar a tarefa o professor pode pedir para o Ambiente avisar os alunos de que a tarefa já pode ser feita e quais são os prazos. Um agente Agendador cadas-

tra os prazos nas agendas dos envolvidos (agente GeraPrazo) e através de um agente Substituidor (personificando o professor) informa aos alunos do início da tarefa (agente EnviaMensagensComo), provavelmente personalizando a forma de entrega (através de um agente Personalizador) para cada aluno, no fim do prazo previsto um agente Substituidor é novamente acionado pelo Agendador (agente Alarme) e envia uma mensagem para os alunos lembrando os mesmo do fim do prazo (EnviaMensagensComo).

Como exemplo do que poderia esta salvo na Base de Conhecimento podemos citar:

- Regras de classificações para os tipos de tarefas que podem ser criadas por um professor;
- Regras de classificações para os tipos de alunos;
- Regras de classificações para os tipos de usuários;
- Regras para escolha de Tutores.

Essas regras poderiam ser utilizados pelos agentes (que escolheriam qual deles usar ou fariam uma combinação) para poderem atingir os seus objetivos.

Neste cenário não utilizamos os agentes do tipo Localizador, Filtrador e Executor. No entanto, seria possível refinar mais o ambiente descrito e acrescentar tarefas que fossem de responsabilidade desses agentes. Também não foram utilizadas as Técnicas de Aprendizagem de Máquina. Novamente, nada impede que em posteriores refinamentos essa parte venha a ser acrescentada.

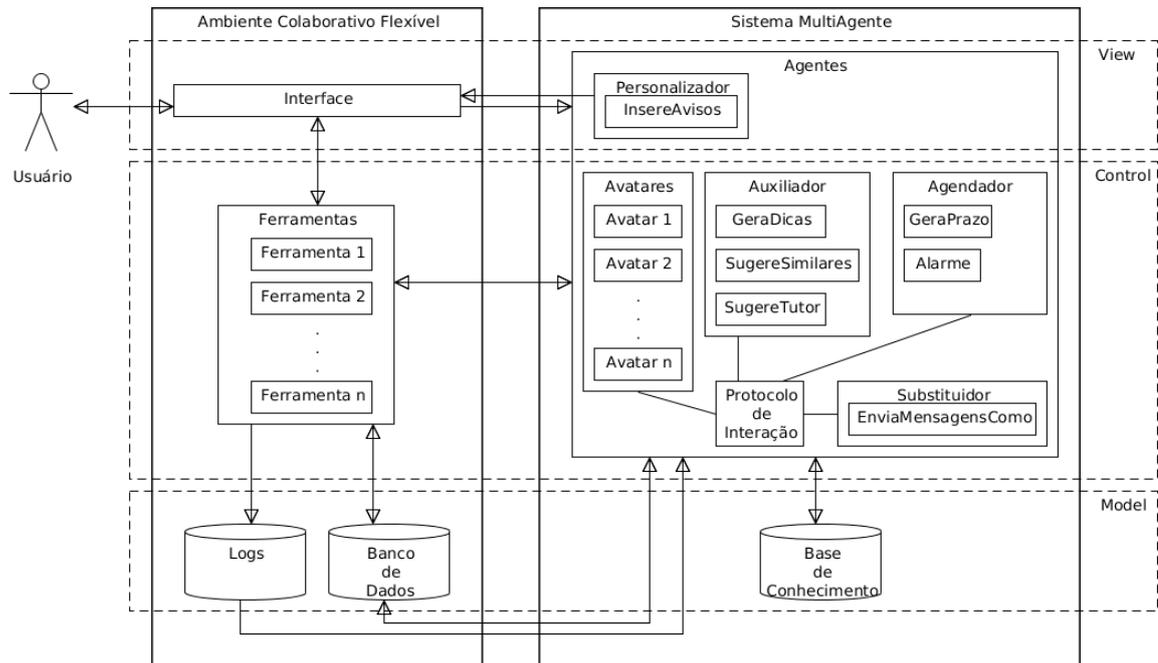
Por fim, na Figura 23 apresentamos a arquitetura do SMA a ser utilizado nesse cenário para a criação dos serviços inteligentes.

### 5.5.2 Cenário Dois – Prevenção de Evasão em Cursos Online

Para uma melhor apropriação das técnicas de Aprendizagem de Máquinas estudadas foi desenvolvido o experimento relatado em (NETO; CASTRO, 2015).

Nele é apresentado uma ferramenta para auxiliar na resolução do problema da desistência por parte dos alunos em cursos online à distância ou semipresenciais. Como o processo de desistência ainda não é totalmente compreendido e tem-se muita discussão no meio acadêmico sobre quais são os motivos que levam um aluno a desistência, existindo inclusive modelos propostos para tentar explicar esse fenômeno conforme relatado em (LEE; CHOI; KIM, 2013; PARK; CHOI, 2009; PARK, 2007), foi testado a possibilidade de se criarem processos que possam auxiliar o professor ou tutor a identificar, com eficiência, possíveis desistentes dentre os alunos sob a sua tutela.

Figura 23 – Cenário Um.



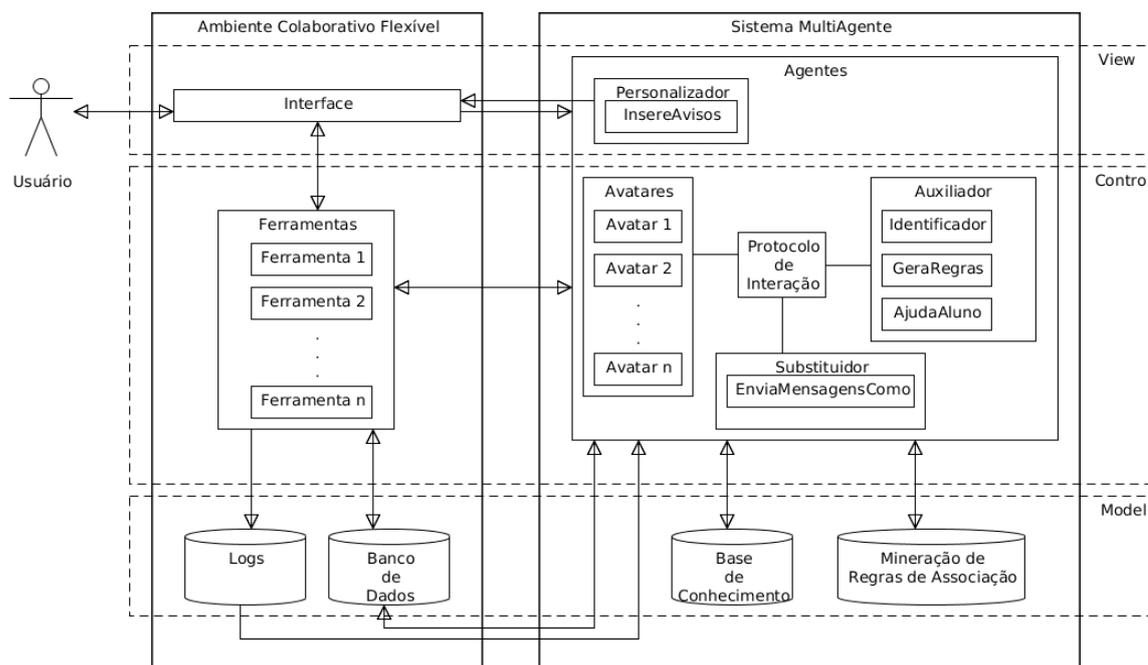
Fonte: Autor.

Neste contexto, o artigo apresentou a utilização de técnicas de aprendizagem de máquina no contexto de mineração de dados para analisar um grande volume de dados de estudantes de cursos online em um Ambiente Virtual de Aprendizagem com objetivo de descobrir Regras de Associação que poderiam ser utilizadas para identificar estudantes que estejam em risco de desistir do curso. Regras de Associação é uma técnica que procura descobrir padrões interessantes em bancos de dados que podem então serem utilizados para prever novos acontecimentos ou classificar novos eventos semelhantes aos já registrados. Para atingir o objetivo de identificar possíveis desistentes a ferramenta faz uma abordagem híbrida para a detecção ao utilizar tanto conhecimento elencado de especialistas como as informações disponíveis para mineração de dados em um Ambiente Virtual de Aprendizagem.

Trazendo esses resultado para o contexto do *framework* de serviços inteligentes, essa detecção de possíveis desistentes poder ser um serviço inteligente disponível em um ambiente. Tal serviço poderia ser implementado conforme a arquitetura apresentada na Figura 24 e comentada a seguir.

Um agente Auxiliador indica os alunos com potencial de serem desistentes (através do agente Indicador) e armazena essa informação na Base de Conhecimento. Para fazer a indicação ele utiliza Regras de Associação (RA) que também estão na Base de Conhecimento. As RA por sua vez são criadas através de um processo de Mineração de Regras de Associação. Tal processo de mineração é feito pelo agente GeraRegras, mas pode ser feito

Figura 24 – Cenário Dois.



Fonte: Autor.

externamente ao ambiente e as RA inseridas diretamente na Base de Conhecimento.

Após identificar os alunos, o sistema poderia iniciar um processo onde fossem enviadas mensagens personalizadas (geradas pelo agente AjudaAluno) aos alunos com risco de evasão. Sendo que tais mensagens seriam como se tivessem sido enviados por um professor/tutor (agente EnviaMensagensComo). Por fim, o professor/tutor teria um aviso na sua Interface com o sistema informando sobre os alunos com risco de evasão (agente InserirAvisos).

## 5.6 Conclusão do Capítulo

Neste capítulo foi explicado o que é uma arquitetura de referência e descrevemos a nossa arquitetura de referência para a construção de uma camada de sistemas inteligentes. Os principais conceitos da arquitetura de referência proposta neste trabalho são:

- **Usuário.** Os utilizadores humanos do ambiente colaborativo flexível.
- **Interface.** O meio por onde o usuário interage com outros usuários ou com o ambiente colaborativo flexível.
- **Ferramentas.** Serviços oferecidos pelo ambiente colaborativo flexível para que o usuário consiga realizar as suas tarefas.

- **Banco de Dados.** Local onde são armazenadas os dados e informações necessários ao funcionamento do ambiente colaborativo flexível.
- **Logs.** São os conjuntos de registros dos eventos que ocorreram dentro do ambiente, sejam ou não causados pelos usuários.
- **Agentes Avatares.** Representam internamente os usuários e servem de interface entre o SMA e os usuários.
- **Agentes Agendadores.** Responsáveis pelo gerenciamento da agenda e atividades dos usuários.
- **Agentes Executores.** Responsáveis por realizar tarefas no lugar do usuário, o usuário ainda pode executar as tarefas ou delega-las ao agente.
- **Agentes Substituidores.** Responsáveis por substituírem o usuário em determinadas tarefas, o ambiente passa então a executar a tarefa e os usuários não mais precisam realizar ela.
- **Agentes Localizadores.** Responsáveis por procurar informações para os usuários em repositórios internos e externos de informação.
- **Agentes Filtradores.** Responsáveis pelo gerenciamento das informações provindas de fontes externas.
- **Agentes Personalizadores.** Responsáveis pela construção personalizada e automática da Interface.
- **Agentes Auxiliadores.** Responsáveis por auxiliar o usuário a compreender quais tarefas precisam ser executadas ou como elas devem ser executadas.
- **Protocolo de Interação.** Especifica quais são os procedimentos e processos que os agentes que fornecem os serviços inteligentes devem utilizar para localizar e utilizar outro agentes fornecedores de serviços inteligentes.
- **Base de Conhecimento.** Implementação da Base de Conhecimento do modelo de referência proposto nesta tese.
- **Técnicas de Aprendizagem de Máquina.** Responsável pela criação das novas tarefas aprendidas pelo *framework*.

A arquitetura proposta é portanto formada principalmente por agentes que espelham as classes de serviços inteligentes para os Ambientes Colaborativos Flexíveis na Web. Através da interação entre esses agentes o ambiente pode fornecer aos seus usuários serviços cada vez mais complexos.

---

Por fim, apresentamos dois cenários de exemplo usando a arquitetura aqui proposta para especificar uma solução que resolva as questões apresentadas nos cenários. Importante notar que para a construção de uma solução não é necessário utilizar todos os tipos de agentes possíveis apresentados, mas sim apenas aqueles que vão ser de fato utilizados na solução.

## 6 Implementação da Camada de Serviços Inteligentes

Neste capítulo descrevemos os procedimentos, processos e escolhas que foram feitas durante a implementação e instanciação de uma camada de serviços inteligentes para o MOrFEU com objetivo de criar uma prova de conceito para a arquitetura de referência proposta neste trabalho.

Salienta-se que o *framework* objetivo deste trabalho é a Arquitetura de Referência descrita no [Capítulo 5](#), no entanto para comprovar a factabilidade dos resultados que encontramos optamos por realizar uma implementação da mesma. Tal implementação (que chamamos de Fantasos) também pode ser considerada um *framework* a nível de código fonte e serviu como prova de conceito do funcionamento de uma camada de serviços inteligentes seguindo as ideias e conceitos apresentados neste trabalho. Neste sentido, podemos dizer que o Modelo de Referência e a Arquitetura de Referência se aproximam mais do sentido que a palavra *framework* tem na área de Inteligência Artificial enquanto o Fantasos se aproxima mais do sentido que a palavra *framework* tem na Engenharia de Software.

Portanto, neste capítulo vamos descrevermos as análises e decisões de projeto que foram realizadas com o objetivo de construir uma implementação da nossa Arquitetura de Referência.

### 6.1 O Protocolo de Interação

Para o desenvolvimento de uma instanciação do *framework* é necessário descrever e implementar um Protocolo de Interação a ser utilizado pelos agentes dentro do SMA. Esse protocolo deve descrever como os agentes localizam e utilizam os serviços oferecidos por outros agentes e como esses serviços são encadeados para poder atender a necessidade de uma tarefa específica.

Neste sentido, optou-se pelo uso da linguagem LCC (*Lightweight Coordination Calculus*) descrita em ([ROBERTSON, 2004](#)) bem como da implementação provida em ([DUPPLAW; BESANA; ROBERTSON, 2008](#)), pois assim abordamos com pouco esforço as duas partes do Protocolo de Interação: Descoberta de Serviços e Composição dos Serviços.

A seguir fazemos um resumo dos problemas de descoberta de serviços e composição de serviços, da linguagem LCC e como ela trata esses dois problemas para em seguida descrevermos um modelo de interação básico para a implementação que foi feita.

### 6.1.1 O Problema da Descoberta de Serviços

Ao analisar o problema do desenvolvimento do protocolo de interação foi percebido que uma parte do protocolo de interação se tratava do problema da descoberta de serviços. Uma das descrições no protocolo de interação é uma funcionalidade que permita que os agentes avatares possam procurar por outros agentes que fornecem serviços inteligentes apropriados à tarefa que o agente avatar está ajudando o usuário a realizar.

Segundo (BETTSTETTER; RENNER, 2000) este tipo problema é alvo dos estudos dos protocolos de descoberta de serviços (service discovery protocols) e como exemplos de protocolos desta classe temos:

- SLP (Service Location Protocol) desenvolvido pelo IETF;
- Jini (uma extensão da linguagem de programação Java);
- UPnP (Universal Plug and Play) atualmente mantido pela Open Connectivity Foundation (OCF);
- SDP (Bluetooth Service Discovery Protocol).

Em (BETTSTETTER; RENNER, 2000) é descrito que um protocolo de descoberta de serviços deve fornecer informações sobre:

- Como localizar e navegar pelos serviços;
- Como escolher o serviço correto;
- Como utilizar o serviço.

Apesar do problema de descoberta de serviços ter começado na área de redes de computadores sua aplicação já foi alvo de estudos dentro de Sistema Multiagentes (SMA) conforme pode-se ver em (CAMPO, 2002; BURSTEIN et al., 2005; KANG; SIM, 2011; RUTA et al., 2014).

### 6.1.2 Problema da Composição dos Serviços

O problema da composição de serviços também já é um problema conhecido e estudado em SMA, como exemplos podemos citar os trabalhos de (ALIBHAI, 2003), (BLAKE, 2003), (GUTIERREZ-GARCIA; SIM, 2010) e (MAITHREYE; DEVI; ENGINEERING, 2016). Este é o problema que os softwares de workflow tentam resolver no contexto de organizações.

O problema consiste em descrever como os agentes devem utilizar os serviços disponíveis (que já foram localizados no ambiente) para a realização das tarefas enquanto

resolve os problemas ocasionados por conflitos, mudanças de prioridades e gerenciamento de recursos.

### 6.1.3 A linguagem LCC

Segundo (ROBERTSON, 2004) a linguagem LCC pode ser utilizada para descrever as normas sociais que regem as interações entre os agentes como processos distribuídos interagindo entre si. Ela consegue representar os conceitos essenciais para a representação e raciocínio sobre normas sociais interferindo o menos possível nas operações e implementações dos agentes. Assim a linguagem LCC permite não apenas uma especificação declarativa como também a execução de um modelo de interação. Outro ponto é que como a LCC é baseada em cálculo de processos ela suporta a validação formal dos modelos de interação nela criados.

Os modelos de interação expressos em LCC são construídos usando uma ou mais cláusulas. Cada uma delas define um papel. Cada papel detalha todas as informações necessárias a execução do papel, incluindo as trocas de mensagens.

Para resolver o problema da descoberta de serviços a implementação provida em (DUPPLAW; BESANA; ROBERTSON, 2008) utiliza um serviço para registro e consulta de serviços, todos os agentes que podem realizar algum papel no Modelo de Interação (MI) devem se registrar nesse serviço e quando o MI está sendo executado este serviço é consultado para verificar quais agentes vão ser acionados.

Para resolver o problema da composição de serviços são utilizados os papéis que fazem parte do MI. Assim dois agentes diferentes (implementações diferentes) podem fornecer o mesmo serviço final (mesmo papel no MI) e o próprio MI descreve como se dá a composição do serviço, sendo que um MI diferente pode compor os mesmos serviços de outra forma.

#### 6.1.3.1 Sintaxe da Linguagem LCC

A sintaxe da linguagem LCC é baseada em Cálculo de Processos (*Process Calculus*) sendo a primeira linguagem a utilizá-la diretamente para modelar normas sociais em Sistemas Multiagentes (ROBERTSON, 2004). Em LCC os comportamentos mais básicos são enviar e receber mensagens. Sendo que tanto o ato de enviar como o de receber uma mensagem podem ser alterados através do uso de restrições. As restrições são codificadas por funções utilizando Cálculo de Predicados de Primeira Ordem (*first order predicate calculus*). Comportamentos complexos são gerados através do uso dos conectores *then* (sequência), *or* (escolha) e *par* (paralelização). Um conjunto de cláusulas especificando comportamentos formam o que chamamos de Modelo de Interação.

Figura 25 – Sintaxe do LCC.

```

Framework := {Clause,...}
Clause := Agent :: Def
Agent := a(Type, Id)
Def := Agent | Message | Def then Def | Def or Def | Def par Def | null ← C
Message := M ⇒ Agent | M ⇒ Agent ← C | M ⇐ Agent | M ⇐ Agent ← C
C := Term | C ∧ C | C ∨ C
Type := Term
M := Term

```

Fonte: (ROBERTSON, 2004).

A Figura 25 apresenta a sintaxe da linguagem LCC em formato BNF<sup>1</sup>. Onde *null* denota um evento que não envolve troca de mensagens. *Term* é o mesmo que um termo na sintaxe da linguagem Prolog e *Id* pode ser uma variável ou um identificador único de um agente. Os operadores  $\leftarrow$ ,  $\wedge$  e  $\vee$  são os operadores lógicos para condicional, conjunção e disjunção.  $M \Rightarrow A$  representa uma mensagem, aonde  $M$  é a mensagem e  $A$  é o agente que vai receber a mensagem.  $M \Leftarrow A$  significa que a mensagem  $M$  do agente  $A$  foi recebida. Por fim, o operador condicional tem prevalência sobre os operadores de mensagem.

### 6.1.3.2 Um Exemplo

Para ilustrar a utilização e o funcionamento do LCC vamos usar um exemplo de solicitação simples de um serviço. Neste cenário temos duas partes, uma fazendo o papel do solicitante do serviço e outra fazendo o papel do executor do serviço.

Uma possível descrição deste Modelo de Interação é a seguinte: O agente solicitante envia uma mensagem para o agente executor solicitando a execução da tarefa, o agente executor então vai realizar a tarefa e envia uma mensagem para o agente solicitante avisando que a tarefa foi feita.

O código (com comentários) desse Modelo de Interação escrito em LCC é apresentado abaixo:

```

1 // Descricao do papel solicitante neste Modelo de Interacao
  a(solicitante, S) ::
    // Instancia a variavel Input com um valor
    null ← getInput(Input)
5 // Envia mensagem para o executor solicitando executar a tarefa
  doTask(Input) => a(executor, E) then
    // Recebe do executor uma mensagem informando que a tarefa foi
      executada
    taskDone(Output) <= a(executor, E)
9
  // Descricao do papel executor neste Modelo de Interacao
  a(executor, E) ::
    // Recebe do solicitante uma mensagem solicitando executar a tarefa
13 doTask(Input) <= a(solicitante, S) then

```

<sup>1</sup> <[https://en.wikipedia.org/wiki/Backus-Naur\\_form](https://en.wikipedia.org/wiki/Backus-Naur_form)>

```
// Executa a tarefa
null <- executeTask(Input,Output) then
// Envia uma mensagem para o solicitante informando que a tarefa foi
  executada
17 taskDone(Output) => a(solicitante)
```

Código 1 – Código LCC de Exemplo

Assim para a execução desse modelo são necessários a existência de dois agentes, um realizando o papel do executor e outro realizando o papel do solicitante. O agente que for realizar o papel do solicitante deve implementar a função `getInput()` e o agente que for realizar o papel do executor deve implementar a função `executeTask()`. As trocas de mensagens são implementadas através da arquitetura provida pelo OpenKnowledge Kernel (DUPPLAW; BESANA; ROBERTSON, 2008).

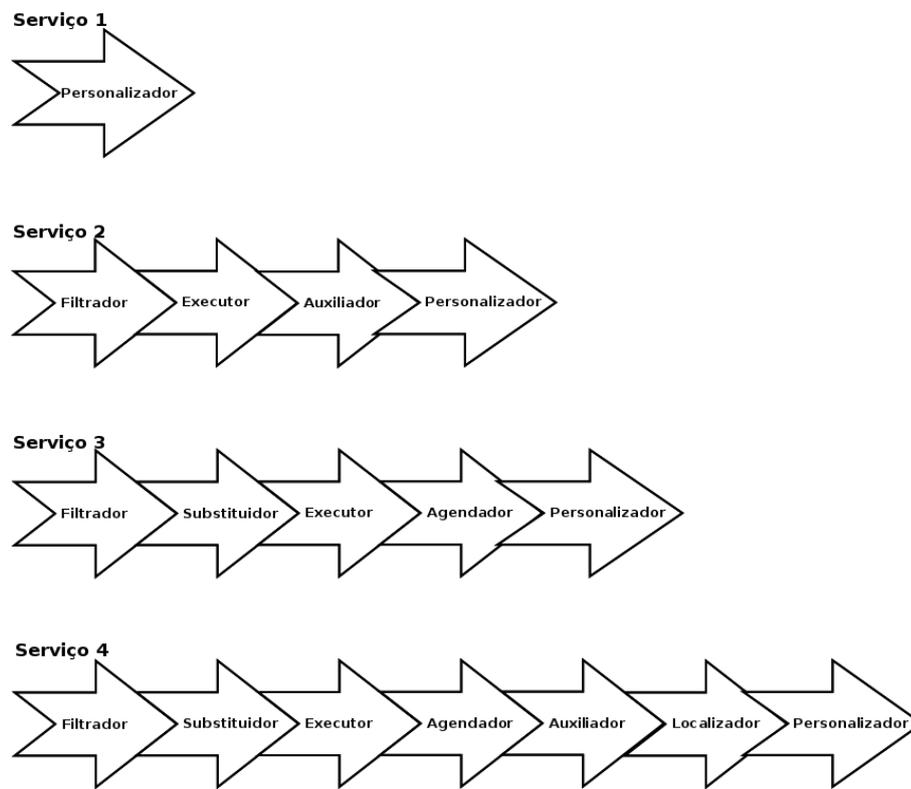
#### 6.1.4 Modelo de Interação Básico

A visão que temos é que os serviços inteligentes mais básicos que um sistema computacional pode prover podem ser classificados em uma das sete classes básicas citadas no Capítulo 3. Porém, não é preciso ficar restrito aos serviços básicos, pois podemos criar serviços mais complexos e que podem ser classificados em mais de uma classe. A criação de tais serviços seria feito então através da união de dois serviços de classes distintas porém aplicados a uma mesma tarefa. Ou seja, além dos serviços básicos que cada classe provê individualmente é possível prover serviços mais complexos combinando mais de uma classe na construção do serviço, a Figura 26 apresenta um desenho dessa ideia. Em um sistema computacional essa ideia pode ser melhor utilizada ao se criar módulos que representam serviços genéricos que possam atender a várias tarefas parecidas. No nosso caso esses módulos seriam os agentes do Sistema Multiagente, assim podemos desenvolver vários agentes, sendo cada um deles especialista em um tipo específico de serviço. Com isso, podemos utilizar os serviços de vários agentes para construir serviços ainda mais complexos.

Para um melhor entendimento do modelo de interação aqui descrito, vamos começar imaginando um sistema computacional como uma caixa preta que recebe como entrada uma variável chamada INPUT para a realização de uma tarefa. Sendo um tipo especial de INPUT as mensagens oriundas de outros ambientes (software e/ou humanos). Tal sistema, também pode procurar por informações para resolver a tarefa em repositórios de informação externos ao sistema. Por fim, o ambiente após realizar a tarefa solicitada escreve o resultado na variável OUTPUT e a envia para o usuário ou para outros sistemas, conforme mostrado na Figura 27.

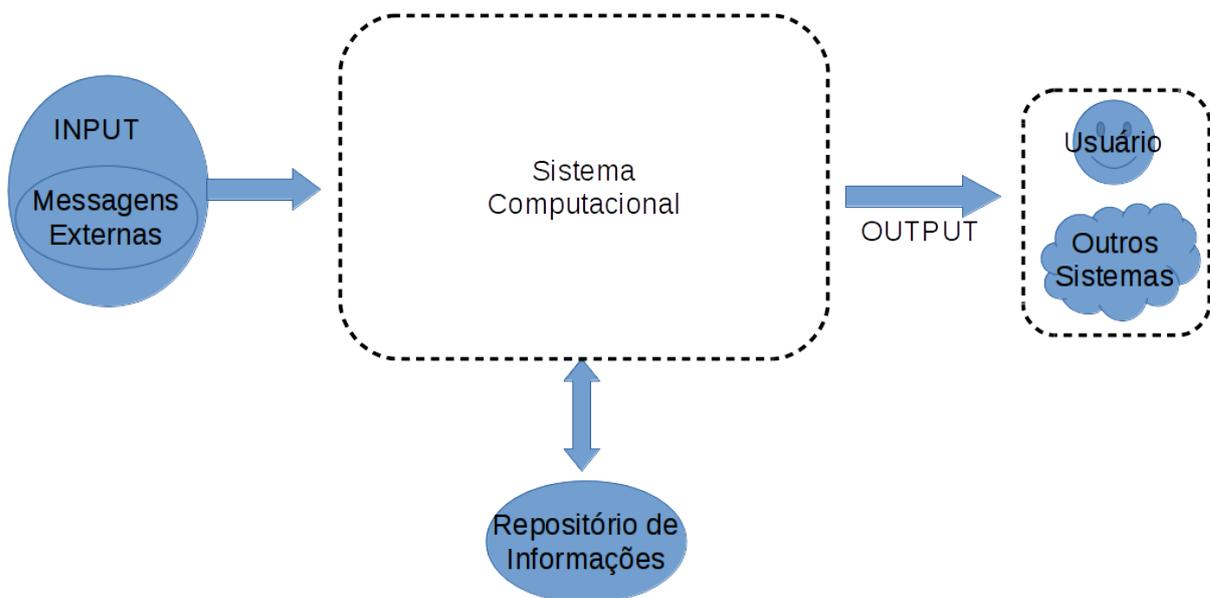
Neste modelo o INPUT pode ser uma solicitação do usuário, ou o sistema detectando sozinho uma necessidade do usuário, ou um evento que indique o início da execução de

Figura 26 – Aumento da complexidade do serviço com combinação de classes.



Fonte: Autor.

Figura 27 – Modelo de um sistema computacional como uma caixa preta.



Fonte: Autor.

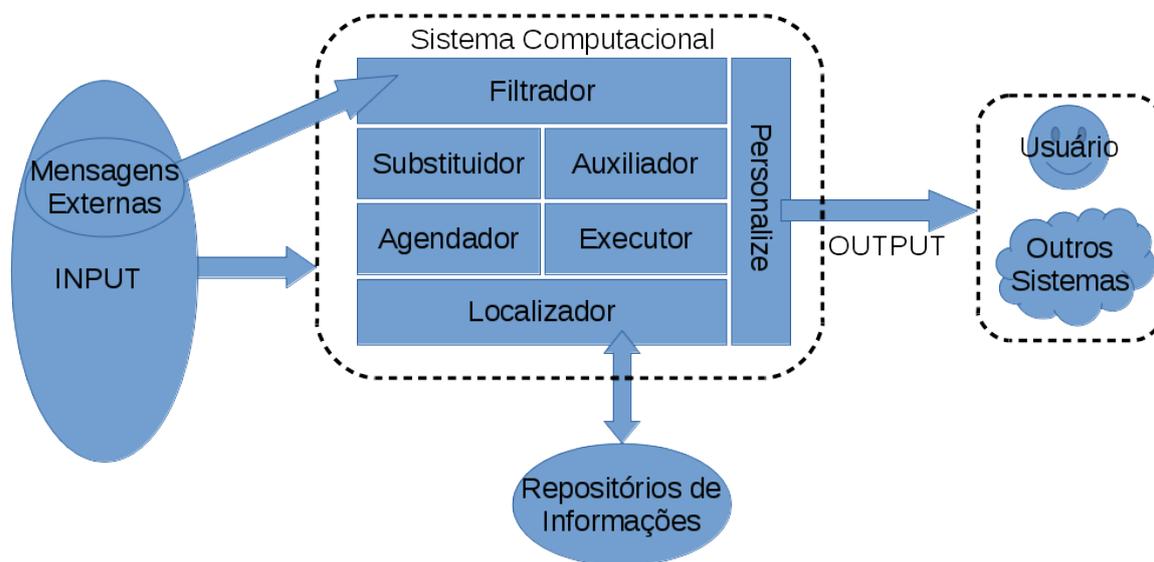
uma tarefa.

As mensagens externas são comunicações que são enviadas para o sistema com o destinatário sendo o usuário. Como exemplo podemos citar: e-mails, SMS, mensagens de voz, etc.

Os repositórios de informação são fontes de dados e informações que são externas ao sistema, mas que permitam que o sistema faça consultas e pesquisas em suas bases. Como exemplo podemos citar: máquinas de busca, ontologias disponíveis online, APIs de outros sistemas, etc.

O modelo de interação proposto neste trabalho pode ser visto como um detalhamento do modelo de funcionamento interno da caixa preta conforme mostrado na Figura 28, onde pode-se verificar que o sistema computacional recebe um INPUT, tal INPUT pode então ser processado em qualquer dos agentes do sistema (sendo que o INPUT do tipo mensagens externas vai obrigatoriamente para o agente Filtrador). Os agentes podem então realizar o seu trabalho e passar o resultado de um para o outro, mas sempre dentro do sistema computacional, ao final o resultado deve ser enviado para um agente Personalizador que vai gerar o OUTPUT que deve ser enviado para o usuário ou para os outros sistemas.

Figura 28 – Visão caixa preta do Modelo de Interação.



Fonte: Autor.

Nesse modelo temos o sistema computacional funcionando como um Sistema Multiagente e contendo os agentes responsáveis pelos tipos de tarefas inteligentes que podem ser implantadas.

Com a utilização de agentes para cada tipo espera-se poder criar e reutilizar códigos que possam ser utilizados em várias tarefas diferentes mas com o mesmo propósito, como

por exemplo:

- Filtragem de SPAM que normalmente é utilizado para filtrar e-mails mas poderia ser utilizado também para filtragem de mensagens SMS;
- Opções de agendamento que podem ser reaproveitadas por vários usuários dentro de um mesmo ambiente;
- Acesso a base de dados externas, que pode ser compartilhado por vários agentes do mesmo ambiente, assim os vários agentes não precisam reimplementar as mesmas funções.

Neste modelo, o INPUT pode ser recebido diretamente em qualquer um dos agentes o qual vai realizar algum serviço e repassar para outro agente caso o serviço precise de algum serviço fornecido por outro agente. Até que eventualmente será repassado para o agente Personalizador que é responsável pela interface com o usuário.

O caso mais simples seria quando o INPUT fosse recebido diretamente por um agente Personalizador (Ex: sistema detecta que o usuário é daltônico) e o OUTPUT já ser enviado para o usuário (Ex: alteração do esquema de cores da interface) ou para outros sistemas (Ex: um e-mail avisando sobre mudança de horário numa reunião).

Um dos casos mais complexos seria quando o INPUT fosse gerado por uma mensagem externa recebida por um agente Filtrador e a execução do serviço passa-se por todos os outros tipos de agente.

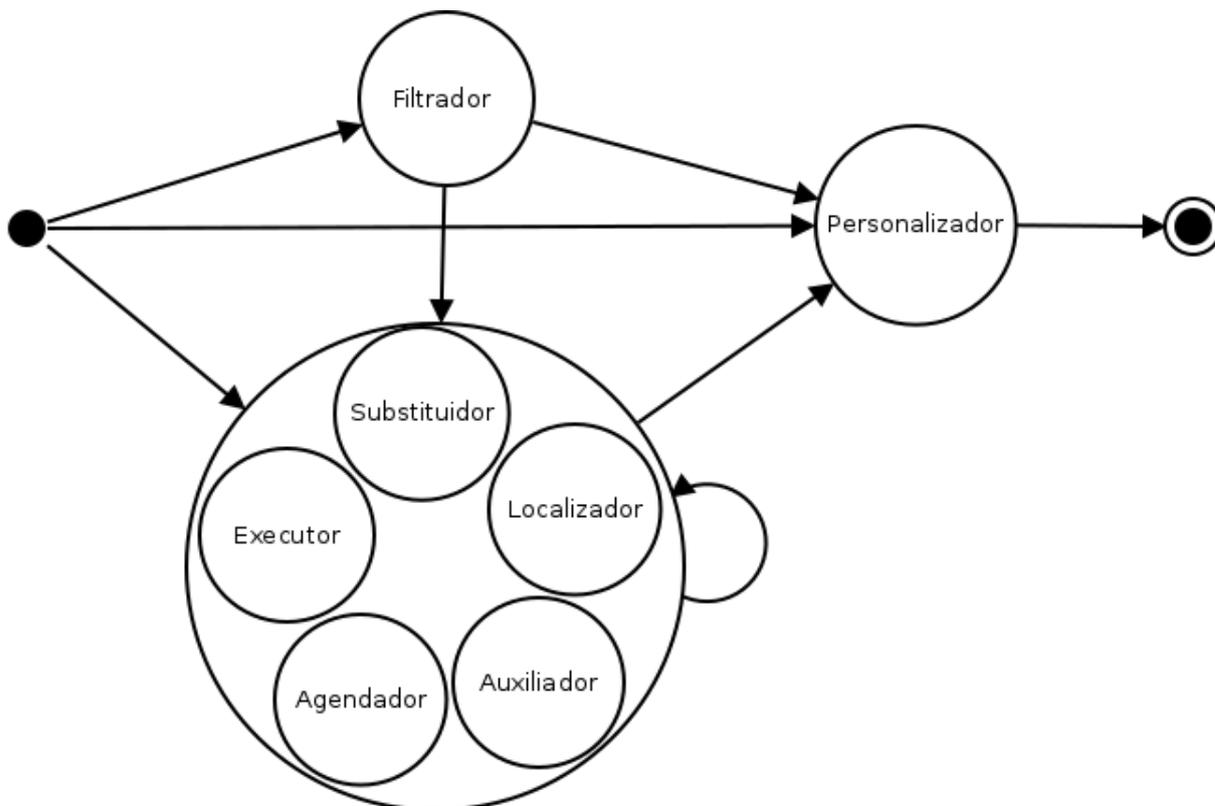
Para exemplificar melhor, o fluxo de execução do serviço pode ser representado pelo autômato mostrado na figura 29.

Dois pontos merecem destaque no autômato: O primeiro é que para terminar a execução obrigamos a passagem por um agente Personalizador. Isto porque este é o responsável por não apenas apresentar o resultado ao usuário como também adaptar o resultado as necessidades e preferências do usuário. Assim, obrigando que toda resposta antes de ser enviada para o usuário passe por um agente deste tipo garantimos que as necessidades e preferências do usuário serão sempre aplicadas quando conveniente.

O segundo ponto é que a única forma de acesso ao um agente Filtrador é quando a ação começa por ele. Isto é importante para deixar claro que esses agentes fazem a recepção das informações enviadas por ambientes externos para o usuário. Uma eventual resposta necessária deve ser repassada adiante até que eventualmente chegue a um agente Personalizador onde será tratada antes de ser enviada ao destinatário.

Assim as classes podem ser vistas como conjuntos de blocos de construção e a implementação de serviço inteligente é justamente juntar os blocos oriundos de várias

Figura 29 – Autômato representando o fluxo de execução do serviço.



Fonte: Autor.

classes para a formação de um serviço. Quanto mais blocos forem usados, mais complexo e interessante é o serviço.

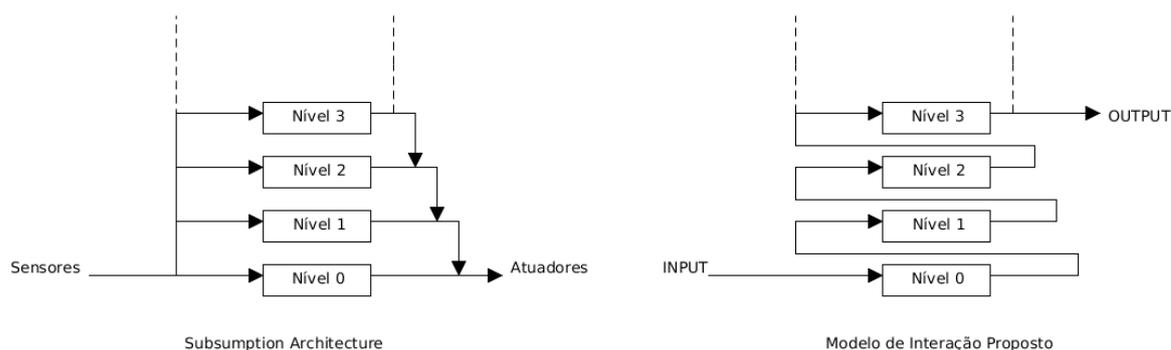
### 6.1.5 Fluxo da Execução dos Serviços

A execução do fluxo da informação do nosso Modelo de Interação (MI) foi influenciada pela *Subsumtion Architecture* (SA). No nosso caso podemos considerar que o MORFEU é equivalente ao nível 0 da SA e os níveis 1 ou maiores fazem parte da Camada de Serviços Inteligentes (CSI) pois é justamente a interação do MORFEU com o usuário que interceptamos e inserimos o serviço inteligente.

Podemos dizer que enquanto na SA cada nível recebe o mesmo INPUT e pode alterar o OUTPUT do nível abaixo do seu, na CSI esse comportamento é diferente. Na CSI cada nível (serviço) recebe como INPUT o OUTPUT do nível abaixo. Após realizar o seu trabalho cada nível então gera uma OUTPUT baseada na sua INPUT e a envia como o seu resultado. Isto está ilustrado na Figura 30 que mostra as duas situações ao mesmo tempo, possibilitando então uma comparação do funcionamento de ambas.

Outro ponto de diferença é que na SA cada nível tem a sua posição fixa. Por exemplo, o nível 2 vai continuar sendo nível 2 mesmo que mais serviços sejam acrescentados ou

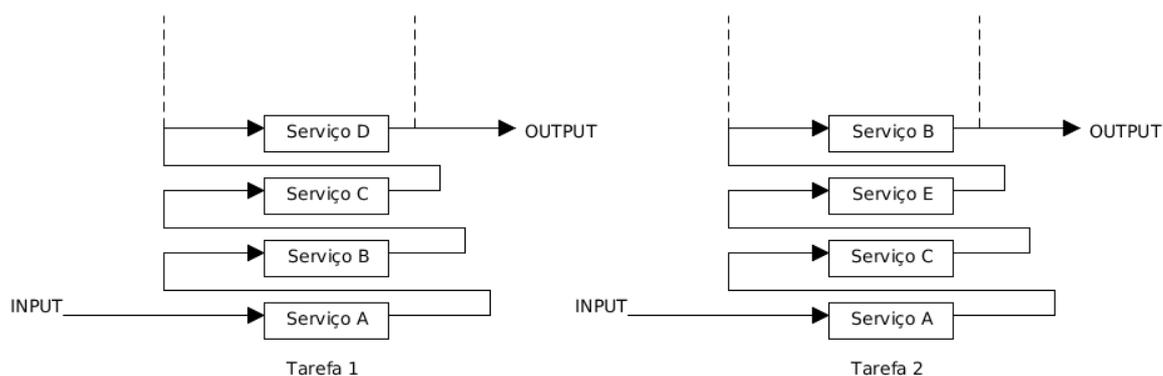
Figura 30 – Comparação Subsumption e o Modelo de Interação Proposto.



Fonte: Autor.

retirados. Enquanto que no MI proposto mudamos esse comportamento. Cada serviço da CSI é visto como um nível, porém esse nível não é fixo, e sim, depende da composição da tarefa em execução. Assim, um mesmo serviço pode ser considerado como nível 2 numa tarefa e como nível 4 em outra tarefa, conforme ilustrado na Figura 31. A única exceção é para os serviços de personalização que sempre serão os últimos a serem executados antes do sistema gerar o OUTPUT.

Figura 31 – Serviços no Modelo de Interação.



Fonte: Autor.

### 6.1.6 Código do Modelo de Interação Básico

Durante a codificação do Modelo de Interação (MI) foram feitas as mudanças com o objetivo de otimizar o processo quando da sua execução:

- Conforme mostrado no autômato da Figura 29, os tipos de agente Substituidor, Localizador, Auxiliador, Agendador e Executor podem ser agrupados em um único

tipo genérico de agente. Como no momento, não temos diferenças de prioridades ou de argumentos de entrada ou saída entre esses tipos de agentes, foi decidido criar um novo papel no MI que substitui todos esses agentes. Chamados esse novo papel de: Other;

- Para otimizar o tempo de execução dentro do OKK o MI foi alterado. A alteração consistiu em uma vez que todos os agentes fornecedores de serviços inteligentes tivessem realizado o seu registra com sucesso, a interação deveria continuar entre eles até que todo o sistema fosse desligado, a outra opção é que a interação fosse por tarefa executada e quando na próxima tarefa um outro ciclo teria de ser iniciado;
- Cada agente deve testar se ele pode executar o seu serviço dado a variável INPUT que ele recebeu. Inicialmente essa restrição ficava explícita no MI, porém para melhorar o desempenho ela foi alterada para ficar implícita. O teste é feito assim que o agente começa a executar o seu procedimento, caso possa oferecer o serviço o procedimento continuar caso contrário ele simplesmente retorna o INPUT;
- O MI descreve também que para cada tipo de agente deve ser oferecida a oportunidade de realizar um serviço no INPUT para cada representante daquele tipo de agente dentro do ambiente.

Passando então a implementação do modelo de interação apresentado na seção anterior e descrito nesta seção, temos o seguinte código com comentários em LCC:

```

// Cabecalho do MI, listando os papeis existentes
r(avatar, initial)
3 r(filter, necessary)
  r(filterMaster, necessary)
  r(other, necessary)
  r(otherMaster, necessary)
7 r(personalizer, necessary)
  r(personalizerMaster, necessary)

// Descricao do papel avatar
11 // O Avatar recebe um INPUT solicita aos agentes que fazem servicos que
    realizem
// os servicos, ao final o resultado salvo em OUTPUT e enviado para o
    usuario.
a(avatar, ID0) ::
  null <- getInput(Input) then
15 ((
    doTask(Input) => a(filterMaster, ID1) <- isMessage(Input) then
      taskDone(Output) <= a(filterMaster, ID1) then
      doTask(Output) => a(otherMaster, ID2) then
19 taskDone(Output) <= a(otherMaster, ID2) then

```

```

doTask(Output) => a(personalizerMaster, ID5) then
taskDone(Output) <= a(personalizerMaster, ID5)
) or (
23 doTask(Input) => a(otherMaster, ID2) then
taskDone(Output) <= a(otherMaster, ID2) then
doTask(Output) => a(personalizerMaster, ID5) then
taskDone(Output) <= a(personalizerMaster, ID5)
27 ) or (
doTask(Input) => a(personalizerMaster, ID5) then
taskDone(Output) <= a(personalizerMaster, ID5)
)) then
31 null <- displayOutput(Output)
// Acrescentado para que a interacao dure ate que o sistema seja
desligado
then a(avatar, ID0)

35 // Descricao do papel filterMaster
// Utilizado para solicitar recursivamente o servico para todos os
agentes do
// tipo filter
a(filterMaster, ID1) ::
39 doTask(Input) <= a(avatar, ID0) then
null <- getPeers("filter",IDs) then
a(filterMaster(Input,Output,IDs),ID6) then
taskDone(Output) => a(avatar, ID0)
43 then a(filterMaster, ID1)

a(filterMaster(Input,Output,IDs),ID6) ::
null <- IDs = []
47 or (
doTask(Input) => a(filter,ID) <- IDs = [ID|RIDs] then
taskDone(Output) <= a(filter,ID) then
a(filterMaster(Output,Output,RIDs),ID6)
51 )

// Descricao do papel filter
a(filter, ID7) ::
55 doTask(Input) <= a(avatar, ID0) then
taskDone(Output) => a(avatar, ID0) <- execTask(Input, Output)
// Acrescentado para que a interacao dure ate que o sistema seja
desligado
then a(filter, ID7)
59

// Descricao do papel otherMaster
// Utilizado para solicitar recursivamente o servico para todos os
agentes do
// tipo other

```

```

63 a(otherMaster , ID2) ::
    doTask(Input) <= a(avatar , ID0) then
    null <- getPeers("other",IDs) then
    a(otherMaster(Input ,Output ,IDs),ID3) then
67 taskDone(Output) => a(avatar , ID0)
    then a(otherMaster , ID2)

    a(otherMaster(Input ,Output ,IDs),ID3) ::
71 null <- IDs = []
    or (
        doTask(Input) => a(other ,ID) <- IDs = [ID|RIDs] then
        taskDone(Output) <= a(other ,ID) then
75 a(otherMaster(Output ,Output ,RIDs),ID3)
    )

    // Descricao do papel other
79 a(other , ID4) ::
    doTask(Input) <= a(otherMaster ,ID3) then
    taskDone(Output) => a(otherMaster ,ID3) <- execTask(Input , Output)
    // Acrescentado para que a interacao dure ate que o sistema seja
    desligado
83 then a(other , ID4)

    // Descricao do papel personalizerMaster
    // Utilizado para solicitar recursivamente o servico para todos os
    agentes do
87 // tipo personalizer
    a(personalizerMaster , ID5) ::
    doTask(Input) <= a(avatar , ID0) then
    null <- getPeers("personalizer",IDs) then
91 a(personalizerMaster(Input ,Output ,IDs),ID8) then
    taskDone(Output) => a(avatar , ID0)
    then a(personalizerMaster , ID5)

95 a(personalizerMaster(Input ,Output ,IDs),ID8) ::
    null <- IDs = []
    or (
        doTask(Input) => a(personalizer ,ID) <- IDs = [ID|RIDs] then
99 taskDone(Output) <= a(personalizer ,ID) then
        a(personalizerMaster(Output ,Output ,RIDs),ID8)
    )

103 // Descricao do papel personalizer
    a(personalizer , ID9) ::
    //doTask(Input) <= a(avatar , ID0) <- canDoTask(Input) then
    doTask(Input) <= a(avatar , ID0) then
107 taskDone(Output) => a(avatar , ID0) <- execTask(Input , Output)

```

```
// Acrescentado para que a interacao dure ate que o sistema seja
desligado
then a(personalizer, ID9)
```

### Código 2 – Código do Modelo de Interação

Um estudo e teste de validação da utilização do LCC para a construção desse modelo bem como o atendimento das necessidades do trabalho desenvolvido nesta Tese está documentado no [Apêndice C](#).

## 6.2 Fantasos - Uma Camada de Serviços Inteligentes

Após a construção, implementação e validação do Protocolo de Interação entre os agentes Avatares e os agentes responsáveis pelos serviços passamos a fase de implementação da arquitetura de referência.

Um passo preliminar foi a escolha de um nome para a camada de serviços inteligentes. Como o objetivo é o de criar serviços inteligentes para o MOrFEU, escolhemos utilizar um nome da mitologia grega que fosse próximo ao deus Morfeu, nesse caso, o nome escolhido foi Fantasos, que é um dos irmãos de Morfeu. Fantasos é deus da fantasia e dos sonhos fantasiosos.

A seguir fazemos o detalhamento do Fantasos.

### 6.2.1 Arquitetura do Fantasos

Para a construção do Fantasos foi escolhido utilizar o *framework* de desenvolvimento de sistemas multiagente JADE<sup>2</sup>. O JADE foi escolhido pelos seguintes motivos:

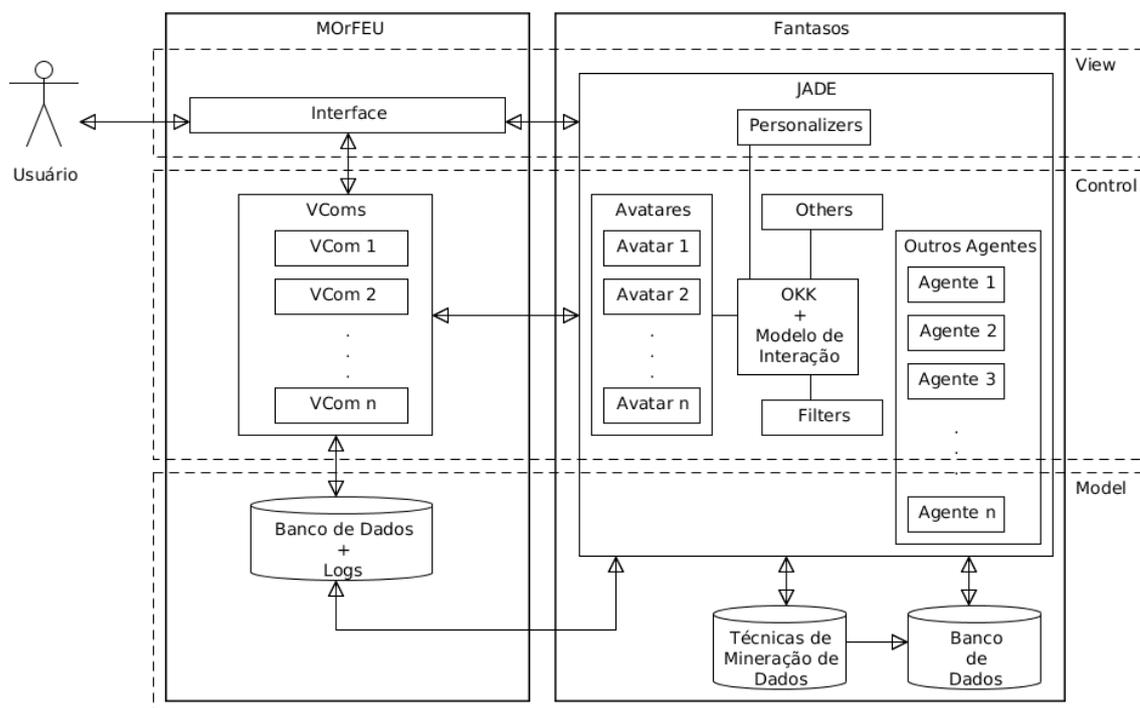
- Implementa a especificação FIPA<sup>3</sup> para sistemas multiagente;
- Foi feito em Java, uma linguagem que o autor tem familiaridade e que tem uma gama grande de bibliotecas já implementadas e disponíveis para uso;
- Mesma linguagem em que o OKK foi implementado, o que facilita a sua utilização;
- Sua licença LGPL permite a sua utilização e distribuição do código produzido sem a necessidade de despesas financeiras.

Para a construção da arquitetura de funcionamento do Fantasos seguimos a arquitetura de referência, o resultado pode ser visto na Figura 32.

<sup>2</sup> <<http://jade.tilab.com>>

<sup>3</sup> <<http://www.fipa.org/>>

Figura 32 – Arquitetura do Fantasos.



Fonte: Autor.

## 6.2.2 Ambiente das Tarefas

Seguindo a Teoria de Agentes podemos definir que o ambiente das tarefas dos agentes do Fantasos é formado pelo Ambiente Colaborativo Flexível na Web (ACFW) e pelos outros agentes do ambiente. Assim neste contexto, devemos definir quais os sensores e quais os atuadores que os agentes do Fantasos tem ao seu dispor.

### 6.2.2.1 Sensores e Atuadores

Conforme a arquitetura os agentes devem ter acesso de leitura e escrita nos seguintes elementos do ACFW MORFEU: Interface, Ferramentas, Logs e Banco de Dados.

O Banco de Dados e os Logs no caso do MORFEU estão no mesmo Sistema Gerenciador de Banco de Dados (SGBD) e portanto com os agentes tendo acesso ao SGBD temos acessos a esses dois itens.

Para as Ferramentas, o MORFEU por ser um ambiente flexível mantém as definições das Ferramentas (os VComs e suas estruturas) em banco de dados e portanto o Fantasos pode acessar as Ferramentas tanto pelo SGBD como pela Interface.

A Interface é o elemento mais difícil de o Fantasos ter o acesso de leitura e escrita, visto que a sua instanciação somente ocorre no navegador do usuário final e somente quando este está acessando o MORFEU. No entanto, a vantagem é que a instanciação da

Interface utiliza protocolos padrões conhecidos e assim é possível prever pelo menos três níveis ao interceptar a instanciação da Interface pelo Fantasos:

- **Nível 1:** No nível de aplicação. Podemos inserir código novo diretamente na aplicação para que a Interface retornada ao usuário final seja antes alterada pelo Fantasos.
- **Nível 2:** No nível do servidor web. Podemos alterar o funcionamento do servidor web de tal forma que possamos interceptar o funcionamento do protocolo HTTP e realizar as consultas ao Fantasos e alterar o resultado final antes de ele ir para o usuário.
- **Nível 3:** No nível do protocolo TCP/IP. Podemos implementar uma solução que analisa o tráfego TCP/IP entre o usuário e o servidor web. Neste caso a solução poderia abrir os pacotes TCP/IP e alterar o conteúdo dos pacotes HTTP. Soluções parecidas com essa opção já existem no mercado e são conhecidos como Firewall de Aplicação.

No nível 1 os sensores e os atuadores teriam acesso às variáveis e funções internas a aplicação e assim seria possível ao agentes criar serviços mais específicos para usuários específicos. Ou seja, temos informações específicas dos usuários de um ACFW específico.

No nível 2 os sensores e os atuadores teriam acesso a menos informações sobre os usuários finais (somente as informações disponíveis no protocolo HTTP), porém teriam acesso a mais usuários (se mais de um ACFW estiver hospedado no servidor web).

No nível 3 os sensores e os atuadores teriam acesso a mesma quantidade de informações (protocolo HTTP) porém de forma mais difícil (via o protocolo TCP/IP) no entanto a quantidade de usuários finais em potencial que ele poderia interagir é a maior dos três níveis (vários servidores web).

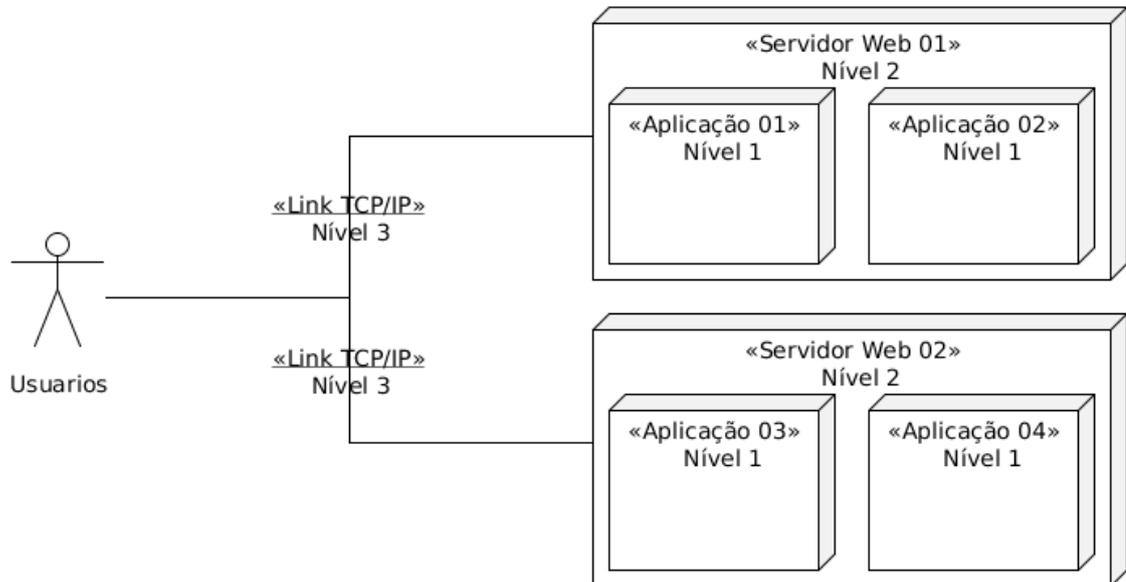
A Figura 33 ilustra os níveis de acesso a interface. Nela podemos observar que quanto menor é o nível mais específica é a aplicação e portanto menos usuários são impactados (no nível 1 somente os usuários de uma aplicação) e por outro lado quanto maior o nível, mais aplicações podem ser afetadas e portanto mais usuários podem ser impactados também.

No caso do Fantasos foi escolhido implementar apenas o nível 1 devido ao tempo e por uma implementação neste nível já servir como prova de conceito para a camada de serviços inteligentes.

#### 6.2.2.2 DataLayer

Assim, para podemos implementar os agentes do Fantasos e seguir o protocolo de interação desenvolvido para o mesmo foi criado e implementado um objeto que chamamos

Figura 33 – Níveis de acesso a interface.



Fonte: Autor.

de DataLayer. Esse objeto representa tanto o tipo do argumento de entrada como o tipo do resultado que cada agente que implementa um serviço inteligente recebe antes de executar o serviço e fornece após finalizar o serviço.

No caso do trabalho aqui desenvolvido o DataLayer é formado pelas seguintes componentes:

- **Action.** Uma string representando o argumento que o *framework* no qual o MOrFEU foi desenvolvido utiliza para saber qual ação o usuário está requisitando.
- **ActionId.** Se a ação do usuário acessa algum objeto da arquitetura do MOrFEU então esta variável vai ter um número de identificação desse objeto (id do objeto).
- **UserId.** Um número inteiro que representa o usuário que está solicitando a ação (o id do usuário).
- **PageCode.** Contém o código HTML que deve ser apresentado ao usuário
- **XPath.** Contém o XPath para ser usado nas operações XML do MOrFEU.
- **Schema.** Contém o schema para ser usado nas operações XML do MOrFEU.
- **UPI.** Contém a identificação da UPI do MOrFEU que está sendo alvo da ação.
- **specific\_version.** Contém a versão da UPI.

- **Message.** Um valor booleano indicando se o conteúdo de PageCode é uma mensagem ou não.

### 6.2.2.3 Implementação do DataLayer

A implementação do DataLayer pode ser dividida em duas partes: Uma parte fica no MOrFEU e outra parte fica no Fantasos.

No MOrFEU foi desenvolvido um módulo que altera a resposta do sistema ao usuário. Tal módulo antes de enviar a resposta para o usuário cria uma instância do DataLayer e o envia para o Fantasos e espera que o Fantasos devolva um DataLayer como resposta ou que um tempo pré-determinado passe (atualmente 3 segundos). Caso o Fantasos envie o DataLayer, o módulo recupera o valor de PageCode e o envia como resposta ao usuário. Caso o Fantasos não responda ou passe o tempo pré-estabelecido o módulo envia a resposta que seria normalmente enviada ao usuário com se não tivesse consultado o Fantasos.

No Fantasos, utilizamos o agente SocketProxyAgent (já desenvolvido e disponibilizado pelo próprio JADE) para receber o DataLayer enviado pelo módulo do MOrFEU e o enviar para um agente do tipo Avatar. O agente do tipo Avatar ao receber o DataLayer executa um ciclo de interação do MI e retorna o resultado ao agente SocketProxyAgent que o envia de volta para o módulo do MOrFEU.

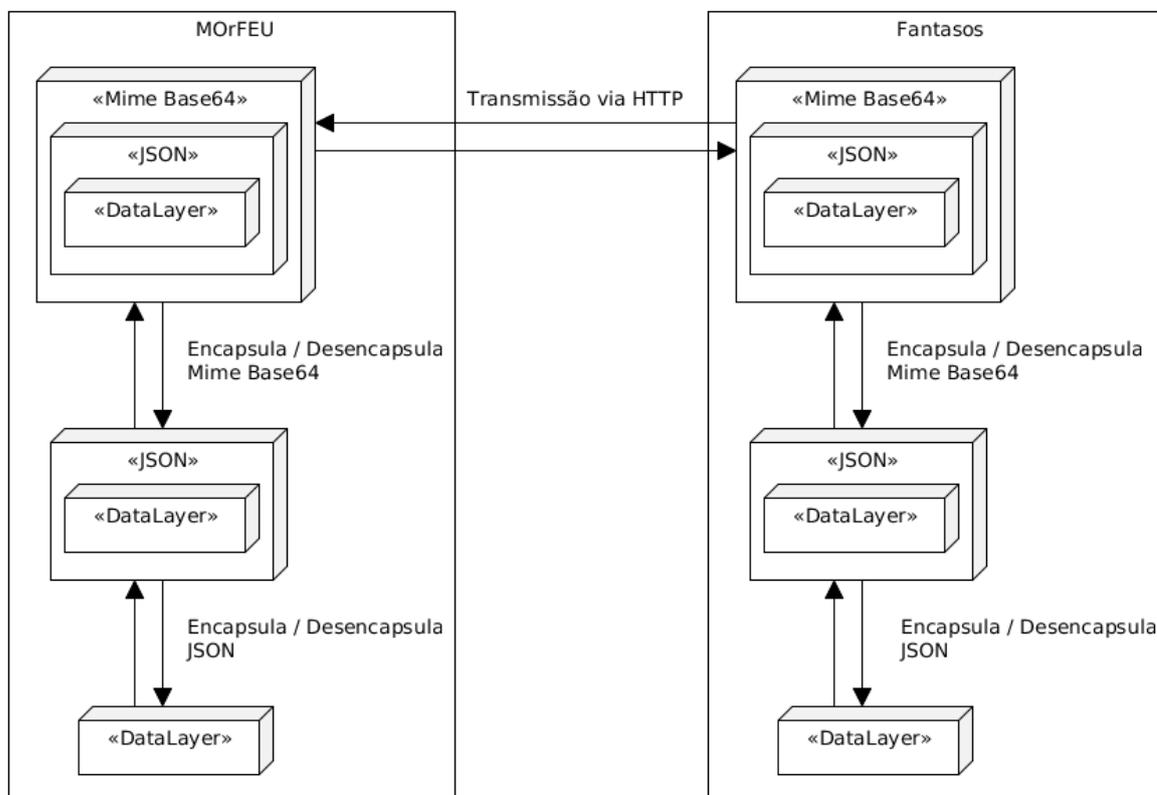
Para a transferência do DataLayer entre o MOrFEU (desenvolvido em PHP) e o Fantasos (desenvolvido em Java) e para garantir a integridade dos dados transferidos usamos o padrão JSON junto com o esquema de codificação Base64 (*Base64 encoding scheme*). Com isso conseguimos uma transferência do objeto DataLayer de forma independente da linguagem de programação e do ambiente onde será executado o código. O processo de transferência do DataLayer esta exemplificado na Figura 34, que mostra a ordem dos passos tanto no momento em que o MOrFEU envia o DataLayer para o Fantasos quanto no momento em que o Fantasos devolve o DataLayer para o MOrFEU.

### 6.2.3 Agentes Implementados

No Fantasos as seguintes classes foram implementadas em JADE e são utilizadas para instanciar agentes:

- **fantasos.agents.Avatar.** Esta é a classe que é utilizada para a instanciação de um Avatar de um usuário. Quando um novo agente Avatar é iniciado ele recebe como parâmetros o id do usuário e o nome do agente Avatar, sendo que o nome segue o padrão “UserNNN”, onde NNN é o id do usuário.

Figura 34 – Transferência do DataLayer.



Fonte: Autor.

- **fantasos.agents.Filter.** Esta classe é usada para instanciar o agente Filter. Ele é um agente do tipo Filtrador. Ele se registra como um serviço Filter no OKK. Ele foi criado para que o OKK tenha sempre um serviço Filtrador registrado e assim o OKK não precisa esperar que outro agente Filtrador se registre. O modelo de interação que estamos usando permite o uso de vários agentes do tipo Filtrador.
- **fantasos.agents.FilterMaster.** Esta classe é usada para instanciar o agente FilterMaster. Ele é usado para registrar no OKK o serviço de FilterMaster. Ele não fornece serviços inteligentes para os usuários mas ajuda na execução do Protocolo de Interação.
- **fantasos.agents.Other.** Esta classe é usada para instanciar o agente Other. Ele pode ser um agente do tipo Substituidor, Auxiliar, Agendador, Executor ou Localizador. Ele se registra como um serviço Other no OKK. Ele foi criado para que o OKK tenha sempre um serviço Other registrado e assim o OKK não precisa esperar que outro agente se registre. O modelo de interação que estamos usando permite o uso de vários agentes do tipo Substituidor, Auxiliar, Agendador, Executor ou Localizador.
- **fantasos.agents.OtherMaster.** Esta classe é usada para instanciar o agente Other-

Master. Ele é usado para registrar no OKK o serviço de OtherMaster. Ele não fornece serviços inteligentes para os usuários mas ajuda na execução do Protocolo de Interação.

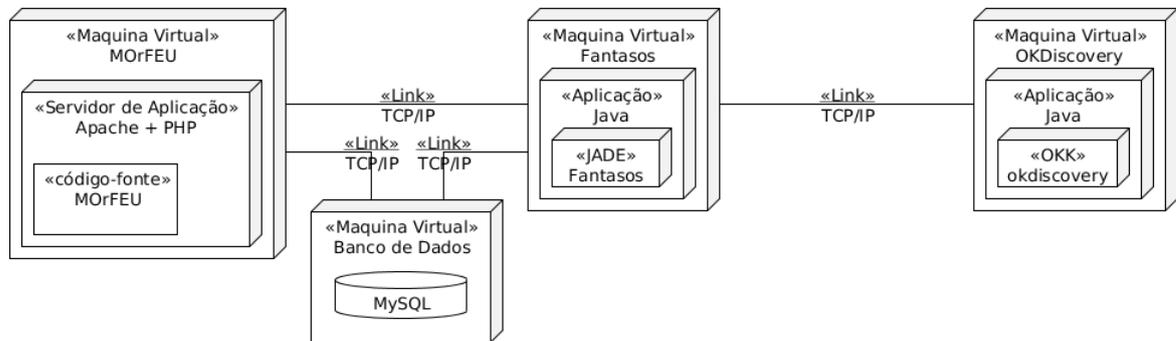
- **fantasos.agents.Personalizer.** Esta classe é usada para instanciar o agente Personalizer. Ele é um agente do tipo Personalizador. Ele se registra como um serviço Personalizer no OKK. Ele foi criado para que o OKK tenha sempre um serviço Personalizer registrado e assim o OKK não precisa esperar que outro agente Personalizer se registre. O modelo de interação que estamos usando permite o uso de vários agentes do mesmo tipo.
- **fantasos.agents.PersonalizerMaster.** Esta classe é usada para instanciar o agente PersonalizerMaster. Ele é usado para registrar no OKK o serviço de PersonalizerMaster. Ele não fornece serviços inteligentes para os usuários mas ajuda na execução do Protocolo de Interação.
- **fantasos.agents.Publisher.** Esta classe é usada para instanciar o agente Publisher. Ele é usado para registrar o nosso modelo de interação no OKK. Ele não fornece serviços inteligentes para os usuários mas ajuda na execução do Protocolo de Interação.
- **fantasos.agents.PoolPort.** Esta classe é usada para instanciar o agente PoolPort. Ele é usado pelos outros agentes que registram serviços no OKK. Ele mantém o registro de quais portas IP já foram usadas e fornece aos agentes quais portas eles devem usar. Assim ele evita que dois agentes diferentes não tentem utilizar a mesma porta IP.
- **fantasos.agents.ServiceProvider.** Esta classe é usada para instanciar um agente de qualquer dos tipos de serviço inteligente (Personalizador, Substituidor, Auxiliador, Agendador, Executor, Localizador e Filtrador). Esta classe foi feita para instanciar vários agentes, no entanto cada agente deve ter um nome único dentro do ambiente JADE. É através dessa classe que os serviços inteligentes que o Fantasos fornece podem ser expandidos.
- **fantasos.agents.Starter.** Esta classe é usada para instanciar o agente Starter. Ele é responsável por iniciar os outros agentes do Sistema Multiagente seguindo os pré-requisitos necessários entre os agentes.

#### 6.2.4 Infraestrutura do Fantasos

Na Figura 35 apresentamos a infraestrutura que foi construída para os testes e execução do Fantasos. Foram utilizados 04 máquinas virtuais assim distribuídas: 01 máquina para o MOrFEU, 01 máquina para o Fantasos, 01 máquina para o serviço

OKDiscovery do OKK e 01 máquina para o sistema de banco de dados a ser utilizado tanto pelo MOrFEU como pelo Fantasos.

Figura 35 – Infraestrutura do Fantasos.



Fonte: Autor.

Como essa infraestrutura foi distribuída usando máquinas virtuais temos a vantagem de poder realizar os testes tanto em ambientes locais como na Internet, dependendo apenas dos tipos de testes a serem realizados.

### 6.2.5 Criação de um Novo Serviço Inteligente

Para criar um novo serviço no Fantasos deve-se seguir o seguinte procedimento:

1. Criar uma classe java herdeira da classe *fantasos.services.other.IntelligentOtherService*
2. Codificar o serviço inteligente implementando as seguintes funções:
  - **boolean haveService (DataLayer data)** Esta função deve avaliar a variável data e retornar VERDADEIRO (true) caso o serviço inteligente possa ser aplicado ou FALSO (false) caso não possa ser aplicado.
  - **boolean doTask (DataLayer In, DataLayer Out)** Esta função deve realizar o serviço inteligente usando a variável In como entrada e o resultado deve ser salvo na variável Out. Caso ocorra algum erro durante a execução do serviço deve ser retornado o valor FALSO (false) ou caso tudo ocorra sem erros deve ser retornado o valor VERDADEIRO (true).
3. Instanciar um novo Agente usando a classe *ServiceProvider* e passando como o primeiro argumento do novo Agente o nome do serviço e como segundo argumento a classe Java que contém a implementação do serviço inteligente.

## 6.3 Conclusão do Capítulo

Como um primeiro passo para a construção da camada de serviços inteligentes foi necessário a especificação e implementação de um protocolo de interação. A descrição do protocolo de interação desenvolvido engloba a descrição dos dois problemas que precisam ser resolvidos: A descoberta de serviços e a composição dos serviços.

Para a implementação do protocolo de interação foram executados dois passos. No primeiro passo escolheu-se usar a linguagem LCC pois ela permite que seja feita uma especificação declarativa e também permite a execução de um modelo de interação que implemente o protocolo de interação. O segundo passo envolveu justamente a especificação do um modelo de interação básico a ser executado.

O modelo de interação criado utiliza a idéia de combinação de vários serviços de classes diferente para a criação de um serviço mais complexo. Assim, ele foi modelado como autômato que executa um fluxo de execução do serviço onde o resultado de um serviço pode ser alterado pelo serviço seguinte no fluxo.

Para a camada de serviços inteligentes implementada demos o nome de Fantasos. O Fantasos foi feito para funcionar em conjunto com o MOrFEU de forma a fornecer serviços inteligentes para os usuários do MOrFEU. Neste sentido, o Fantasos depende do funcionamento do MOrFEU para poder funcionar corretamente, enquanto que o MOrFEU pode funcionar sem o Fantasos, mas neste caso ele perderias as funcionalidades do Fantasos e somente poderia fornecer as suas funcionalidades básicas para os seus usuários.

O principal desafio na construção do Fantasos foi o de encontrar uma forma de os agentes terem sensores com acesso a interface do usuário. Para a resolução desse problema foi verificado que o Fantasos poderia criar sensores em três níveis diferentes de acesso: nível de aplicação, nível de servidor web e nível do protocolo TCP/IP. Para demonstrar a factabilidade do Fantasos optou-se pela implementação dos sensores apenas no nível de aplicação.

Outro desafio interessante que ocorreu foi o da transferência da entidade DataLayer (usada nos resultados parciais das várias tarefas executadas no fluxo de execução do serviço) entre o MOrFEU e o Fantasos, visto um ser desenvolvido em PHP e o outro ser desenvolvido em Java. Tal desafio foi superado com uso das tecnologias JSON e codificação Mime Base64.

Por fim, foi apresentado a lista de agentes implementados no MOrFEU, a infraestrutura criada para os testes do MOrFEU com o Fantasos e o procedimento de criação de novos serviços inteligentes dentro do Fantasos.

## 7 Possibilidades de Utilização

Neste capítulo exploramos a utilização do Fantasos para criação de serviços inteligentes. Discutimos sobre o uso das técnicas de aprendizagem de máquina para a criação de serviços inteligentes no contexto definido neste trabalho. Por fim vamos discutir como podemos implementar os serviços inteligentes dentro do Fantasos.

### 7.1 Criando Serviços Inteligentes

Conforme mostrado na revisão sistemática ([Capítulo 3](#)) a maioria (83,78%) dos serviços inteligentes estão focados em auxiliar o usuário a compreender quais tarefas precisam ser executadas e como executá-las. Dentro deste tipo de serviço podemos dividir a forma como a ajuda é feita em 3 tipos: via um sistema de recomendação (13,89%), via um sistema de suporte à decisão (13,89%) e via um sistema especialista (72,22%).

#### 7.1.1 Sistemas de Recomendação

Sistemas de recomendação combinam várias técnicas de computação para selecionar itens e recomendá-los aos usuários do sistema com base nas características do usuário e os contextos que podem ser aplicados ao usuário.

“Sistemas de Recomendação são primariamente direcionados para pessoas que não tem experiência pessoal ou competência suficientes para avaliar uma lista de itens potencialmente sobrecarregada que um sítio web, por exemplo, pode oferecer” ([RICCI; ROKACH; SHAPIRA, 2011](#)).

Neste sentido, um serviço que oferece uma lista pequena de boas recomendações seria considerado um serviço inteligente. É o que faz por exemplo a NetFlix ao mostrar sugestões de filmes para os seus usuários, ela não mostra toda a lista de filmes disponíveis mas apenas aqueles com mais chances de agradarem o usuário.

De forma similar, podemos criar agentes para o Fantasos que funcionem como sistemas de recomendação para os usuários.

#### 7.1.2 Sistemas de Suporte à Decisão

Sistemas de suporte à decisão são sistemas que normalmente incluem técnicas de sistemas baseados em conhecimento para auxiliar o usuário durante um processo de decisão, geralmente feitos para serem usados pelo pessoal do nível gerencial de uma organização.

Segundo (SPRAGUE, 1980) os sistemas de suporte à decisão tem as seguintes características:

- Normalmente são construídos para resolverem problemas menos estruturados;
- Combinam o uso de técnicas analíticas ou modelos analíticos com os acessos normais aos dados;
- Tem um foco especial em características que as tornam fáceis de usar e em um modo interativo;
- Enfatizam flexibilidade e adaptabilidade para tratar as mudanças no ambiente.

Assim, percebemos que sistemas de suporte à decisão podem ser utilizados para criar serviços inteligentes.

### 7.1.3 Sistemas Especialistas

Sistemas Especialistas são sistemas que simulam as ações e técnicas de um especialista em um certo domínio. Seguindo as ideias apresentadas neste trabalho, percebemos que qualquer grupo de agentes que funcione como um sistema especialista de um domínio específico vai estar fornecendo um serviço inteligente para os seus usuários.

### 7.1.4 Técnicas de Aprendizagem de Máquina

Sistemas de recomendação, sistema de suporte à decisão e sistemas especialistas podem ser desenvolvidos com o uso de uma ou várias técnicas de aprendizagem de máquina. Como exemplo de técnicas de aprendizagem de máquina podemos citar:

- Algoritmos Genéticos;
- Aprendizagem por Dicionário Esperso;
- Aprendizagem por Reforço;
- Aprendizagem por Representação;
- Aprendizagem por Similaridade e Métrica;
- Árvores de Decisão;
- Clustering (Particionamento);
- Deep Learning (Aprendizado Profundo);
- Lógica de Programação Indutiva;

- Máquinas de Vetores de Suporte;
- Rede Neural Artificial;
- Redes Bayesianas;
- Regras de Associação.

Com o objetivo de verificar quais os possíveis tipos de serviços inteligentes que poderiam ser fornecidos em um ACFW, foram desenvolvidos subprojetos de pesquisa integrados ao escopo deste trabalho, utilizando técnicas de mineração de dados. Um dos projetos foi resumido no Cenário Dois do [Capítulo 5](#). Outro projeto foi o de criação de uma infraestrutura de coleta de dados para a mineração de dados no projeto UAB.

## 7.2 Projeto de Mineração de Dados UAB

O objetivo do projeto de mineração de dados UAB é o de coletar a massa de dados educacionais que estão espalhados pelos vários ambientes educacionais e fazem parte do sistema Universidade Aberta do Brasil, coordenado pela CAPES/MEC. Um resumo dos resultados obtidos nesse projeto podem ser lidos no [Apêndice D](#).

No projeto piloto foram coletadas informações e aplicadas técnicas de mineração de dados para encontrar regras de associação. Os dados coletados foram do Centro de Educação a Distância (CED) da UFAM e foram encontradas regras de associação interessantes, que poderiam ser utilizadas, por exemplo, para implementar Agentes que forneçam serviços inteligentes de recomendação para usuários professores em um ambiente educacional.

Na seção seguinte, mostramos como tais serviços podem ser implementados no Fantosos.

### 7.2.1 Implementação no Fantosos

Para a implementação do serviço de recomendação baseado em mineração de regras de associação temos pelo menos duas formas de implementação no Fantosos:

- Na primeira forma, podemos implementar as regras diretamente dentro de um agente, como conhecimento explícito desse agente. Ou seja, o processo de mineração seria externo ao Fantosos, mas as regras encontradas seriam incorporadas nele na forma de um ou mais agentes;
- Na segunda forma, o processo de mineração também seria incorporado ao Fantosos.

A segunda forma, por ser mais interessante, é a que vamos descrever em mais detalhes. Para isso, descrevemos os agentes necessários na [Tabela 11](#).

Tabela 11 – Agentes e descrição

Agente	Descrição
ColetaDados	Responsável por coletar os itens que serão processados na mineração de dados para construir as regras de associação. A lista dos dados e como eles podem ser gerados são recuperados da base de conhecimento (pode estar em forma de arquivo ou em um banco de dados), assim se a lista for atualizada o próximo ciclo de coleta dos dados já é ajustado de acordo. Os dados coletados também serão armazenados na base de dados.
GerarRegras	Responsável por utilizar uma técnica de mineração de dados para gerar as regras de associação. Ele utiliza os dados coletados que estão armazenados na base de dados. As regras de associação geradas são então armazenadas também na base de conhecimento.
ChecaRegra	Responsável por verificar para cada usuário quais regras de associação são aplicadas a ele. Para cada usuário que alguma regra resultar em um retorno positivo, ele verifica para qual professor ele deve enviar um aviso. A mensagem de aviso e o professor que deve receber é então enviada para o Agente AvisaUsuário.
AvisaUsuário	É um agente do tipo Personalizador que vai inserir na interface do usuário um aviso. Os avisos e qual usuário devem recebe-los são enviados para este agente por outros agentes. Ele guarda internamente a lista de avisos que ainda não foram enviados e os envia oportunamente.

Fonte: Autor.

## 7.3 Um Exemplo no MOrFEU

Para exemplificar um possível uso do Fantasos para a criação de serviços inteligentes vamos utilizar o método pedagógico Controvérsia Acadêmica. Ela foi escolhida pois é uma das ferramentas que teve descrita sua implementação e utilização no MOrFEU (SANTOS, 2013). Portanto, ao mostramos como serviços inteligentes poderiam ser providos para a Controvérsia Acadêmica demonstramos também a factabilidade do Fantasos.

### 7.3.1 Controvérsia Acadêmica

Segundo (MENDONÇA et al., 2003) a Controvérsia Acadêmica é um método de aprendizagem colaborativa que utiliza a incompatibilidade entre duas ideias, informações, conclusões, teorias ou opiniões de dois alunos diferentes para fazer com que ambos cheguem a um consenso.

Para tanto, as atividades da Controvérsia Acadêmica são divididas nas seguintes fases:

- **Atividades Pré-Instrucionais.** A turma recebe uma bibliografia básica sobre o tema a ser estudado e uma descrição das atividades e habilidades sociais a serem desenvolvidas. A turma é dividida em grupos de quatro alunos e cada grupo é subdividido em duas duplas de alunos. Para cada dupla de cada grupo é designada uma posição (contra ou a favor) sobre o tema.
- **Pesquisa dos Argumentos para Apoiar a Posição Designada.** Cada dupla pesquisa informações, fatos, experiências e outras evidências relevantes no apoio da

posição designada. Duplas que estão na mesma posição designada podem conversar e trocar informações entre si.

- **Advogar Posição.** Dentro de cada grupo, cada par faz uma apresentação para o par oposto. Cada par ao realizar a sua apresentação deve fazer o melhor que a sua habilidade permita, para apresentar os argumentos que sustentam a sua posição designada e ao ouvirem a apresentação do par oposto devem prestar atenção nos dados e na lógica sobre o qual se baseam os argumentos apresentados pela outra dupla. Ao final, uma discussão aberta é feita onde cada par deve defender a sua opinião designada enquanto ataca a opinião contrária.
- **Inversão de Perspectivas.** Nesta fase as posições designadas de cada dupla são invertidas. Cada dupla deve então realizar uma nova pesquisa sobre a sua nova posição designada, podendo recorrer as anotações e discussões já realizadas no passo anterior para conseguir os melhores argumentos para a sua nova posição. No final desta fase, as duplas devem novamente realizar uma discussão aberta com a dupla oposta.
- **Síntese e Integração das Melhores Evidências e Raciocínio em uma Única Posição.** Nesta fase as duplas são desfeitas e cada grupo deve desenvolver uma síntese das diferentes idéias e fatos e adotar uma posição única na qual todos os membros do grupo possam concordar e comprometer-se. A síntese deve então ser apresentada em um relatório e uma apresentação oral do relatório deve ser feita pelo grupo para a turma.

### 7.3.2 Exemplos de serviços inteligentes

Utilizando as idéias apresentadas e discutidas na [subseção 6.1.4](#) podemos elencar os seguintes serviços para cada fase da Controvérsia Acadêmica:

1. Atividades Pré-Instrucionais;
  - a) Alunos receberem sugestões extras de bibliografia (Localizador);
  - b) Professor recebe sugestões de grupos de alunos (Auxiliador).
2. Pesquisa dos Argumentos para Apoiar a Posição Designada;
  - a) Alunos recebem sugestões extras de bibliografia (Localizador);
  - b) Professor recebe lista de alunos/duplas com dificuldades (Auxiliador);
  - c) Mensagens em nome do Professor são postadas nos fóruns (Executor).
3. Advogar Posição;

- a) Professor recebe lista de alunos/duplas com dificuldades (Auxiliador);
  - b) Mensagens em nome do Professor são postadas nos fóruns (Executor).
4. Inversão de Perspectivas;
- a) Alunos recebem sugestões extras de bibliografia (Localizador);
  - b) Mensagens em nome do Professor são postadas nos fóruns (Executor).
5. Síntese e Integração das Melhores Evidências e Raciocínio em uma Única Posição.
- a) Lembretes para os alunos sobre a data de entrega (Agendador).

Avaliando a lista de serviços e agrupando os serviços semelhantes das diferentes fases chegamos a seguinte lista de agentes para implementar esses serviços:

- **SugereBibliografia.** Responsável por procurar em repositórios externos referências e material de leitura que será indicada aos alunos. Com o passar do tempo e com as informações discutidas pelos alunos nos fóruns melhores fontes de referência podem ser encontradas.
- **SugereGrupos.** Apresenta ao Professor sugestões para as divisões dos grupos de alunos. Informações sobre os alunos podem ser utilizadas (recuperadas através dos agentes avatares) para tentar deixar o grupo o mais heterogêneo possível o que deve incentivar as discussões dentro do grupo.
- **PostProfessor.** Monitora os fóruns e posta mensagens, em nome do Professor, nos fóruns com o objetivo de incentivar as discussões.
- **LembreteAluno.** Monitora os grupos que ainda não entregaram o relatório e avisa, sempre que for oportuno, sobre o prazo para entregar o relatório.

### 7.3.3 Implementação

Para uma explicação de como implementar os agentes listados na seção anterior no Fantoso vamos descrever os algoritmos em português. Deve-se notar que esses agentes vão participar de interações de acordo com o Modelo de Interação conforme construído na [subseção 6.1.4](#). Porém, não é necessário que eles executem todas as ações da tarefa, as ações podem ser executadas por vários agentes, portanto nos algoritmos que vamos mostrar nesta seção utilizaremos chamadas para outros agentes hipotéticos implementados no ambiente.

## 7.3.3.1 Agente SugereBibliografia

```

funcao haveService (data : DataLayer) : logico
inicio
3   se upi_controversia_academica(data)
      E (controversia_academica_fase(data) = (1,2 ou 4)) entao
      retorna VERDADEIRO
      senao
7     retorna FALSO
      fimse
fimfuncao

11 funcao doTask (In : DataLayer, Out : DataLayer) : logico
    inicio
        sugestoes <- recupera_agente_sugestoes_salvas(In)
        insere_sugestoes(In,sugestoes)
15   Out <- In
        retorna VERDADEIRO
    fimfuncao

```

Código 3 – Código Agente SugereBibliografia

Em [Código 3](#) apresentamos o código em português das duas funções que o agente SugereBibliografia deve implementar no Fantasos. Abaixo descrevemos as funções que foram utilizadas dentro deste código:

- **upi\_controversia\_academica()** verifica se a UPI pertence ao tipo Controversia Acadêmica.
- **controversia\_academica\_fase()** retorna um número inteiro indicando em qual fase está a controvérsia acadêmica, começando no 1 para representar a fase de Atividades Pré-institucionais e terminando no 5 para a fase de Síntese e Integração.
- **recupera\_agente\_sugestoes\_salvas()** consulta um agente para recuperar sugestões que o agente tenha para o usuário. Esse agente faz e armazena essas sugestões ficando de prontidão para responder as consultas e atualiza as sugestões salvas sempre que ele achar conveniente.
- **insere\_sugestoes()** procedimento para inserir as sugestões dentro do objeto DataLayer para que seja utilizado por um agente Personalizado antes de ser enviado aos usuário final.

## 7.3.3.2 Agente SugereGrupos

```

funcao haveService (data : DataLayer) : logico
inicio

```

```

3  se upi_controversia_academica(data)
    E (controversia_academica_fase(data) = 1)
    E controversia_academica_professor(data) entao
    retorna VERDADEIRO
7  senao
    retorna FALSO
    fimse
fimfuncao
11
funcao doTask (In : DataLayer, Out : DataLayer) : logico
inicio
    sugestoes <- recupera_agente_grupo_sugestoes_salvas(In)
15 insere_sugestoes(In,sugestoes)
    Out <- In
    retorna VERDADEIRO
fimfuncao

```

Código 4 – Código Agente SugereGrupos

Em [Código 4](#) apresentamos o código em portugol das duas funções que o agente SugereGrupos deve implementar no Fantasos. Abaixo descrevemos as funções que foram utilizadas dentro deste código:

- **upi\_controversia\_academica()** verifica se a UPI pertence ao tipo Controversia Acadêmica.
- **controversia\_academica\_fase()** retorna um número inteiro indicando em qual fase esta a controvérsia acadêmica, começando no 1 para representar a fase de Atividades Pré-institucionais e terminando no 5 para a fase de Síntese e Integração.
- **upi\_controversia\_academica\_professor()** verifica se o usuário no contexto da UPI é um professor. Informações de usuário e UPI estão registrados no DataLayer.
- **recupera\_agente\_grupo\_sugestoes\_salvas()** consulta um agente para recuperar sugestões que o agente tenha para o usuário. Esse agente faz e armazena essas sugestões ficando de prontidão para responder as consultas e atualiza as sugestões salvas sempre que ele achar conveniente.
- **insere\_sugestoes()** procedimento para inserir as sugestões dentro do objeto DataLayer para que seja utilizado por um agente Personalizado antes de ser enviado aos usuário final.

### 7.3.3.3 Agente PostProfessor

```

2 funcao haveService (data : DataLayer) : logico

```

```
início
  se upi_controversia_academica(data)
    E controversia_academica_professor(data)
6    E (controversia_academica_fase(data) = (2,3 ou 4))
    E agente_tem_post(data) entao
    retorna VERDADEIRO
  senao
10   retorna FALSO
  fimse
fimfuncao

14 funcao doTask (In : DataLayer, Out : DataLayer) : logico
  início
    post <- recupera_agente_post(In)
    publica_postagem(In,post)
18   Out <- In
    retorna VERDADEIRO
  fimfuncao
```

Código 5 – Código Agente PostProfessor

Em [Código 5](#) apresentamos o código em português das duas funções que o agente PostProfessor deve implementar no Fantoso. Abaixo descrevemos as funções que foram utilizadas dentro deste código:

- **upi\_controversia\_academica()** verifica se a UPI pertence ao tipo Controversia Acadêmica.
- **upi\_controversia\_academica\_professor()** verifica se o usuário no contexto da UPI é um professor. Informações de usuário e UPI estão registrados no DataLayer.
- **controversia\_academica\_fase()** retorna um número inteiro indicando em qual fase esta a controvérsia acadêmica, começando no 1 para representar a fase de Atividades Pré-institucionais e terminando no 5 para a fase de Síntese e Integração.
- **agente\_tem\_post()** verifica com um agente se tem algum post que deva ser criado em nome do professor, se tiver retorna VERDADEIRO senão retorna FALSO.
- **recupera\_agente\_post()** consulta um agente para recuperar as postagens que o agente tenha para um usuário. Esse agente faz e armazena essas postagens, bem como onde elas devem ser postadas e fica de prontidão para responder as consultas.
- **publica\_postagem()** procedimento para publicar as postagens, informações de contexto e do usuário podem ser recuperadas do DataLayer.

## 7.3.3.4 Agente LembreteAluno

```
funcao haveService (data : DataLayer) : logico
inicio
  se upi_controversia_academica(data)
4     E controversia_academica_aluno(data)
     E (controversia_academica_fase(data) = 5)
     E agente_tem lembrete(data) entao
     retorna VERDADEIRO
8  senao
     retorna FALSO
  fimse
fimfuncao
12
funcao doTask (In : DataLayer, Out : DataLayer) : logico
inicio
  aviso <- recupera_agente_lembrete(In)
16 publica_lembrete(In,aviso)
  Out <- In
  retorna VERDADEIRO
fimfuncao
```

Código 6 – Código Agente LembreteAluno

Em [Código 6](#) apresentamos o código em português das duas funções que o agente LembreteAluno deve implementar no Fantaso. Abaixo descrevemos as funções que foram utilizadas dentro deste código:

- **upi\_controversia\_academica()** verifica se a UPI pertence ao tipo Controversia Acadêmica.
- **upi\_controversia\_academica\_aluno()** verifica se o usuário no contexto da UPI é um aluno. Informações de usuário e UPI estão registrados no DataLayer.
- **controversia\_academica\_fase()** retorna um número inteiro indicando em qual fase esta a controvérsia acadêmica, começando no 1 para representar a fase de Atividades Pré-institucionais e terminando no 5 para a fase de Síntese e Integração.
- **agente\_tem\_lembrete()** verifica com um agente se tem algum lembrete deva ser mostrado ao usuário, se tiver retorna VERDADEIRO senão retorna FALSO.
- **recupera\_agente\_lembrete()** consulta um agente para recuperar os lembretes que o agente tenha para um usuário. Esse agente faz e armazena esses lembretes e fica de prontidão para responder as consultas.
- **publica\_lembrete()** procedimento para publicar os lembretes, informações de contexto e do usuário podem ser recuperadas do DataLayer.

## 7.4 Conclusão do Capítulo

Neste capítulo foi discutido como técnicas de aprendizagem de máquina podem ser utilizadas para gerarem serviços inteligentes em um ACFW. Sendo que a forma mais escolhida para a implementação deste serviço é na forma de um auxílio ao usuário a realizar uma tarefa que ele já faz no ambiente. Ou seja, tenta-se reduzir o esforço cognitivo do usuário ao usar o sistema, quanto menos esforço mais inteligente é o sistema.

Tradicionalmente, tal auxílio se apresenta em três formas distintas: Sistemas de recomendação, Sistemas de Suporte à Decisão, e por último, Sistemas Especialistas. Tais sistemas, por sua vez, podem ser desenvolvidos através do uso de uma ou da combinação de várias Técnicas de Aprendizagem de Máquina.

Para exemplificar os tipos de serviços inteligentes que poderiam ser desenvolvidos e como eles poderiam ser implementados no Fantoso citamos os trabalhos realizados no projeto de mineração de dados UAB e os possíveis exemplos de serviços inteligentes para o método de aprendizagem colaborativo chamado Controvérsia Acadêmica. Além da especificação também foram descritas as possíveis implementações para ambos os casos.

## 8 Conclusão

Apresentamos a seguir os resultados obtidos até o momento, os trabalhos em andamento e seus possíveis desdobramentos, trabalhos futuros bem como considerações finais sobre o projeto. Também é comentado os resultados alcançados e suas possíveis vantagens.

### 8.1 Resultados Obtidos

Como resultados alcançados podemos destacar: Um levantamento (através de revisão sistemática) sobre construção de sistemas inteligentes em ambientes colaborativos; uma classificação funcional para serviços inteligentes; a concepção de um modelo de referência; e a definição de uma arquitetura de referência para construção de uma camada de serviços inteligentes; além de relatos de resultados parciais descrevendo artefatos concretos como o Fantasos e os recursos a ele relacionados.

#### 8.1.1 Resultados da Revisão Sistemática

A revisão sistemática foi realizada com objetivo de se verificar como são feitos atualmente os serviços inteligentes em ambientes colaborativos, e resultou no levantamento das técnicas de Inteligência Artificial utilizadas com mais frequência na criação de serviços inteligentes em ambientes colaborativos na web. Também foi percebido que a maioria dos serviços inteligentes encontrados tem como foco o auxílio ao usuário com tarefas específicas do domínio do sistema. Serviços para áreas correlatas a execução do trabalho ou mais genéricas (por exemplo: gerenciamento de compromissos e prazos ou procura e filtragem de informações) foram menos frequentes.

Ainda como resultado da revisão sistemática, podemos apontar uma necessidade de mais estudos sejam realizados com o objetivo de fomentar a criação de serviços em ambientes colaborativos para usuários com necessidades especiais. Apesar de também podermos ter serviços inteligentes nesta área eles não foram pesquisados devido ao escopo definido para a atuação desse projeto pois tais pesquisas necessitariam de uma melhor pesquisa e entendimento sobre as interfaces e dispositivos atualmente em uso.

Definimos uma classificação para serviços inteligentes baseada nas funcionalidades oferecidas ao usuário pelos sistemas para a execução do trabalho do usuário dentro do ambiente (Agendador, Auxiliador, Executor, Filtrador, Localizador, Personalizador e Substituidor). Tal classificação ajudou nas etapas seguintes do processo de pesquisa. Para verificar se a classificação de serviços inteligentes utilizada apresentava um mínimo de

evidência de ser passível de uso foi feita uma comparação de listas de tarefas realizadas por um Humano Assistente Virtual com as possíveis tarefas realizadas por um Software Assistente Virtual.

A classificação para serviços inteligentes definida neste trabalho pode ainda ser utilizada como uma método para a criação de serviços inteligentes. Pois, sabendo os tipos de serviços inteligentes possíveis de serem criados podemos melhorar uma tarefa que esteja sendo executada em um ambiente computacional. Por exemplo, caso ao analisar uma determinada tarefa podemos encontrar meios de automatizá-la, caso a automação seja completa teremos criados um novo serviço do tipo Substituidor, caso seja parcial a automação podemos ter criado um novo serviço do tipo Executor (caso a tarefa possa ser realizada diretamente pelo ambiente) ou do tipo Auxiliador (caso possamos apenas automatizar as tarefas secundárias a tarefa central).

### 8.1.2 Resultados do Modelo de Referência

A definição de um modelo de referência para a solução do problema guiou a construção de uma arquitetura que constituiu o *framework* buscado, o qual teve uma instância implementada. O modelo ajuda a identificar em nível macro os principais conceitos e relacionamentos envolvidos na solução do problema. Além disso, o modelo pode ser utilizado para a instanciação de várias outras arquiteturas de referência diferentes da que foi instanciada neste trabalho.

O modelo de referência pode ainda ser utilizado em futuras discussões tanto para explicar e descrever o domínio do problema como também para realizar comparações entre diferentes soluções do problema.

Por fim, o modelo de referência pode ser utilizado também para o estudo e o desenvolvimento de várias outras arquiteturas de referência diferentes da que foi construída neste projeto.

### 8.1.3 Resultados da Arquitetura de Referência

A definição de uma arquitetura para a camada de serviços inteligentes auxiliou ao montar os conceitos e relacionamentos em componentes de um sistema computacional definindo também um estilo arquitetural a ser utilizado durante as fases seguintes do projeto.

A arquitetura de referência proporcionou ainda a implementação de uma camada de serviços inteligentes para o MOrFEU na forma do Fantasos. Além disso, o Fantasos por ser um framework no sentido de Engenharia de Software pode ser utilizado para expandir os serviços inteligentes atualmente projetados ou desenvolvidos para o MOrFEU. Assim,

novos serviços podem ser continuamente integrados ao ambiente do MOrFEU de forma transparente para o usuário.

#### 8.1.4 Resultados Parciais

Os artigos (NETO; CASTRO, 2015) e (NETO; CASTRO, 2017) foram publicados no escopo do projeto. Sendo o primeiro um relato do resultado de uma pesquisa com a experimentação de técnicas de Aprendizagem de Máquina com o objetivo de auxiliar na criação da camada de serviços inteligentes para o MorFEU. No artigo é apresentado uma ferramenta para auxiliar na resolução do problema da desistência por parte dos alunos em cursos online à distância ou semipresenciais. Enquanto o segundo foi resultado do projeto de mineração de dados educacionais UAB e resultou em uma proposta de uma arquitetura de referência para um ETL distribuído.

## 8.2 Desdobramentos

Além dos resultados já alcançados, as seguintes atividades se originaram no escopo desta investigação:

- Projeto de mineração de dados educacionais do projeto UAB. Durante o estudo das técnicas de mineração de dados e aprendizagem de máquina foi visualizada uma oportunidade de usar as bases de dados educacionais que estão distribuídas pelos vários Ambientes Virtuais de Aprendizagem participantes do projeto UAB;
- Foi desenvolvida uma abordagem para elicitação de requisitos para serviços inteligentes, estando a publicação de tal relato sendo encaminhada;
- A plataforma MOrFEU foi reconstruída incorporando a camada de serviços inteligentes e está disponível no site do projeto ([gsiufam.com](http://gsiufam.com)) e no [github.com](https://github.com).

## 8.3 Trabalhos Futuros

O projeto de mineração de dados educacionais do projeto UAB vai avançar para uma fase de coleta de dados em vários ambientes da UAB com objetivo de coletar uma massa de dados que possa ser considerada como BigData além de potencialmente criar a primeira base de dados educacionais publica do Brasil.

Seria interessante o desenvolvimento de uma camada de programação em alto nível para o Fantasos que disponibiliza-se um conjunto mínimo de tarefas básicas que os usuários poderiam utilizar para criar serviços mais complexos sem a necessidade de

reprogramarem o ambiente, levando a flexibilidade do MOrFEU (usando o Fantasos) mais um passo adiante.

## 8.4 Considerações Finais

O trabalho apresentado busca contribuir com os Ambientes Colaborativos Flexíveis ao oferecer a possibilidade de criação de Serviços Inteligentes para os usuários dos mesmos.

Como instância específica da classe dos problemas foi utilizado o Projeto MOrFEU pois sua infraestrutura de serviços básicos foi desenvolvida segundo o arcabouço teórico de Ambientes Colaborativos Flexíveis. No entanto o MorFEU ainda precisava, para ser utilizado em sua plenitude, de serviços que incorporam funcionalidades usualmente associados com o comportamento inteligente.

Além da relevância acadêmica decorrente dos temas investigados, o projeto aqui descrito apresenta um potencial para interferir positivamente no cenário da pesquisa local, não apenas por estar associado a um programa de pós-graduação de referência local e nacional como também por tratar de temas estratégicos para a região amazônica, como a formação de redes colaborativas de trabalho e aprendizagem através de ambientes virtuais que sejam flexíveis e aderentes às diferentes realidades e atores da região.

## Referências

- ABADI, M. et al. TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, p. 265–284, 2016. Disponível em: <<https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>>. Citado na página 20.
- ALIBHAI, Z. What is Contract Net Interaction Protocol? *IRMS Laboratory*, 2003. ISSN 00043702. Citado na página 104.
- ANAYA, A. R.; BOTICARIO, J. G. Ranking Learner Collaboration According to their Interactions. In: *IEEE EDUCON Education Engineering*. Madrid, Spain: IEEE, 2010. p. 797–803. ISBN 9781424465712. Citado 8 vezes nas páginas 58, 59, 87, 156, 162, 163, 166 e 168.
- ARRUDA, C. R.; PIMENTEL, M. d. G. C. Projeto e Implementação de um Sistema Colaborativo de Edição. *Revista Eletrônica de Iniciação Científica da Sociedade Brasileira de Computação (REIC-SBC)*, 2001. Disponível em: <<http://143.54.31.10/reic/edicoes/2001e2/cientificos/ProjetoImplementacaodeumSistemaColaborativodeEdicao.pdf>>. Citado na página 46.
- AZEVEDO, B. F. T.; TAVARES, O. d. L. Um Ambiente Inteligente para Aprendizagem Colaborativa. In: UFES (Ed.). *Simpósio Brasileiro de Informática na Educação (SBIE)*. Vitória, ES, Brasil: [s.n.], 2001. p. 331–339. Disponível em: <<http://ceie-sbc.tempsite.ws/pub/index.php/sbie/article/view/144>>. Citado 7 vezes nas páginas 55, 58, 87, 156, 162, 166 e 168.
- BAKER, B. M. *A conceptual framework for making knowledge actionable through capital formation*. [S.l.]: University of Maryland University College, 2007. Citado 2 vezes nas páginas 83 e 200.
- BAKER, R.; ISOTANI, S.; CARVALHO, A. Mineração de Dados Educacionais: Oportunidades para o Brasil. *Revista Brasileira de Informática na Educação*, v. 19, n. 02, p. 3–13, aug 2011. ISSN 1414-5685. Disponível em: <<http://www.br-ie.org/pub/index.php/rbie/article/view/1301>>. Citado na página 193.
- BARBOZA, A. P. L.; MOTTA, C. L. R. da; SANTORO, F. M. Hipermídia Adaptativa para Aprendizado Colaborativo. In: *Simpósio Brasileiro de Sistemas Colaborativos*. Vila Velha, ES, Brasil: IEEE, 2008. p. 88–98. ISBN 978-0-7695-3500-5. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4700788>>. Citado na página 54.
- BELTRAME, W. A. R.; MONTEIRO, E. R.; CURY, D. *Multi-Organizador Flexível de Espaços Virtuais*. Fortaleza, Ceara: [s.n.], 2008. Citado 2 vezes nas páginas 48 e 49.
- BETTSTETTER, C.; RENNEN, C. A comparison of service discovery protocols and implementation of the service location protocol. *Proceedings of the 6th EUNICE Open European Summer School Innovative Internet Applications*, p. 13–15, 2000. Disponível

em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.115.4652{&}rep=rep1{&}ty>>. Citado na página 104.

BICA, F.; VERDIN, R.; VICARI, R. M. Projeto de um Agente Fuzzy para Inferir a Auto-Eficácia do Aluno no Contexto de Sistemas Tutores Inteligentes. *IEEE Latin America Transactions*, v. 4, n. 6, p. 423–428, 2006. ISSN 1548-0992. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4472147>>. Citado 9 vezes nas páginas 54, 58, 59, 87, 156, 162, 163, 166 e 168.

BITTENCOURT, I. I.; COSTA, E. Modelos e Ferramentas para a Construção de Sistemas Educacionais Adaptativos e Semânticos. *Revista Brasileira de Informática na Educação*, v. 19, n. 01, p. 85, 2011. ISSN 1414-5685. Disponível em: <<http://www.br-ie.org/pub/index.php/rbie/article/view/1300>>. Citado 9 vezes nas páginas 54, 56, 58, 59, 156, 162, 163, 166 e 168.

BLAKE, M. B. Coordinating multiple agents for workflow-oriented process orchestration. *Information Systems and e-Business Management*, v. 1, n. 4, p. 387–404, 2003. ISSN 16179846. Disponível em: <<http://www.springerlink.com/index/TK7JDGBRAQ0PYWCR.pdf>>. Citado na página 104.

BORGES, M. R. d. S.; CAVALCANTI, M. C. R.; CAMPOS, M. L. M. Suporte por Computador ao Trabalho Cooperativo. In: *Congresso da Sociedade Brasileira de Computação*. Canela, RS, Brasil: SBC, 1995. v. 15, n. 1. Disponível em: <[http://www.researchgate.net/publication/228584209{\\\\_}Suporte{\\\\_}por{\\\\_}Computador{\\\\_}ao{\\\\_}Traba](http://www.researchgate.net/publication/228584209{\\_}Suporte{\\_}por{\\_}Computador{\\_}ao{\\_}Traba)>. Citado 5 vezes nas páginas 42, 45, 58, 151 e 161.

BROOKS, R. A. A Robust Layered Control System For A Mobile Robot. *IEEE Journal on Robotics and Automation*, v. 2, n. 1, p. 14–23, 1986. ISSN 08824967. Citado 3 vezes nas páginas 37, 38 e 39.

BROWN, P. et al. Reference Architecture Foundation for Service Oriented Architecture Version 1.0. *OASIS Standard*, 2012. Disponível em: <<https://www.oasis-open.org/committees/download.php/42644/GOLDSTANDARD-2011-06-16.pdf>>. Citado 2 vezes nas páginas 23 e 90.

BURSTEIN, M. et al. A semantic Web services architecture. *IEEE Internet Computing*, v. 9, n. 5, p. 72–81, 2005. ISSN 10897801. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1510607>>. Citado na página 104.

CAMPO, C. Service Discovery in Pervasive Multi-Agent Systems. *Discovery*, 2002. Citado na página 104.

CARNEIRO, M. L. F.; GELLER, M.; TAROUÇO, L. M. R. Groupware e os Ambientes para EAD. *Informática na Educação: Teoria & Prática*, Porto Alegre/RS, v. 5, n. 2, p. 11–21, nov 2002. Disponível em: <<http://www.seer.ufrgs.br/InfEducTeoriaPratica/article/download/4959/3479>>. Citado na página 45.

CARO, M. F.; JUMÉNEZ, J. A.; PATERNINA, A. M. Architectural Modeling of Metamemory Judgment in Case-based Reasoning Systems. *Conferencia Latinoamericana En Informatica (CLEI)*, Ieee, p. 1–8, 2012. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6427152>>. Citado 7 vezes nas páginas 58, 59, 157, 162, 163, 166 e 168.

- CHEN, H.-d.; ZHANG, X.-z. A Multi-paradigm Learning Model Based on Multi-agent Technology. In: *International Conference on Machine Learning and Cybernetics*. Beijing: IEEE, 2002. v. 1, p. 1387 – 1390. Citado 6 vezes nas páginas 58, 59, 157, 162, 166 e 168.
- CONCEIÇÃO, K. N. da. Uma Aplicação Web com Agentes Inteligentes para dar Suporte ao Ensino à Distância na Área de Gestão do Conhecimento. In: *Congreso Argentino de Ciencias de la Computación*. [s.n.], 2000. ISBN <http://hdl.handle.net/10915/23487>. Disponível em: <http://hdl.handle.net/10915/23487>. Citado 5 vezes nas páginas 55, 58, 162, 166 e 168.
- CORDEIRO, A. D.; BASTOS, R. C. Potencializando Decisões Gerenciais com Inteligência Artificial. *IEEE Latin America Transactions*, v. 4, n. 6, p. 429–435, 2006. ISSN 1548-0992. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4472148>. Citado 6 vezes nas páginas 58, 59, 157, 162, 163 e 166.
- CORREIA, A.; FONSECA, B.; PAREDES, H. Computer Supported Cooperative Work: a field still to cogitate. In: *Iberian Conference on Information Systems and Technologies (CISTI)*. [S.l.]: IEEE, 2011. p. 1–5. Citado na página 45.
- DANTAS, B. T. et al. Risys – Uma Ferramenta de Apoio à Gerência de Riscos em um Ambiente Colaborativo de Gestão de Projetos. In: *Simpósio Brasileiro de Sistemas Colaborativos*. Fortaleza, Ceara: IEEE, 2009. p. 90–98. ISBN 978-1-4244-5292-7. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5460512>. Citado 7 vezes nas páginas 58, 59, 157, 162, 163, 166 e 168.
- DANTAS, B. T. et al. Uma Ferramenta Web de Apoio à Coordenação de Projetos em um Ambiente Colaborativo. In: *Simpósio Brasileiro de Sistemas Colaborativos*. IEEE, 2008. p. 146–157. ISBN 978-0-7695-3500-5. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4700793>. Citado 8 vezes nas páginas 54, 58, 59, 157, 162, 163, 166 e 168.
- DONG, A.; LI, H. Ontology-based Information Integration in Virtual Learning Environment. In: *IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE, 2005. p. 762 – 765. ISBN 076952415X. Disponível em: [http://ieeexplore.ieee.org/xpls/abs{\\\\_}all.jsp?arnumber=1517](http://ieeexplore.ieee.org/xpls/abs{\\_}all.jsp?arnumber=1517). Citado 4 vezes nas páginas 58, 157, 162 e 166.
- DUPPLAW, D.; BESANA, P.; ROBERTSON, D. *OpenKnowledge Manual*. [S.l.], 2008. Disponível em: <http://groups.inf.ed.ac.uk/OK/Deliverables/D9.2.pdf>. Citado 3 vezes nas páginas 103, 105 e 107.
- ELIAS, T. *Learning Analytics : Definitions , Processes and Potential*. 2011. 23 p. Disponível em: <http://learninganalytics.net/LearningAnalyticsDefinitionsProcessesPotential.pdf>. Citado 4 vezes nas páginas 83, 194, 195 e 200.
- ELLIS, C.; WAINER, J. Groupware and Computer Supported Cooperative Work. In: *Multiagent systems: a modern approach to distributed artificial intelligence*. [S.l.]: MIT Press, 1999. cap. 10, p. 425. ISBN 0-262-23203-0. Citado 2 vezes nas páginas 42 e 44.
- ELLIS, C. A.; GIBBS, S. J.; REIN, G. Groupware: some issues and experiences. *Communications of the ACM*, v. 34, n. 1, p. 39–58, 1991. ISSN 00010782. Citado 5 vezes nas páginas 39, 40, 42, 44 e 151.

FIGUEIRA, A. d. S. et al. Módulo de Avaliação Automática de Questões Discursivas no Ambiente Virtual de Aprendizagem LabSQL. In: *Iberian Conference on Information Systems and Technologies (CISTI)*. Lisboa: IEEE, 2013. p. 1–5. Citado 8 vezes nas páginas 19, 58, 59, 157, 162, 163, 166 e 168.

FILHO, J. A. B. L.; QUARTO, C. C.; FRANCA, R. M. Clustering Algorithm for the Socio-affective Groups Formation in Aid of Computer Supported Collaborative Learning. In: *Simpósio Brasileiro de Sistemas Colaborativos*. IEEE, 2010. p. 24–27. ISBN 978-1-4244-8446-1. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5698527>>. Citado 7 vezes nas páginas 54, 57, 58, 59, 158, 162 e 166.

FILIPPO, D.; PIMENTEL, M.; WAINER, J. Metodologia de pesquisa científica em sistemas colaborativos. In: *Sistemas Colaborativos*. [s.n.], 2012. p. 379–404. ISBN 9788535246698. Disponível em: <<http://www.elsevier.com.br/site/produtos/Detalhe-Produto.aspx?tid=82736{&}seg=3{&}tit=SISTEMASCOLABOR>>. Citado na página 28.

FONTES, L. M. d. O. et al. An Animated Pedagogical Agent to Support Problem-Based Learning. *Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de*, v. 8, n. 2, p. 56–63, 2013. ISSN 1932-8540. Citado 7 vezes nas páginas 20, 58, 59, 159, 162, 166 e 168.

FREDDO, A. R. et al. Uma Arquitetura Multiagente para o Suporte ao Desenvolvimento Colaborativo de Software em Pequenas Equipes. *Simpósio Brasileiro de Sistemas Colaborativos*, Ieee, p. 178–183, 2009. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5460502>>. Citado 6 vezes nas páginas 54, 58, 158, 162, 166 e 168.

FUKS, H. et al. *The 3c collaboration model*. [S.l.]: Information Science Reference (an imprint of IGI Global), 2007. 637–644 p. Citado 2 vezes nas páginas 40 e 41.

GATIUS, M.; GONZ, M. Using Domain Knowledge for Fostering the Collaborative Ability of a Web Dialogue System. In: *International Conference on Next Generation Web Services Practices (NWeSP)*. [S.l.]: IEEE, 2011. p. 129–134. ISBN 9781457711275. Citado 7 vezes nas páginas 58, 59, 158, 162, 163, 166 e 168.

GHARSELLAOUI, A.; BELLIK, Y.; JACQUET, C. Requirements of Task Modeling in Ambient Intelligent Environments. *International Conference on Ambient Computing, Applications, Services and Technologies*, n. Section V, p. 71–78, 2012. Citado 4 vezes nas páginas 32, 79, 80 e 81.

GOULART, R.; GIRAFFA, L. Utilizando a Tecnologia de Agentes na Construção de Sistemas Tutores Inteligentes em Ambiente Interativo. In: UFES (Ed.). *Simpósio Brasileiro de Informática na Educação (SBIE)*. Vitória, ES, Brasil: [s.n.], 2001. p. 419–430. Disponível em: <<http://ceie-sbc.tempsite.ws/pub/index.php/sbie/article/view/154>>. Citado 7 vezes nas páginas 58, 59, 158, 162, 163, 166 e 168.

GRUDIN, J.; POLTROCK, S. Taxonomy and Theory in Computer Supported Cooperative Work. *The oxford handbook of organizational psychology*, n. 2012, p. 1323–1348, 2012. Citado 5 vezes nas páginas 42, 43, 44, 45 e 46.

GUTIÉRREZ-CARREÓN, G.; DARADOUMIS, T.; JORBA, J. Toward a Semantic Approach for Automatic Composition of Learning Grid Services. In: *International*

*Conference on Complex, Intelligent and Software Intensive Systems*. [S.l.]: IEEE, 2008. p. 940–945. ISBN 0769531091. Citado 4 vezes nas páginas 58, 158, 162 e 166.

GUTIERREZ-GARCIA, J. O.; SIM, K. M. Self-organizing agents for service composition in cloud computing. *Proceedings - 2nd IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2010*, p. 59–66, 2010. ISSN 18650929. Citado na página 104.

HUHNS, M. N.; SINGH, M. P. Personal Assistants. *IEEE Internet Computing*, v. 2, n. 5, p. 90–92, 1998. Disponível em: <<http://dx.doi.org/10.1109/4236.722236>>. Citado na página 72.

IEPSEN, E.; BERCHT, M.; REATEGUI, E. Detection and Assistance to Students who Show Frustration in Learning of Algorithms. In: *Frontiers in Education (FIE)*. Oklahoma City, OK: IEEE, 2013. p. 1183 – 1189. ISBN 9781467352611. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs/\\_all.jsp?arnumber=6685](http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=6685)>. Citado 8 vezes nas páginas 19, 58, 59, 158, 162, 163, 166 e 168.

JENNINGS, N. R.; WOOLDRIDGE, M. Applications of Intelligent Agents. *Electronics Information and Planning*, v. 26, n. 5, p. 273–281, 1999. ISSN 03049876. Citado na página 35.

KANG, J.; SIM, K. M. Towards agents and ontology for cloud service discovery. *Proceedings - 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2011*, n. 4, p. 483–490, 2011. Citado na página 104.

KEMCZINSKI, A. et al. Colaboração e Cooperação – Pertinência, Concorrência ou Complementaridade. *Revista Produção Online*, Florianópolis, v. 7, n. 3, nov 2008. Disponível em: <<http://www.producaoonline.org.br/rpo/article/view/68>>. Citado na página 41.

KUNZEL, M. F. et al. DIMI 3D - Companion Agent in a Learning Virtual Environment. In: *Workshop and School of Agent Systems, their Environment and Applications*. IEEE, 2011. p. 107–109. ISBN 978-0-7695-4645-2. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6209291>>. Citado 7 vezes nas páginas 54, 58, 59, 158, 162, 166 e 168.

LEE, Y.; CHOI, J.; KIM, T. Discriminating factors between completers of and dropouts from online learning courses. *British Journal of Educational Technology*, v. 44, n. 2, p. 328–337, 2013. ISSN 00071013. Citado na página 98.

LI, Y.; MEI, L.; WANG, J. A Personalized Recommendation System in E-Learning Environment Based on Semantic Analysis. In: *International Conference on New Trends in Information Science and Service Science and Data Mining (ISSDM)*. IEEE, 2012. p. 802–807. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs/\\_all.jsp?arnumber=6528](http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=6528)>. Citado 5 vezes nas páginas 58, 159, 162, 166 e 168.

LIMA, M. P.; DAVID, J. M. N.; DANTAS, B. T. Gerenciamento de Riscos e Contexto em Ambientes Colaborativos de Gestão de Projetos de Desenvolvimento de Software. In: *Simpósio Brasileiro de Sistemas Colaborativos*. IEEE, 2010. p. 95–102. ISBN 978-1-4244-8445-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5698502>>. Citado 7 vezes nas páginas 58, 59, 159, 162, 163, 166 e 168.

- LUGO, G. A. G.; TACLA, C. A. Inferring Activities of an Actor by Means of Context Ontologies. *Conferencia Latinoamericana En Informatica (CLEI)*, Ieee, p. 1–10, oct 2012. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6427159>>. Citado 8 vezes nas páginas 54, 58, 59, 159, 162, 163, 166 e 168.
- MACKENZIE, C. M. et al. Reference Model for Service Oriented Architecture 1.0. *OASIS Standard*, 2006. Disponível em: <<https://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdfhttp://lists.oasis-open.org/archives/soa-rm-editors/200511/pdf00001.pdf>>. Citado 4 vezes nas páginas 32, 73, 74 e 199.
- MAFFON, H. P. et al. Modelagem da Arquitetura de um Sistema Tutor Inteligente Aplicado ao Câncer de Mama. In: *Pan American Health Care Exchanges (PAHCE)*. Medellín, Colombia: IEEE, 2013. p. 1–6. ISBN 978-1-4673-6257-3. ISSN 2327-8161. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6568295>>. Citado 8 vezes nas páginas 54, 58, 59, 159, 162, 163, 166 e 168.
- MAHMOUD, Q. H.; MAAMAR, Z. Applying the MVC design pattern to multi-agent systems. *Canadian Conference on Electrical and Computer Engineering*, n. May, p. 2420–2423, 2007. ISSN 08407789. Citado na página 91.
- MAITHREYE, S.; DEVI, M. M.; ENGINEERING, E. Multi-Agent Framework For Cloud Service Composition. p. 2–6, 2016. Citado na página 104.
- MARONI, V. et al. SimDeCS: Architecture of a Multiagent System for Simulation of Decision Making in Health Care. In: *Workshop and School of Agent Systems, their Environment and Applications*. IEEE, 2011. p. 110–112. ISBN 978-0-7695-4645-2. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6209292>>. Citado 8 vezes nas páginas 56, 58, 59, 159, 162, 163, 166 e 168.
- MELO, B. M. de et al. Módulo de Avaliação Automática de Código SQL com Feedback em Forma de “Dicas” no Ambiente Virtual de Aprendizagem LabSQL. *Iberian Conference on Information Systems and Technologies (CISTI)*, p. 1–7, 2013. Citado 8 vezes nas páginas 20, 58, 59, 159, 162, 163, 166 e 168.
- MELO, F. R. de et al. Conteúdo Didático Multinível para Personalização Reativa em Sistemas Tutores Inteligentes. *Controle y Automacao*, v. 23, n. 6, p. 679–693, dec 2012. ISSN 0103-1759. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84873303227&partnerID=tZotx>>. Citado 8 vezes nas páginas 56, 58, 59, 160, 162, 163, 166 e 168.
- MENDONÇA, A. P. et al. Um Ambiente Telemático para Mediar a Controvérsia Acadêmica. In: *Simpósio Brasileiro de Informática na Educação (SBIE)*. [S.l.: s.n.], 2003. p. 346–355. Citado na página 128.
- MENEZES, C. S. et al. MOrFEU – Multi-Organizador Flexível de Espaços Virtuais para Apoiar a Inovação Pedagógica em EAD. *Simpósio Brasileiro de Informática na Educação (SBIE)*, v. 1, n. 1, p. 451–460, 2008. Disponível em: <<http://ceie-sbc.educacao.ws/pub/index.php/sbie/article/view/727/713>>. Citado 5 vezes nas páginas 21, 23, 27, 47 e 48.
- NETO, F. A. d. A.; CASTRO, A. Elicited and mined rules for dropout prevention in online courses. In: *Proceedings - Frontiers in Education Conference, FIE*. [S.l.: s.n.], 2015. v. 2014. ISBN 9781479984534. ISSN 15394565. Citado 2 vezes nas páginas 98 e 138.

NETO, F. A. d. A.; CASTRO, A. A Reference Architecture for Educational Data Mining. In: *Proceedings - Frontiers in Education Conference, FIE*. [S.l.: s.n.], 2017. ISBN 9781509059201. Citado na página 138.

NOVAK, J. D.; CAÑAS, a. J. The Theory Underlying Concept Maps and How to Construct and Use Them. *IHMC CmapTools*, p. 1–36, 2008. ISSN 1809-4398. Disponível em: <<http://cmap.ihmc.us/Publications/ResearchPapers/TheoryUnderlyingConceptMaps.pdf>>. Citado na página 75.

OLAND, E.; ANDERSEN, T. S.; KRISTIANSEN, R. Subsumption architecture applied to flight control using composite rotations. *Automatica*, Elsevier Ltd, v. 69, p. 195–200, 2016. ISSN 00051098. Disponível em: <<http://dx.doi.org/10.1016/j.automatica.2016.02.034>>. Citado na página 39.

OLIVEIRA, E. A.; TEDESCO, P. I-Collaboration: Um Modelo de Colaboração Inteligente Personalizada para ambientes de EAD. *Revista Brasileira de Informática na Educação*, v. 18, n. 1, 2010. Disponível em: <<http://ceie-sbc.educacao.ws/pub/index.php/rbie/article/view/1213>>. Citado 8 vezes nas páginas 56, 58, 59, 160, 162, 163, 166 e 168.

PARK, J. H. Factors Related to Learner Dropout in Online Learning. *Proceedings of the 2007 Academy of Human Resource Development Annual Conference*, n. 2005, p. 1–8, 2007. Disponível em: <<http://www.eric.ed.gov/ERICWebPortal/recordDetail?accno=ED504556>>. Citado na página 98.

PARK, J.-h.; CHOI, H. J. Factors Influencing Adult Learners' Decision to Drop Out or Persist in Online Learning. *Journal of Educational Technology & Society*, v. 12, n. 4, p. 207–217, 2009. Citado na página 98.

PENICHET, V. M. R. et al. A Classification Method for CSCW Systems. *Electronic Notes in Theoretical Computer Science*, v. 168, n. SPEC. ISS., p. 237–247, 2007. ISSN 15710661. Citado 2 vezes nas páginas 42 e 43.

PESSOA, J.; NETTO, H.; MENEZES, C. S. FAmCorA: um Framework para a Construção de Ambientes Cooperativos Inteligentes de Apoio a Aprendizagem na Internet Baseado em Web Services e Agentes. In: UNISINOS (Ed.). *Simpósio Brasileiro de Informática na Educação (SBIE)*. São Leopoldo, RS, Brasil: [s.n.], 2002. p. 94–104. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/169>>. Citado 8 vezes nas páginas 55, 58, 59, 160, 162, 163, 166 e 168.

PÍCCOLO, H. L. et al. Ambiente Interativo e Adaptável para ensino de Programação. In: ICC (Ed.). *Simpósio Brasileiro de Informática na Educação (SBIE)*. SBC, 2011. p. 2–9. Disponível em: <<http://ceie-sbc.educacao.ws/pub/index.php/sbie/article/view/1496>>. Citado na página 53.

POZNA, C.; FOLDESI, P.; KOVACS, J. The Personal Assistant Application, Problem Definition. In: *International Conference on Cognitive Infocommunications*. Budapest, Hungary: IEEE, 2013. p. 851–856. ISBN 9781479915460. Citado na página 72.

PRIMO, T. T.; VICARI, R. M. A Recommender System that Allows Reasoning and Interoperability over Educational Content Metadata. In: *IEEE International Conference on Advanced Learning Technologies (ICALT)*. Athens, GA: IEEE, 2011. p. 598 – 599. ISBN 9780769543468. Citado 7 vezes nas páginas 58, 59, 160, 162, 163, 166 e 168.

- RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to Recommender Systems. In: *Recommender Systems Handbook*. Springer, Boston, MA, 2011. cap. 1, p. 1–35. ISBN 978-0-387-85820-3. Disponível em: <<http://www.inf.unibz.it/~ricci/papers/intro-rec-sys-handbook.p>>. Citado na página 125.
- ROBERTSON, D. A Lightweight Coordination Calculus for Agent Systems. n. JULY 2004, p. 4–6, 2004. Citado 4 vezes nas páginas 103, 105, 106 e 182.
- ROMERO, C.; VENTURA, S.; GARCÍA, E. Data Mining in Course Management Systems: Moodle Case Study and Tutorial. *Computers & Education*, v. 51, n. 1, p. 368–384, aug 2008. ISSN 03601315. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360131507000590><http://linkinghub.elsevier.com/retrieve/pii/S0360131507000590>>. Citado na página 195.
- ROWLEY, J. The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science*, v. 33, n. 2, p. 163–180, 2007. ISSN 0165-5515. Disponível em: <<http://alturl.com/7qike>>. Citado 2 vezes nas páginas 83 e 200.
- RUSSEL, S.; NORVIG, P. *Inteligência Artificial*. Terceira. [S.l.]: Elsevier Editora Ltda, 2013. Citado 3 vezes nas páginas 33, 34 e 36.
- RUTA, M. et al. Semantic-based resource discovery and orchestration in Home and Building Automation: a multi-agent approach. *IEEE Transactions on Industrial Informatics*, v. 10, n. 1, p. 730–741, 2014. ISSN 1551-3203. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6560391>>. Citado na página 104.
- SANTOS, A. V.; REIS, M. M. Ambiente Virtual Inteligente para o ensino de Controle Estatístico da Qualidade - STCEQ.Net. In: *Workshop de Informática na Escola*. São Leopoldo, RS, Brasil: SBC, 2005. p. 2477–2485. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/832>>. Citado na página 53.
- SANTOS, L. N. *Um Modelo Conceitual para Ambientes Virtuais Flexíveis*. 216 p. Tese (Doutorado) — UFAM, 2013. Citado 4 vezes nas páginas 49, 51, 85 e 128.
- SANTOS, L. N.; CASTRO, A. N.; MENEZES, C. S. Morfeu : Criando Ambientes Virtuais Flexíveis Na Web Para Mediar a Colaboração. In: *Congresso Iberoamericano de Informática Educativa*. Santiago/Chile: [s.n.], 2010. p. 114–121. Citado 5 vezes nas páginas 21, 23, 47, 50 e 51.
- SANTOS, L. N.; CASTRO, A. N.; MENEZES, C. S. Flexible Virtual Environments for Teaching and Learning. In: *Frontiers in Education (FIE)*. [s.n.], 2012. p. 1–6. ISBN 978-1-4673-1352-0. ISSN 15394565. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6462495>>. Citado 3 vezes nas páginas 21, 23 e 47.
- SPRAGUE, R. H. A Framework for the Development of Decision Support Systems. *MIS quarterly*, v. 4, n. 4, p. 1–26, 1980. ISSN 0276-7783. Citado na página 126.
- STAHL, G.; KOSCHMANN, T.; SUTHERS, D. Computer-supported collaborative learning: An historical perspective. *Cambridge handbook of the learning sciences*, v. 2006, p. 409–426, 2006. ISSN 10508406. Disponível em: <[http://gerrystahl.net/cscl/CSCL{\\\\_}English.](http://gerrystahl.net/cscl/CSCL{\\_}English.)> Citado 2 vezes nas páginas 45 e 46.

- STERLING, L.; TAVETER, K. *The Art of Agent-Oriented Modeling*. Cambridge, Massachusetts: The MIT Press, 2009. ISBN 9780262013116. Citado 4 vezes nas páginas 33, 35, 36 e 37.
- THIRY, M.; BARCIA, R. Um Ambiente de Ensino Colaborativo Baseado em Agentes Inteligentes. *Congreso Argentino de Ciencias de la Computación*, 1999. Disponível em: <<http://sedici.unlp.edu.ar/handle/10915/23496>>. Citado 8 vezes nas páginas 55, 58, 59, 160, 162, 163, 166 e 168.
- TOBAR, C. M. et al. Uma Arquitetura de Ambiente Colaborativo para o Aprendizado de Programação. In: UFES (Ed.). *Simpósio Brasileiro de Informática na Educação (SBIE)*. Vitória, ES, Brasil: [s.n.], 2001. v. 1, n. 1, p. 367–376. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/148>>. Citado 8 vezes nas páginas 56, 58, 59, 160, 162, 163, 166 e 168.
- TONG, Y.; LU, D. A Flexible Multimedia Cooperative Product Design Platform Framework. In: *International Conference on Computer Supported Cooperative Work in Design*. IEEE, 2002. p. 198–204. Disponível em: <<http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=1047>>. Citado 4 vezes nas páginas 58, 160, 162 e 166.
- TURNER, J. T.; GIVIGI, S. N.; BEAULIEU, A. Implementation of a subsumption based architecture using model-driven development. *SysCon 2013 - 7th Annual IEEE International Systems Conference, Proceedings*, p. 331–338, 2013. Citado na página 39.
- VICENTE, D.; VASCONCELOS, J.; RESTIVO, F. Multi-agent Systems in Education. In: *Iberian Conference on Information Systems and Technologies (CISTI)*. [S.l.]: IEEE, 2011. p. 1–6. Citado 7 vezes nas páginas 58, 59, 161, 162, 163, 166 e 168.
- WILGES, B. et al. Integration of BDI agent with Fuzzy Logic in a Virtual Learning Environment. *IEEE Latin America Transactions*, v. 10, n. 1, p. 1370–1376, jan 2012. ISSN 1548-0992. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6142486>>. Citado 8 vezes nas páginas 55, 58, 59, 161, 162, 163, 166 e 168.
- WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, v. 10, n. 02, p. 115, 1995. ISSN 0269-8889. Disponível em: <<http://www.journals.cambridge.org/abstract/S0269888900008>>. Citado 2 vezes nas páginas 35 e 39.
- YUAN, S.-t.; CHEN, Y.-c. Semantic Ideation Learning for Agent-Based E-Brainstorming. *Knowledge and Data Engineering, IEEE Transactions on*, v. 20, n. 2, p. 261–275, 2008. Citado 7 vezes nas páginas 58, 59, 161, 162, 163, 166 e 168.
- ZAMBIASI, S. P.; RABELO, R. J. Um Arcabouço para Assistentes de Software Virtuais em Ambientes Colaborativos. *WESAAC 2008 – II Workshop - Escola de Sistemas de Agentes para Ambientes Colaborativos*, 2008. Disponível em: <[http://www.gsigma.ufsc.br/publications/files2/wesaac2008\(Saulo\).pdf](http://www.gsigma.ufsc.br/publications/files2/wesaac2008(Saulo).pdf)>. Citado 8 vezes nas páginas 55, 58, 60, 161, 162, 165, 166 e 168.

# Apêndices

# APÊNDICE A – Revisão Sistemática sobre Serviços Inteligentes em Ambientes Colaborativos

## A.1 Introdução

A implementação de Serviços Inteligentes em softwares já existentes vem melhorando e apoiando a utilização de outras ferramentas pelo homem. Como exemplo pode-se citar a existência de sistemas de recomendação auxiliando os usuários na escolha de opções em um dado sistema. Outro exemplo é o uso de sistemas de apoio a decisão fornecendo cursos de ação juntamente com justificativas sobre a escolha de um determinado curso. E temos também casos de criação de Serviços Inteligentes em Ambientes Colaborativos na Web melhorando a utilização dos ambientes pelos seus usuários. No entanto, ainda é pouco explorado o problema de se criar Serviços Inteligentes em Ambientes Colaborativos.

Portanto faz-se necessário um estudo dos Serviços Inteligentes já fornecidos em Ambientes Colaborativos, identificando assim quais são os problemas resolvidos, os Serviços Inteligentes utilizados, bem como problemas ainda em aberto. Espera-se que, ao analisar os pontos em comum entre os Serviços Inteligentes já existentes, ser possível gerar novo conhecimento na forma de modelos e frameworks para a criação de serviços. Espera-se ainda poder utilizar tal modelagem de forma genérica na resolução de problemas na área.

### A.1.1 Conceitos Básicos e Definições

Para um melhor aproveitamento deste trabalho são apresentados alguns conceitos chaves e definições utilizados durante a pesquisa aqui realizada.

#### A.1.1.1 Cooperação X Colaboração

Os termos cooperação e colaboração acabam sendo utilizados de forma intercambiável em alguns trabalhos encontrados na literatura. Em outros trabalhos, é feita uma distinção entre o significado dos termos, mesmo que em alguns casos ainda seja possível haver dúvidas para classificar alguma coisa como cooperação ou colaboração.

Desta forma, optamos por não fazer distinção entre os termos, utilizando tanto a palavra cooperação como colaboração para o mesmo significado. No entanto, teve-se o cuidado de analisar se o sentido dado pelos autores dos artigos para a palavra cooperação

ou colaboração é o mesmo do modelo de colaboração 3C (ELLIS; GIBBS; REIN, 1991). Sendo este também o sentido adotado neste trabalho.

#### A.1.1.2 Serviços Inteligentes no Contexto deste Trabalho

Todo sistema de informática fornece um ou mais serviços aos seus usuários, desde serviços simples (inserção de registros em banco de dados) até serviços complexos (busca por padrões em bancos de dados de larga escala). Alguns desses serviços podem ser ou não considerados inteligentes. A classificação de Serviços como sendo ou não Inteligentes pode ser um trabalho bastante complexo, porém devido ao foco deste trabalho, foi-se decidido fazer uma simplificação nessa classificação. Desta forma, neste trabalho, para que um serviço seja considerado inteligente ele deve ter os seguintes requisitos atendidos:

- Fornecer um serviço não trivial;
- Ter sido implementado utilizando alguma técnica de Inteligência Artificial.

#### A.1.1.3 Ambientes Colaborativos e Ambientes Virtuais de Aprendizagem

Segundo (BORGES; CAVALCANTI; CAMPOS, 1995), um ambiente colaborativo deve:

- Implementar processos de apoio à cooperação;
- Possibilitar a produção em conjunto;
- Permitir a troca de informações.

Assim, um ambiente computacional com esses pré-requisitos é chamado Groupware ou Sistema de Trabalho Cooperativo apoiado por Computador (CSCW – *Computer Supported Cooperative Work*).

Além disso, segundo (BORGES; CAVALCANTI; CAMPOS, 1995), os Sistemas de Aprendizagem Cooperativo apoiado por Computador (CSCL – *Computer Supported Cooperative Learning*), em português mais conhecidos como Ambientes Virtuais de Aprendizagem – AVA, são um subconjunto dos CSCW.

Por causa disso, nesta revisão sistemática são apresentados os resultados das análises tanto de pesquisas em *Groupware/CSCW* como de pesquisas em AVA/CSCL.

### A.1.2 Questões de Pesquisa

Com o objetivo de orientar o trabalho desta revisão sistemática, foram elencadas duas questões para serem respondidas conforme a análise de cada artigo. Essas questões são:

- **Quais Serviços Inteligentes estão em uso no ambiente colaborativo?** Ao analisar os artigos, focamos a atenção aos Serviços Inteligentes implementados ou em construção nos artigos. Foi-se observado qual a necessidade atendida pelo serviço e como ele resolvia o problema. O objetivo de fazer tal trabalho é para investigar se existem elementos comuns, quais técnicas foram utilizadas na construção e montar uma lista de itens a serem unificados e reaproveitados.
- **Quais técnicas de Inteligência Artificial são citadas nesses trabalhos?** Finalmente, com esta indagação estamos interessados em saber quais técnicas de IA estão sendo aplicadas na solução dos problemas descritos nos artigos. Com isso, espera-se poder encontrar mais de uma técnica para resolver o mesmo problema aumentando assim a flexibilidade das soluções daquele problema.

## A.2 A Revisão Sistemática

A revisão sistemática foi realizada seguindo o procedimento descrito a seguir:

1. Especificação dos critérios de inclusão e exclusão de artigos;
2. Construção de um conjunto de controle;
  - a) Busca inicial por artigos relevantes;
  - b) Filtragem dos artigos.
3. Construção de strings de busca;
  - a) Especificação das strings;
  - b) Escolha dos indexadores de artigos científicos na web;
  - c) Validação das strings criadas.
4. Coleta dos artigos para a revisão sistemática;
  - a) Filtragem dos artigos baseado nos resumos;
  - b) Filtragem dos artigos baseado no conteúdo completo.
5. Análise dos resultados.

### A.2.1 Critérios de Inclusão e Exclusão de Artigos

Os critérios de inclusão e exclusão de artigos foram criados e divididos em duas partes de acordo com a fase da revisão sistemática para a qual eles foram utilizados:

- Durante a construção do conjunto de artigos de controle:

1. Incluir resultados considerados relevantes para as questões de pesquisa;
  2. Excluir teses e monografias devido ao tempo necessário para o processamento desses ser maior em relação ao tempo necessário para processamento de artigos científicos;
  3. Excluir relatórios técnicos e white papers devido a não terem necessariamente um processo de revisão por pares para a sua publicação;
  4. Excluir resultados não publicados em periódicos com revisão de pares.
- Durante a coleta dos artigos para a revisão sistemática:
    - Incluir artigos inseridos no contexto de CSCW ou CSCL;
    - Excluir artigos onde não é citado, indicado ou demonstrado algum serviço inteligente para CSCW ou CSCL;
    - Excluir artigos onde não são citadas, indicadas ou demonstradas as técnicas de Inteligência Artificial utilizadas.

## A.2.2 Construção do Conjunto de Controle

Para obter os artigos de controle para a realização da revisão sistemática foi adotado o seguinte procedimento:

1. Pesquisar no Google Acadêmico com a string "Ambiente Colaborativo Serviço Inteligente";
2. A cada lote de 20 resultados realizar a leitura integral dos resultados e aplicar os critérios de inclusão e exclusão;
3. Concluir o processo quando ao final de algum lote o conjunto de controle tiver mais de 5 artigos relevantes.

Como resultado foi montado um grupo de controle com 9 artigos.

## A.2.3 Strings de Busca

Os artigos do conjunto de controle foram utilizados para descobrir palavras-chaves e conceitos a serem utilizados na construção da string de busca. Nesta revisão sistemática foram encontradas tanto palavras-chaves em inglês como em português e portanto foram utilizadas duas strings de busca, uma em português (vide [Código 7](#)) e outra em inglês (vide [Código 8](#)). Assim para cada palavra-chave ou conceito escrito em português também foi utilizada o seu equivalente em inglês, o mesmo foi feito do inglês para o português.

As strings de busca foram testadas e validadas nos indexadores de artigos IEEEExplore<sup>1</sup> e Scopus<sup>2</sup>. Foram escolhidas essas duas máquinas de busca devido à sua abrangência, facilidade de uso e suporte ao uso de strings de busca grandes.

Para a próxima fase, as strings de busca foram utilizadas somente após passarem por uma validação. O teste para validar uma string de busca consistiu em realizar a busca utilizando a string sendo testada em todos os indexadores, fazer a união de todos os resultados encontrados e verificar se dentro dos resultados encontrados estão todos os artigos membros do conjunto de controle.

### A.2.3.1 Montagem das Strings de Busca

Seguindo-se a metodologia para revisão sistemática é necessário montar uma string de busca que será adaptada e utilizada nas máquinas de busca aonde serão feitas as pesquisas. Para tanto, começamos especificando a População, a Intervenção, a Comparação e o Resultado que estamos buscando nessa revisão sistemática:

- **População:** Ambientes Colaborativos
- **Intervenção:** Inteligência Artificial
- **Comparação:** N/A
- **Resultado:** Serviços Inteligentes

Após especificado o que estamos buscando alguns termos da string de busca foram facilmente identificados. Ainda assim, para se ter uma maior abrangência da pesquisa realizada foram retiradas palavras-chaves e expressões-chave dos artigos de controle.

```
1 ( "Ambiente Colaborativo" OR "Ambientes Colaborativos" OR "Ambiente
    Cooperativo" OR "Ambientes Cooperativos" OR "Aprendizagem
    Colaborativa" OR "Aprendizagem Cooperativa" OR "Groupware" OR "
    Ambiente Virtual de Aprendizagem" OR "Ambientes Virtuais de
    Aprendizagem"
) AND (
    "Inteligência Artificial" OR "Agente Inteligente" OR "Agentes
    Inteligentes" OR "Multiagente" OR "Multiagentes" OR "Lógica de
    Primeira Ordem" OR "Lógica de Predicados" OR "Rede Neural" OR "Redes
    Neurais" OR "Algoritmo Genético" OR "Algoritmos Genéticos" OR "Lógica
    Fuzzy" OR "Sistema Especialista" OR "Sistemas Especialistas" OR "
    Raciocínio Baseado em Regras" OR "Raciocínio Baseado em Casos" OR "
    Raciocínio Baseado em Memória" OR "Ontologia" OR "Ontologias" OR "
    Sistema de Recomendação" OR "Sistemas de Recomendação" OR "Motor de
```

<sup>1</sup> <<http://ieeexplore.ieee.org>>

<sup>2</sup> <<http://www.scopus.com>>

*Inferência*" OR "*Rede Semântica*" OR "*Redes Semânticas*" OR "*Semântica Latente*" OR "*Machine Learning*" OR "*Arquitetura BDI*" OR "*Regras de Produção*" OR "*Processamento de Língua Natural*" OR "*Processamento de Linguagem Natural*" OR "*Base de Conhecimento*"

) AND (

5 "*Serviço Inteligente*" OR "*Serviços Inteligentes*" OR "*Sistema Inteligente*" OR "*Sistemas Inteligentes*" OR "*Ambiente Inteligente*" OR "*Ambientes Inteligentes*" OR "*Ambiente Virtual Inteligente*" OR "*Ambientes Virtuais Inteligentes*" OR "*Ambiente Cooperativo Inteligente*" OR "*Ambiente Colaborativo Inteligente*" OR "*Busca Inteligente*" OR "*Assistente Inteligente*" OR "*Assistentes Inteligentes*")

Código 7 – String de Busca em Português

( "*Cooperative Work*" OR "*Cooperative Environment*" OR "*Cooperative Environments*" OR "*Collaborative Work*" OR "*Collaborative Environment*" OR "*Collaborative Environments*" OR "*Collaborative Learning*" OR "*Cooperative Learning*" OR "*Groupware*" OR "*CSCW*" OR "*Virtual Learning Environment*" OR "*Virtual Learning Environments*"

) AND (

3 "*Artificial Intelligence*" OR "*Intelligent Agent*" OR "*Intelligent Agents*" OR "*Multi-agent*" OR "*Multi-agents*" OR "*First-order Logic*" OR "*First-order Predicate Calculus*" OR "*Lower Predicate Calculus*" OR "*Quantification Theory*" OR "*Predicate Logic*" OR "*Artificial Neural Network*" OR "*Artificial Neural Networks*" OR "*Genetic Algorithm*" OR "*Genetic Algorithms*" OR "*Fuzzy logic*" OR "*Expert System*" OR "*Expert Systems*" OR "*Case Based Reasoning*" OR "*Case-Based Reasoning*" OR "*Rule-based System*" OR "*Rule Based System*" OR "*Rule Based Systems*" OR "*Rule Induction*" OR "*Memory-based Learning*" OR "*Instance-based Learning*" OR "*Ontology*" OR "*Ontologies*" OR "*Recommender System*" OR "*Recommender Systems*" OR "*Recommendation System*" OR "*Recommendation Systems*" OR "*Inference Engine*" OR "*Semantic Network*" OR "*Semantic Networks*" OR "*Frame Network*" OR "*Frame Networks*" OR "*Latent Semantic Analysis*" OR "*Machine Learning*" OR "*BDI*" OR "*Belief-Desire-Intention*" OR "*Natural Language Processing*" OR "*Knowledge base*"

) AND (

"*Intelligence*" OR "*Smart*" OR "*Smartness*" OR "*Clever*" OR "*Cleverness*"

)

Código 8 – String de Busca em Inglês

## A.2.4 Coleta dos Artigos

Como resultado inicial da utilização das strings nos indexadores foram encontrados 180 artigos. Foi feita então uma primeira filtragem, onde foram lidos os resumos dos artigos e aplicados os critérios de inclusão e exclusão. Nesta primeira filtragem em caso de dúvida o artigo continuava na revisão sistemática.

Após essa fase, os artigos restantes foram analisados um a um e aplicados os critérios de inclusão e exclusão.

Após todas essas etapas foram selecionados um total de 37 artigos para a composição desta revisão sistemática.:

Os artigos, bem como os resumos, são apresentados na Tabela 12.

Tabela 12 – Artigos coletados para a revisão sistemática

Artigo	Resumo
(ANAYA; BOTICARIO, 2010)	O artigo descreve e demonstra um método independente de domínio para quantificar o nível de colaboração dos alunos em um AVA. O método utiliza a técnica de Aprendizagem de Máquina Árvores de Decisão (Decision Trees) para inferir a relação entre os indicadores estatísticos e o nível de colaboração do aluno. Assim o AVA pode apresentar essas informações para o Tutor de forma a melhorar o processo de ensino-aprendizagem oferecendo assim um acompanhamento inteligente do desempenho do aluno.
(AZEVEDO; TAVARES, 2001)	O artigo relata existirem diversos softwares para educação sendo os principais os STI. Relata também a possibilidade de melhoria deles com a utilização do paradigma da colaboração resultando numa aprendizagem colaborativa. O artigo descreve então a criação de um Ambiente Inteligente para Aprendizagem Colaborativa o qual pode ser utilizado tanto em uma rede local como através da Internet utilizando um Sistema Multiagente composto de 9 agentes descritos no artigo. O STI criado é capaz de aprender automaticamente as informações relevantes sobre os usuários.
(BICA; VERDIN; VICARI, 2006)	O artigo descreve a utilização do agente MAE dentro de um ambiente de ensino e aprendizagem na Web. O agente MAE faz a captura e monitoração da Auto-Eficácia ajudando assim o aluno a regular o seu próprio processo de aprendizagem. O artigo demonstra a melhora de desempenho de um STI ao incorporar o agente MAE. A melhoria é alcançada através da utilização de técnicas de Afetividade, Lógica Fuzzy e Sistema Multiagentes dando ao sistema a capacidade de realizar um acompanhamento inteligente do desempenho do aluno.
(BITTENCOURT; COSTA, 2011)	O artigo tem o objetivo geral de propor uma solução para a construção de ambientes educacionais adaptativos e semânticos utilizando ontologias, agentes e serviços Web semânticos. Para atingir esse objetivo são então propostos 3 modelos: MASSAYO-RM (modelo conceitual), MASSAYO-CM (modelo computacional) e MASSAYO-IM (modelo de interoperabilidade). Os serviços providos são a automatização de atividades dos usuários, armazenamento e recuperação de informações além de agentes para suporte as atividades dos usuários.

continua

Tabela 12 – Artigos coletados para a revisão sistemática

continuação

Artigo	Resumo
(CHEN; ZHANG, 2002)	O artigo descreve o funcionamento e os serviços oferecidos por um ambiente colaborativo de ensino chamado EAD OnLine para a realização de cursos de EaD. A ferramenta abordada no artigo é um AVA para ensino de gestão do conhecimento construído com o uso de agentes inteligentes. O ambiente funciona como um Sistema Tutor Inteligente e disponibiliza um agente para agendamento de horários, um agente tutor responsável por responder perguntas dos alunos e um agente inteligente para filtragem de mensagens.
(CORDEIRO; BASTOS, 2006)	O artigo apresenta uma aplicação capaz de ser incorporada a um ambiente já existente para a criação de uma ferramenta de busca inteligente de informações. Para realizar o serviço a aplicação utiliza de técnicas de Web Semântica, Ontologia e Redes Neuro-Fuzzy para gerar um sistema inteligente capaz de recuperar informações para servirem de apoio em um processo de tomada de decisão do usuário do ambiente.
(DANTAS et al., 2009)	O artigo descreve a utilização de um Sistema Especialista para recomendar ações para gerenciamento de risco em projetos executados em ambientes colaborativos. O foco do artigo foi mais na parte de descrição do funcionamento das regras para tomada de decisão
(DANTAS et al., 2008)	O artigo descreve o funcionamento de uma ferramenta para auxiliar na execução de projetos através de ambiente colaborativos na Web. A ferramenta ajuda pois é capaz de acompanhar os membros do grupo quanto a sua contribuição no projeto, podendo avisar ao coordenador em caso de desvios. Portanto, a ferramenta faz um acompanhamento e avaliação automática da participação de membros em projetos colaborativos.
(DONG; LI, 2005)	O artigo propõe e descreve a integração dinâmica de Ontologias de domínio para Recuperação de informação num esforço para entregar conteúdo relevante para os aprendizes de um Ambiente Virtual de Aprendizagem. O artigo se preocupa também em demonstrar a utilização de web services para ter acesso a base de dados heterogêneas na recuperação de conteúdo relevante para os aprendizes.
(CARO; JUMÉNEZ; PATERNINA, 2012)	O artigo descreve um processo para melhorar o desempenho de sistemas de Raciocínio Baseado em Casos (RBC). A melhoria é feita através da alteração da modelagem de Julgamentos de Metamemória nos sistemas de RBC para melhorar o seu desempenho através do uso de técnicas de Metacognição, Clusterização e Filtragem Colaborativa. Como prova de conceito é desenvolvido um Sistema Tutor Inteligente (STI) capaz de adaptar o ambiente de aprendizado de acordo com o desempenho do aluno.
(FIGUEIRA et al., 2013)	O artigo apresenta a evolução e melhoria da acurácia de um sistema especialista para a correção automática de questões discursivas sobre SQL na disciplina Banco de Dados. O sistema utiliza de técnicas de Processamento de Linguagem Natural, Comparação Textual e N-Gramas para realizar a correção das questões no lugar do Tutor e apresenta acurácia satisfatória.

continua

Tabela 12 – Artigos coletados para a revisão sistemática

continuação

Artigo	Resumo
(FILHO; QUARTO; FRANCA, 2010)	O artigo descreve uma metodologia para criação de grupos sócio-afetivos em Ambientes de Aprendizagem Colaborativa. Os grupos são criados utilizando Algoritmo de Clustering e Afetividade Computacional. Os grupos criados tem como objetivo facilitar para o mediador o andamento do processo de colaboração durante as atividades em grupo.
(FREDDO et al., 2009)	O artigo propõe o desenvolvimento de uma ferramenta de apoio ao processo colaborativo e suporte às atividades de uma equipe local de pequeno porte de desenvolvimento de software, sendo o foco do trabalho proposto o processo de documentação. A ferramenta proposta resolve o problema ao criar Assistentes Pessoais para os usuários do Ambiente Colaborativo. Tais assistentes são então encarregados de intermediar as trocas de informação entre os membros do projeto, organizar a documentação, capturar e representar as operações de um membro da equipe. Assim a ferramenta ajuda a equipe a manter a documentação atualizada e melhora a cooperação no grupo.
(GATIUS; GONZ, 2011)	O artigo descreve o uso de Ontologias de Domínio em Sistemas de Diálogo na Web com o objetivo de aumentar a adaptabilidade e capacidade de colaboração desses sistemas. Para tanto, são utilizadas técnicas de Linguagem Natural na interação com o usuários e Ontologias para a recuperação de informações útil ao usuário. O objetivo é de conseguir guiar o usuário na busca de informações.
(GOULART; GI-RAFFA, 2001)	O artigo descreve a construção de um STI utilizando sistemas multiagentes para aumentar as possibilidades pedagógicas de um ambiente de ensino baseado em jogo educacional bem como e os resultados apresentados com a utilização do STI proposto.
(GUTIÉRREZ-CARREÓN; DARADOU MIS; JORBA, 2008)	O artigo propõe um modelo para composição automática de um Grid de serviços de e-Learning utilizando schemas e técnicas de matching de ontologias. O modelo utiliza as definições sintáticas e semânticas das características dos diferentes níveis dos serviços envolvidos no framework abstrato proposto pela IMS Global Learning Consortium. Seu objetivo é adaptar a interface de acordo com a necessidade e histórico dos usuários.
(IEPSEN; BERCHT; REATEGUI, 2013)	O artigo apresenta um trabalho de pesquisa para detecção de estudantes com frustração durante o processo de aprendizagem de algoritmos para poder corrigir o andamento do processo de aprendizagem. Assim o artigo investiga as possíveis relações entre os dados das interações do usuário com o ambiente e o seu estado psicológico e através de técnicas de Data Mining para aprendizagem de máquinas são geradas regras para detecção do estado de frustração. Dessa forma o ambiente pode adaptar o processo de aprendizagem ao aluno.
(KUNZEL et al., 2011)	O artigo propõe a utilização de um Agente Companheiro (AC) com emoções no auxílio ao processo de aprendizagem em um Ambiente Virtual de Aprendizagem (AVA). O AC diferencia-se de um Sistema Tutor Inteligente (STI) pois ele se apresenta como um colega virtual do estudante no ambiente. O AC tem a função de motivar e acompanhar o aluno durante a realização das tarefas. Seu objetivo final é facilitar o processo de ensino-aprendizagem.

continua

Tabela 12 – Artigos coletados para a revisão sistemática

continuação

Artigo	Resumo
(LI; MEI; WANG, 2012)	O artigo propõe auxiliar os aprendizes em E-Learning a escolherem os Objetos de Aprendizagem desejado por eles através do uso de um Sistema de Recomendação Semântico Personalizado. O Recomendador vai rastrear os logs dos usuários para aprender os objetivos de aprendizado e os hábitos dos alunos para com isso criar listas de recomendação dos Objetos de Aprendizagem apropriados para os alunos.
(LIMA; DAVID; DANTAS, 2010)	O artigo descreve as modificações realizadas numa ferramenta de detecção de riscos em um CSCW para expandir a capacidade da ferramenta para detectar riscos. Para tanto foram utilizadas técnicas de Ontologias e Contexto para substituir as técnicas de Lógica de Primeira Ordem utilizados anteriormente na ferramenta. O objetivo da ferramenta é descobrir e recomendar ações para tratamento de risco durante o gerenciamento de um projeto.
(LUGO; TACLA, 2012)	O artigo descreve uma arquitetura composta por Ontologias, Contexto e um Sistema Multiagente para a criação de um assistente pessoal capaz de representar e dar significado a dados coletados por sensores com o objetivo de apresentar sugestões de ações a um membro do ambiente de processo de software.
(FONTES et al., 2013)	O artigo descreve um método para criação de um Tutor Inteligente capaz de auxiliar os alunos no processo de ensino-aprendizagem também tenha um papel motivacional durante a aprendizagem. O Tutor é criado através da utilização de um agente pedagógico aninado com características socioafetivas e outros três agentes para detecção de estudantes passivos e recomendação de Objetos de Aprendizagem podendo se adaptar ao usuário. Assim, espera-se melhorar o método de Aprendizado Baseado em Problemas.
(MELO et al., 2013)	O artigo descreve o desenvolvimento de um módulo de software responsável por apresentar dicas à um estudante de como alcançar a melhor resposta para um exercício. O sistema auxilia os estudantes nas resoluções de exercícios, atividades em grupo e avaliações através de um feedback automático na forma de dicas.
(MAFFON et al., 2013)	O artigo apresenta um STI aplicado ao ensino da anatomia da mama feminina. O STI integra várias técnicas de IA tais como Sistema Especialista, Rede Neural, Ontologia e Sistema Hipermissão em um mesmo sistema de forma a conseguir adaptar a sua interface ao estilo do aluno.
(MARONI et al., 2011)	O artigo apresenta um framework para criação de simuladores virtuais inteligentes. O framework utiliza as técnicas de Sistema Multiagente, Redes Bayesiana e Serious Games na sua construção. A criação desses simuladores virtuais é de grande ajuda na área da saúde pois reduz custos na capacitação de profissionais além de reduzir riscos de práticas educativas clínicas. Como prova de conceito foi criado o SimDeCS (um STI para a área de saúde).

continua

Tabela 12 – Artigos coletados para a revisão sistemática

continuação

Artigo	Resumo
(MELO et al., 2012)	O artigo descreve uma proposta para reestruturar e formalizar um conteúdo em diferentes níveis de apresentação do mesmo conceito na construção de um STI conexionista. Para isso é utilizado a técnica de Rede Neural Artificial com as entradas sendo as características psicológicas, situação sócio cultural e familiarização com tecnologias do aluno. O STI reage então às escolhas do aluno mostrando o conteúdo da melhor forma possível para o aluno, facilitando assim o processo de aprendizagem.
(OLIVEIRA; TEDESCO, 2010)	O artigo descreve o modelo I-Collaboration para ser utilizado na criação de um CVA (Companheiro Virtual de Aprendizagem) visando mediar e promover a colaboração entre os usuários do ambiente através da personalização do tratamento dispensado a cada usuário.
(PESSOA; NETTO; MENEZES, 2002)	O artigo descreve o framework FamCorA para criação de aplicações CSCL. O framework utiliza as tecnologia de agentes e web services para atingir o objetivo de criar aplicações CSCL reutilizáveis parcialmente ou totalmente na elaboração de outras aplicações CSCL com o reaproveitamento de provedores de serviços já desenvolvidos.
(PRIMO; VICARI, 2011)	O artigo apresenta uma infraestrutura para criação de um Sistema de Recomendação para auxiliar professores na escolha de Objetos de Aprendizagem a serem utilizados durante um atividade de ensino-aprendizagem. O trabalho principal da pesquisa esta no raciocínio sobre as metainformações e a filtragem dos itens da lista apresentados pelo Sistema de Recomendação.
(THIRY; BARCIA, 1999)	O artigo apresenta um ambiente colaborativo, flexível e distribuído para suportar o ensino a à distância. O ambiente foi construindo com o uso da técnica de Multiagentes e Raciocínio Baseado em Casos para incorporar princípios pedagógicos no modelo de ensino.
(TOBAR et al., 2001)	O artigo propõe um modelo de arquitetura a ser seguido na construção de um ambiente colaborativo para aprendizado de programação. A ênfase é dada no ensino de programação e não em um ambiente genérico de aprendizagem ou colaboração. Na arquitetura proposta o ambiente teria um Tutor inteligente utilizando técnicas de Processamento de Linguagem Natural e Redes Neurais Artificiais para auxiliar o aluno durante o processo de aprendizagem.
(TONG; LU, 2002)	O artigo apresenta um framework para construção de ambientes CSCW utilizados por Designers. As principais características do framework são: agentes inteligentes responsáveis pelas configurações das ferramentas cooperativas de design, permite trabalho entre os Designers de modo síncrono e assíncrono, controle consistente do processo cooperativo de design e um modelo extensível de organização multimídia. A estratégia de configuração das ferramentas por um agente inteligente permite a separação do conhecimento das ferramentas do conhecimento de configuração das ferramentas, isto possibilita a diminuição da complexidade de uso das mesmas.

continua

Tabela 12 – Artigos coletados para a revisão sistemática

continuação

Artigo	Resumo
(VICENTE; VASCONCELOS; RES-TIVO, 2011)	O artigo faz um survey sobre os Sistemas Multiagente e a sua utilização em ambientes virtuais de aprendizagem citando vários autores e trabalhos. Percebe-se na maioria dos trabalhos a utilização de um SMA como um Assistente Pessoal na forma de um STI.
(WILGES et al., 2012)	O artigo descreve e demonstra a utilização de um framework para avaliação inteligente de alunos em um Ambiente Virtual de Aprendizagem (AVA). Tal framework integra técnicas de BDI e Lógica Fuzzy na sua criação para fazer o acompanhamento dos alunos.
(YUAN; CHEN, 2008)	O estudo desenvolve uma arquitetura e um mecanismo de inferência através de agentes inteligentes com o objetivo de fazer as sessões de e-brainstorming serem mais eficientes
(ZAMBIASI; RA-BELO, 2008)	O artigo descreve um modelo para criação de Assistentes de Software Virtual (ASV) com o objetivo de automatizar atividades de responsabilidade dos usuários do ambiente. Os ASV seriam criados utilizando a técnica de Sistema Multiagente aonde existiria uma agente no ambiente para cada usuário do ambiente além de outros agentes necessários para o funcionamento dos ASVs.

Fonte: Autor.

### A.3 Análise dos Resultados

Após a leitura e análise dos artigos foram feitas algumas classificações dos artigos com o objetivo de melhorar o estudo dos resultados. Assim os artigos foram classificados em grupos, descritos a seguir.

#### A.3.1 CSCW x CSCL

Nesta classificação os artigos foram separados dependendo se o contexto do artigo é um sistema de trabalho (CSCW) ou de aprendizagem (CSCL) colaborativo apoiado por computador. O resultado pode ser visto na Tabela 13

Um fato para chamar a atenção foi a concentração do desenvolvimento de Serviços Inteligentes em CSCL. Tal fato chama a atenção para a possibilidade da aplicação dos serviços desenvolvidos especificamente para CSCL serem tornados genéricos e aplicados aos CSCW, uma vez que segundo (BORGES; CAVALCANTI; CAMPOS, 1995), o primeiro é um subconjunto do último.

Tabela 13 – Artigos CSCW x CSCL

Tipo	Artigos	Qtd	Pct
CSCW	(CORDEIRO; BASTOS, 2006; DANTAS et al., 2009; DANTAS et al., 2008; FREDDO et al., 2009; GATIUS; GONZ, 2011; LIMA; DAVID; DANTAS, 2010; LUGO; TACLA, 2012; TONG; LU, 2002; YUAN; CHEN, 2008; ZAMBIASI; RABELO, 2008)	10	27,03%
CSCL	(ANAYA; BOTICARIO, 2010; AZEVEDO; TAVARES, 2001; BICA; VERDIN; VICARI, 2006; BITTENCOURT; COSTA, 2011; CHEN; ZHANG, 2002; CONCEIÇÃO, 2000; DONG; LI, 2005; CARO; JUMÉNEZ; PATERNINA, 2012; FIGUEIRA et al., 2013; FILHO; QUARTO; FRANCA, 2010; GOULART; GIRAFFA, 2001; GUTIÉRREZ-CARREÓN; DARADOUMIS; JORBA, 2008; IEPSEN; BERCHT; REATEGUI, 2013; KUNZEL et al., 2011; LI; MEI; WANG, 2012; FONTES et al., 2013; MELO et al., 2013; MAFFON et al., 2013; MARONI et al., 2011; MELO et al., 2012; OLIVEIRA; TEDESCO, 2010; PESSOA; NETTO; MENEZES, 2002; PRIMO; VICARI, 2011; THIRY; BARCIA, 1999; TOBAR et al., 2001; VICENTE; VASCONCELOS; RESTIVO, 2011; WILGES et al., 2012)	27	72,97%

Fonte: Autor.

### A.3.2 Técnicas de IA Utilizadas

As técnicas de IA implementadas nos artigos da revisão sistemática estão elencadas na Tabela 14.

Tabela 14 – Artigos por técnicas de IA utilizadas

Técnica	Artigos	Qtd	Pct
Sistemas Multiagente	(AZEVEDO; TAVARES, 2001; BICA; VERDIN; VICARI, 2006; BITTENCOURT; COSTA, 2011; CHEN; ZHANG, 2002; CONCEIÇÃO, 2000; CORDEIRO; BASTOS, 2006; FREDDO et al., 2009; GOULART; GIRAFFA, 2001; KUNZEL et al., 2011; LUGO; TACLA, 2012; FONTES et al., 2013; MARONI et al., 2011; OLIVEIRA; TEDESCO, 2010; PESSOA; NETTO; MENEZES, 2002; THIRY; BARCIA, 1999; TONG; LU, 2002; VICENTE; VASCONCELOS; RESTIVO, 2011; WILGES et al., 2012; YUAN; CHEN, 2008; ZAMBIASI; RABELO, 2008)	20	54,40%
Ontologias	(BITTENCOURT; COSTA, 2011; CORDEIRO; BASTOS, 2006; DONG; LI, 2005; GATIUS; GONZ, 2011; GUTIÉRREZ-CARREÓN; DARADOUMIS; JORBA, 2008; LI; MEI; WANG, 2012; LIMA; DAVID; DANTAS, 2010; LUGO; TACLA, 2012; MAFFON et al., 2013; PRIMO; VICARI, 2011; TOBAR et al., 2001; YUAN; CHEN, 2008)	12	32,43%
Afetividade Computacional	(BICA; VERDIN; VICARI, 2006; CHEN; ZHANG, 2002; FILHO; QUARTO; FRANCA, 2010; IEPSEN; BERCHT; REATEGUI, 2013; KUNZEL et al., 2011; FONTES et al., 2013; OLIVEIRA; TEDESCO, 2010)	7	18,92%
Clusterização	(CHEN; ZHANG, 2002; CARO; JUMÉNEZ; PATERNINA, 2012; FILHO; QUARTO; FRANCA, 2010)	3	8,11%

continua

Tabela 14 – Classificação pelas técnicas de IA utilizadas

continuação

Técnica	Artigos	Qtd	Pct
Contexto Computacional	(LIMA; DAVID; DANTAS, 2010; LUGO; TACLA, 2012; OLIVEIRA; TEDESCO, 2010)	3	8,11%
Lógica Fuzzy	(BICA; VERDIN; VICARI, 2006; MELO et al., 2013; WILGES et al., 2012)	3	8,11%
Mineração de Dados	(BITTENCOURT; COSTA, 2011; IEPSSEN; BERCHT; REATEGUI, 2013; MELO et al., 2013)	3	8,11%
Processamento de Linguagem Natural	(FIGUEIRA et al., 2013; GATIUS; GONZ, 2011; TOBAR et al., 2001)	3	8,11%
Raciocínio Baseado em Casos	(BITTENCOURT; COSTA, 2011; CARO; JUMÉNEZ; PATERNINA, 2012; THIRY; BARCIA, 1999)	3	8,11%
Rede Neural Artificial	(MAFFON et al., 2013; MELO et al., 2012; TOBAR et al., 2001)	3	8,11%
Aprendizagem por Reforço	(GOULART; GIRAFFA, 2001; YUAN; CHEN, 2008)	2	5,40%
Filtragem Colaborativa	(CARO; JUMÉNEZ; PATERNINA, 2012; PRIMO; VICARI, 2011)	2	5,40%
Lógica de Primeira Ordem	(DANTAS et al., 2009; DANTAS et al., 2008)	2	5,40%
N-Gramas	(FIGUEIRA et al., 2013; MELO et al., 2013)	2	5,40%
Semântica Latente	(PESSOA; NETTO; MENEZES, 2002; TOBAR et al., 2001)	2	5,40%
Serious Game	(MARONI et al., 2011; VICENTE; VASCONCELOS; RESTIVO, 2011)	2	5,40%
Algoritmo Genético	(VICENTE; VASCONCELOS; RESTIVO, 2011)	1	2,70%
Árvores de Decisão	(ANAYA; BOTICARIO, 2010)	1	2,70%
Metacognição	(CARO; JUMÉNEZ; PATERNINA, 2012)	1	2,70%
Raciocínio Baseado em Memória	(PESSOA; NETTO; MENEZES, 2002)	1	2,70%
Rede Bayesiana	(MARONI et al., 2011)	1	2,70%
Rede Neuro-Fuzzy	(CORDEIRO; BASTOS, 2006)	1	2,70%
Rede Semântica	(DANTAS et al., 2008)	1	2,70%
Regras de Produção	(THIRY; BARCIA, 1999)	1	2,70%
Sistema Hipermedia Adaptativo	(MAFFON et al., 2013)	1	2,70%
Web Semântica	(CORDEIRO; BASTOS, 2006)	1	2,70%

Fonte: Autor.

Aqui a concentração de pesquisas realizadas em Sistemas Multiagente (SMAs) pode ser explicada devido à diversidade e complexidade dos problemas tratados, pois com a utilização de agentes para modelar a resolução do problema é possível dividir o mesmo em vários problemas menores. Estes problemas podem então ser resolvidos individualmente sem afetar a resolução dos outros.

Além disso, é importante notar que em muitos casos os SMAs foram utilizados em conjunto com outras técnicas de IA para resolução do problema.

### A.3.3 Serviços Inteligentes

Durante a análise dos artigos da revisão sistemática foi percebida uma preferência por acrescentar Serviços Inteligentes ao ambiente na forma de uma melhoria de alguma atividade executada por algum usuário do ambiente. Em outras palavras, as melhorias são intervenções pontuais visando aumento de produtividade na interação entre o ambiente e os seus usuários.

Outro ponto notado foi a preferência por implementar essas melhorias nos ambientes como algum tipo de Software Assistente Virtual (SAV) executando uma função específica, podendo interagir com o usuário ou, em alguns casos, até mesmo chegando a substituir os usuários na execução de algumas tarefas.

O potencial de tarefas possíveis de serem executadas em um ambiente CSCW é grande e fortemente dependente dos objetivos do ambiente. Por exemplo, uma boa parte das tarefas executadas em um CSCW com objetivo de se escreverem livros didáticos são diferentes das tarefas a serem executadas em um CSCW com objetivo de gerenciar o andamento de projetos. No entanto, ainda assim existem algumas tarefas comuns em todos os ambientes CSCW, como por exemplo tarefas relacionadas ao gerenciamento de agendas e negociação de reuniões.

De modo a realizar uma análise apropriada, foi necessário adotar ou, na impossibilidade de tal, conceber uma forma de classificar os serviços em diferentes tipos genéricos de Serviços Inteligentes fornecidos.

#### A.3.3.1 Classes de Serviços Inteligentes em Ambientes Colaborativos na Web

Para a criação de classes genéricas de serviços inteligentes foi utilizado como ponto de partida a lista de seis classes de tarefas realizadas por um SAV em um ambiente colaborativo conforme citado em (ZAMBIASI; RABELO, 2008): classe 1 - substituição do usuário em determinadas tarefas; classe 2 - gerenciamento da agenda e atividades do usuário; classe 3 - procura de informações em repositórios; classe 4 - auxílio ao usuário a compreender que tarefas precisam ser executadas e como elas seriam executadas; classe 5 - execução de algumas tarefas do usuário; e classe 6 - filtro e gerenciamento das informações provindas de redes ubíquas.

Em uma tentativa preliminar de classificar os serviços dos artigos da revisão sistemática nessas seis classes, foram encontradas dificuldades no enquadramento de algumas tarefas. Por exemplo, em alguns casos nem sempre as informações provêm de redes ubíquas e, para serviços responsáveis por personalizar o ambiente para o seu usuário, não era possível classificá-los nas seis classes existentes. Tais problemas foram superados com a alteração da descrição da classe 6, que passou a ser: classe 6 – filtro e gerenciamento das informações provindas de fontes externas. Criamos também uma nova classe: classe 7 – personalização automática do ambiente ao usuário.

Assim, na Tabela 15 são mostradas as 7 classes de Serviço Inteligentes bem como a porcentagem de artigos da revisão sistemática onde é percebido um exemplo do Serviço Inteligente daquela categoria. O somatório das porcentagens é maior que 100% pois muitos trabalhos apresentaram mais de um tipo de serviço inteligente.

Percebe-se uma forte concentração nas pesquisas onde o usuário é auxiliado na compreensão da escolha de quais tarefas devem ser executadas e como elas devem ser executadas. Ou seja, a concentração é na criação de Serviços Inteligentes para auxiliar no trabalho realizado no ambiente em questão seja ele um CSCW ou um CSCL.

Essa conclusão parece ser natural pois podemos ver como um objetivo tanto dos

Tabela 15 – Artigos por Tipos de Serviços Inteligentes

Classe	Artigos	Qtd	%
Auxílio ao usuário a compreender que tarefas precisam ser executadas e como	(ANAYA; BOTICARIO, 2010) (AZEVEDO; TAVARES, 2001) (BICA; VERDIN; VICARI, 2006) (BITTENCOURT; COSTA, 2011) (CHEN; ZHANG, 2002) (CONCEIÇÃO, 2000) (DANTAS et al., 2009) (DANTAS et al., 2008) (CARO; JUMÉNEZ; PATERNINA, 2012) (FREDDO et al., 2009) (GATIUS; GONZ, 2011) (GOULART; GIRAFFA, 2001) (IEPSEN; BERCHT; REATEGUI, 2013) (KUNZEL et al., 2011) (LI; MEI; WANG, 2012) (LIMA; DAVID; DANTAS, 2010) (LUGO; TACLA, 2012) (FONTES et al., 2013) (MELO et al., 2013) (MAFFON et al., 2013) (MARONI et al., 2011) (MELO et al., 2012) (OLIVEIRA; TEDESCO, 2010) (PESSOA; NETTO; MENEZES, 2002) (PRIMO; VICARI, 2011) (THIRY; BARCIA, 1999) (TOBAR et al., 2001) (VICENTE; VASCONCELOS; RESTIVO, 2011) (WILGES et al., 2012) (YUAN; CHEN, 2008) (ZAMBIASI; RABELO, 2008)	31	83,78%
Procura de informações em repositórios	(BITTENCOURT; COSTA, 2011) (CORDEIRO; BASTOS, 2006) (DONG; LI, 2005) (GATIUS; GONZ, 2011) (GUTIÉRREZ-CARREÓN; DARADOUMIS; JORBA, 2008) (LI; MEI; WANG, 2012) (PESSOA; NETTO; MENEZES, 2002) (PRIMO; VICARI, 2011) (VICENTE; VASCONCELOS; RESTIVO, 2011) (ZAMBIASI; RABELO, 2008)	10	27,03%
Personalização automática do ambiente ao usuário	(CHEN; ZHANG, 2002) (CARO; JUMÉNEZ; PATERNINA, 2012) (GUTIÉRREZ-CARREÓN; DARADOUMIS; JORBA, 2008) (IEPSEN; BERCHT; REATEGUI, 2013) (FONTES et al., 2013) (MAFFON et al., 2013) (MELO et al., 2012) (OLIVEIRA; TEDESCO, 2010) (TONG; LU, 2002) (VICENTE; VASCONCELOS; RESTIVO, 2011)	10	27,03%
Substituição do usuário em determinadas tarefas	(ANAYA; BOTICARIO, 2010) (FIGUEIRA et al., 2013) (FILHO; QUARTO; FRANCA, 2010) (TONG; LU, 2002) (VICENTE; VASCONCELOS; RESTIVO, 2011) (WILGES et al., 2012) (YUAN; CHEN, 2008)	7	18,91%
Execução de algumas tarefas do usuário	(BITTENCOURT; COSTA, 2011) (FIGUEIRA et al., 2013) (FILHO; QUARTO; FRANCA, 2010) (FREDDO et al., 2009) (TONG; LU, 2002) (VICENTE; VASCONCELOS; RESTIVO, 2011) (ZAMBIASI; RABELO, 2008)	7	18,91%
Filtro e gerenciamento das informações providas de fontes externas	(CONCEIÇÃO, 2000) (GUTIÉRREZ-CARREÓN; DARADOUMIS; JORBA, 2008) (ZAMBIASI; RABELO, 2008)	3	8,11%
Gerenciamento da agenda e atividades do usuário	(CONCEIÇÃO, 2000) (ZAMBIASI; RABELO, 2008)	2	5,40%

Fonte: Autor.

CSCW como dos CSCL fazer com que os seus usuários executem tarefas dentro do ambiente para se atingir algum objetivo previamente estipulado.

### A.3.4 Formas de Auxílio ao Usuário com Serviços Inteligentes

Como mostrado anteriormente, os artigos da revisão sistemática tiveram uma grande concentração na área de serviços para auxiliar os usuários na identificação de tarefas a serem executadas e em como executar essas tarefas.

Com o objetivo de conhecer melhor essa classe de serviços inteligentes, foi verificado quais eram os tipos de sistemas utilizados nos artigos para se implementar tais serviços.

Revisando os artigos foi possível notar a utilização de três tipos de sistemas para implementar os serviços e um quarto tipo que é uma especialização de um dos tipos já existentes:

- **Sistema de Recomendação.** Quando o serviço apresenta ao usuário recomendações de ações a serem feitas ou opções possíveis para alguma atividade, porém ele não informa uma ordem de preferência ou ranqueamento. Ou seja, o usuário não recebe uma explicação sobre por que escolher uma opção em detrimento de outra.
- **Sistema de Apoio a Decisão.** Quando o serviço apresenta ao usuário ações a serem feitas além de apresentar uma ordem de preferência ou ranqueamento. Aqui o usuário além de ter uma explicação ou pelo menos uma indicação de por que escolher uma opção em vez de outra ele continua tendo a liberdade de não escolher a melhor sugestão.
- **Sistema Especialista.** Quando o serviço oferece uma função ou opção no ambiente a qual normalmente seria preciso a intervenção de um especialista para poder ocorrer.
  - **Sistema Tutor Inteligente (STI).** Um Sistema Especialista com objetivo de ensinar ao usuário um assunto ou matéria específica. Deve apresentar os módulos: estudante (conhecimento sobre o estudante), domínio (conhecimento sobre o assunto), pedagógico (metodologias de ensino), especialista (acompanha o desenvolvimento dos alunos e ajusta o processo de aprendizagem) e comunicação.

Assim na Tabela 16 temos a distribuição das formas de auxílio ao usuário através de serviços inteligentes que foram encontrados durante a revisão sistemática.

Percebe-se uma tendência para implementação desse tipo de serviço (com objetivo de auxiliar os usuários na execução de tarefas no ambiente CSCW) através da captura do conhecimento de um especialista em um assunto importante para as atividades a serem exercidas no ambiente específico em questão.

Tabela 16 – Artigos por Formas de Auxílio ao Usuário com Serviços Inteligentes

Tipo	Artigos	Qtd	%
Sistema de Recomendação	(BICA; VERDIN; VICARI, 2006) (GATIUS; GONZ, 2011) (LI; MEI; WANG, 2012) (PRIMO; VICARI, 2011) (YUAN; CHEN, 2008)	5	13,89%
Sistema de Apoio à Decisão	(BITTENCOURT; COSTA, 2011) (DANTAS et al., 2009) (LIMA; DAVID; DANTAS, 2010) (LUGO; TACLA, 2012) (THIRY; BARCIA, 1999)	5	13,89%
Sistema Especialista - Sistema Tutor Inteligente	(AZEVEDO; TAVARES, 2001) (BICA; VERDIN; VICARI, 2006) (CONCEIÇÃO, 2000) (CARO; JUMÉNEZ; PATERNINA, 2012) (GOULART; GIRAFFA, 2001) (FONTES et al., 2013) (MELO et al., 2013) (MAFFON et al., 2013) (MARONI et al., 2011) (MELO et al., 2012) (THIRY; BARCIA, 1999) (TOBAR et al., 2001) (VICENTE; VASCONCELOS; RESTIVO, 2011) (WILGES et al., 2012)	14	38,89%
Sistema Especialista - Outros Tipos	(ANAYA; BOTICARIO, 2010) (BITTENCOURT; COSTA, 2011) (CHEN; ZHANG, 2002) (DANTAS et al., 2008) (FIGUEIRA et al., 2013) (FREDDO et al., 2009) (IEPSEN; BERCHT; REATEGUI, 2013) (KUNZEL et al., 2011) (OLIVEIRA; TEDESCO, 2010) (PESSOA; NETTO; MENEZES, 2002) (YUAN; CHEN, 2008) (ZAMBIASI; RABELO, 2008)	12	33,33%

Fonte: Autor.

As informações sobre classes de serviços inteligentes e as formas mais utilizadas para auxílio ao usuário serviram de inspiração para a tarefa de planejar um *framework* que fornece serviços inteligentes em Ambientes Colaborativos Flexíveis. A proposta deste *framework* é apresentada no próximo capítulo.

## A.4 Discussão sobre os Resultados

Apesar da quantidade de pesquisas em andamento, foram identificadas dois pontos aonde as pesquisas estão tendo pouca atenção, sendo assim áreas interessantes para se fazer novas pesquisas.

O primeiro ponto é a ausência nos artigos analisados da possibilidade de um usuário de um Ambiente Colaborativo ser um usuário com necessidades especiais. Pesquisas nesta área podem trazer uma mudança de foco, a qual pode acabar auxiliando todos os usuários.

O segundo ponto é não ter sido citado ou proposto nos artigos analisados um *framework* para a construção de Serviços Inteligentes em Ambientes Colaborativos. Apesar da grande concentração de pesquisa nesta área ainda não foi feita uma consolidação e agregação dos esforços para a criação de um *framework* genérico para a criação de Serviços Inteligentes. Tal ferramenta possibilitaria aos pesquisadores concentrarem o foco na resolução de problemas através dos Assistentes Pessoais e não na construção dos mesmos.

Focando neste segundo ponto discute-se a seguir uma proposta para a criação desse

framework.

#### A.4.1 Um Framework de Serviços Inteligentes para CSCW/CSCL

Neste trabalho foram identificadas sete classes de Serviços Inteligentes fornecidos em um ambiente CSCW. Desta forma, um framework para a criação de Serviços Inteligentes deve contemplar essas sete classes, bem como não impedir a existência de mais de um serviço oferecido por classe.

A partir do estudo dessa revisão sistemática foi possível detectar a necessidade da classe 4 (auxílio ao usuário a compreender que tarefas precisam ser executadas e como) utilizar sistemas especialistas, sistemas de recomendação e sistemas de apoio a decisão. Assim seria útil a existência de modelos genéricos desses três tipos de sistemas para um possível aproveitamento no ambiente CSCW.

Devido à complexidade de tal framework é proposto a utilização da técnica de SMA para a sua implementação. Levando em consideração que a maioria dos artigos dessa revisão utilizou-se de SMAs obtendo bons resultados, espera-se reproduzir esse sucesso no framework proposto.

A proposta inicial para o framework é apresentada no seguinte esboço:

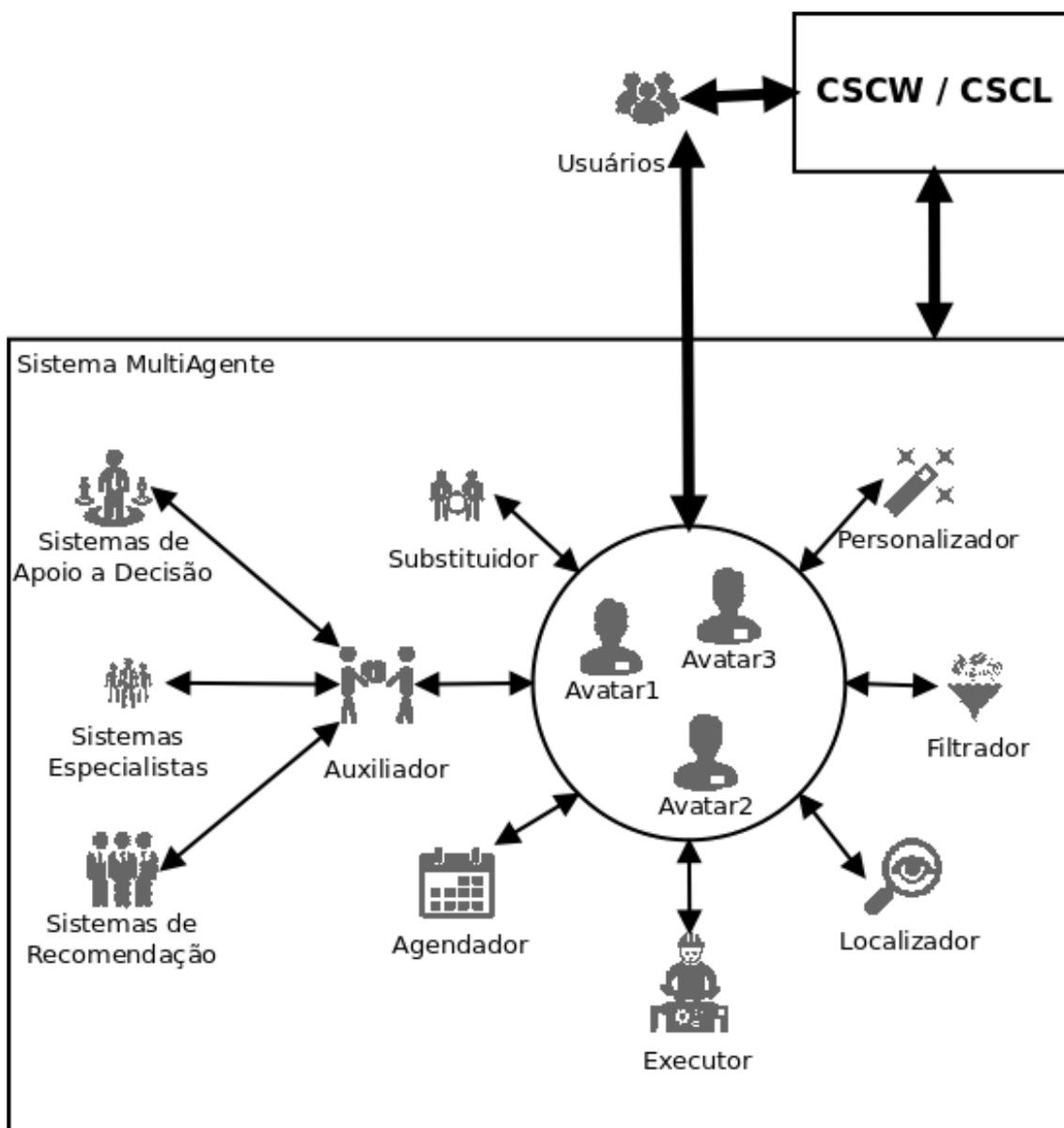
No SMA cada usuário vai ter um agente avatar representando o usuário dentro do SMA. A interação do usuário com o SMA se dá através do seu avatar, o qual vai funcionar como um orquestrador para as operações do SMA para aquele usuário. Para o usuário, o seu avatar vai se comportar como sendo um assistente pessoal inteligente.

Para cada uma das sete classes de Serviços Inteligentes para Ambientes Colaborativos opta-se por ter um agente orquestrador daquele tipo de serviço. Assim, para a classe 1 temos o agente Substituidor, para a classe 2 temos o agente Agendador, para a classe 3 temos o agente Localizador, para a classe 4 temos o agente Auxiliador, para a classe 5 temos o agente Executor, para a classe 6 temos o agente Filtrador, e para a classe 7 temos o agente Personalizador.

A opção de utilizar um agente orquestrador (avatar do usuário) para solicitar serviços para outros agentes orquestradores (das sete classes de serviço) é possibilitar o desenvolvimento de um framework genérico aonde possa ser possível expandi-lo ou escolher quais partes são interessantes a um ambiente CSCW específico.

Cada agente orquestrador das tarefas de cada uma das classes de serviço teria acesso a outros agentes encarregados de executarem as tarefas necessárias. No exemplo dado, o agente orquestrador Auxiliador pode requisitar os serviços de outros agentes responsáveis por implementam Sistemas de Apoio a Decisão, Sistemas Especialistas e Sistemas de Recomendação. Assim poderiam ser especificados modelos genéricos de agentes

Figura 36 – Esboço do framework.



Fonte: Autor.

para resolver problemas pontuais e eles poderiam ser integrados na solução através de um dos agentes orquestradores.

## A.5 Conclusão

Esse artigo apresenta uma revisão sistemática sobre Serviços Inteligentes oferecidos em ambientes CSCW e CSCL. Como observações diretas da revisão sistemática temos:

- Foi verificada a existência de uma concentração significativamente maior de Serviços Inteligentes sendo desenvolvidos para ambientes CSCL em relação a CSCW em geral, podendo isto indicar possíveis aplicações da descobertas em CSCL para CSCW;
- Foi verificado que, dentre as técnicas de IA utilizadas na criação de Serviços Inteligentes, a mais comum é a de SMA, sendo usada em 54% dos artigos analisados. Sendo então uma boa indicação da melhor forma para se criar Serviços Inteligentes ser através do uso de SMAs;
- Foi criada e apresentada uma classificação para os tipos de Serviços Inteligentes fornecidos em ambientes CSCW, sendo apresentadas sete classes principais;
- Foi verificado como forma mais comum de criação de um Serviço Inteligente a implementação de um Sistema Especialista capaz de reproduzir o comportamento de um especialista de uma área importante para o ambiente CSCW específico aonde o Serviço Inteligente vai ser criado.

Da análise dos resultados da revisão sistemática resultaram as seguintes conclusões:

- Não foram encontrados estudos para a criação de Serviços Inteligentes em Ambientes Colaborativos aonde os usuários tenham necessidades especiais;
- Foi notada a falta de uma proposta de framework para criação de Serviços Inteligentes em CSCW;
- Foi apresentado um esboço de um possível framework baseado em SMA para a criação de Serviços Inteligentes em Ambientes Colaborativos.

As próximas atividades programadas são um maior estudo do framework e realizar uma proposta mais detalhada para o seu funcionamento.

# APÊNDICE B – Lista de Tarefas Sugeridas para Terceirização

Nesta parte descrevemos o processo da criação de uma lista de tarefas sugeridas para terceirização que foi utilizada para verificar se a classificação para serviços inteligentes desenvolvida durante este projeto é uma classificação adequada.

A lista de tarefas foi construída apenas com a descrição de possíveis trabalhos executados por um Assistente Humano Virtual (AHV). Para o nosso experimento imaginamos o cenário aonde esses serviços fossem executados por um Assistente Software Virtual (ASV), assim, o cliente demandante da tarefa não saberia dizer se quem executou de fato a tarefa foi um AHV ou um ASV. O objetivo era validar que a proposta de classificação de serviços inteligentes utilizada durante o projeto é consistente.

## B.1 Procedimento

Atualmente, as principais tarefas terceirizadas para um AHV geralmente são do campo da administração, no entanto existem AHV especialistas em muitos campos tais como: marketing, desenvolvimento de sítios web, turismo, etc. Portanto a nossa lista de tarefas sugeridas para serem terceirizadas para um AHV deve englobar tarefas que fazem sentido para qualquer contexto de trabalho e tarefas que fazem sentido apenas em alguns contextos de trabalho. Assim montamos um procedimento que tem o cuidado de englobar tanto a terceirização de tarefas genéricas como de tarefas específicas.

Portanto, para tentar garantir que a lista de tarefas aqui construída abrange-se várias áreas de domínios diferentes, foram seguidos os passos descritos abaixo:

1. Pesquisa no Google com as palavras-chaves: classification, task, virtual e assistant;
2. Descartados resultados que não fossem relacionados com terceirização de serviços para um AHV;
3. Coleta das listas que aparecem nos 10 melhores resultados;
4. Junção de todas as listas em uma única lista;
5. Revisão da lista única.
  - a) Retirar tarefas repetidas;
  - b) Agrupar tarefas semelhantes sobre uma tarefa mais genérica.

Após o passo 4 a lista única continha 125 tarefas e após o último passo a lista ficou com 74 tarefas.

## B.2 Lista de Tarefas

Nesta seção apresentamos a versão final da lista em inglês para não ser perdido o sentido original das tarefas:

1. Adding tags and images to website and blog posts;
2. Answering support tickets;
3. Article and blog post creation;
4. Bookkeeping and payroll duties: calculating hours, adding expenses, updating salaries;
5. Checking email, responding to customer inquiries and managing spam;
6. Collecting documents for tax season; perhaps complete business taxes too;
7. Coming up with detailed profiles and inserting links to company website;
8. Composing documents from handwritten drafts, faxes and dictations;
9. Composing press releases and newsletters and submitting to news release directories;
10. Conducting a keyword research for website and performing a blog analysis;
11. Conducting background, credit and criminal checks on staffers;
12. Converting, merging and splitting .PDF files;
13. Creating and sending out greeting cards, invitations, newsletters and thank you notes;
14. Creating images for your social media posts, your podcast episodes and your blog posts;
15. Creating new list of email contacts, email newsletters and promotional copy;
16. Creating online forms for content submission, customer feedback or inquiries;
17. Creating, filing and presenting basic reports (reports on weekly tasks, deliverables, sales);
18. Database building, entry and updates (sales, contacts, updating email or contact lists on your CRM, etc.);

19. Designing logos, ebook covers, headers, icons and other graphic elements;
20. Developing and delivering slideshow presentations;
21. Developing, updating and optimizing an SEO and web marketing strategy;
22. Doing regular backups to prevent data loss;
23. Editing audio files by removing background noise and improving volume levels;
24. Engaging with audience: responding to inquiries, sharing relevant information, thanking customers for mentions and purchases and posting promotions;
25. Establishing follow-up emails and autoresponders and edit according to response rate;
26. Establishing, updating and managing a calendar of important events;
27. File Management;
28. Filter and reply to comments on your blog;
29. Finding appropriate, high-traffic websites to place advertisements;
30. Gathering data / research from your industry, or even from your own content;
31. Generating listicles (list articles) on industry-related matters;
32. Helping moderate your Forum;
33. Hotel and Flight Booking;
34. Interviewing job applicants and speaking with references;
35. Interviewing previous customers to compose case studies;
36. Liaison between you and other team members;
37. Looking for a content marketing firm or publisher to post branded content;
38. Manage and update Social Media Accounts;
39. Managing Social Media groups you run;
40. Monitoring weekly and monthly Google Analytics reports; observing site traffic;
41. Observing the company's competitors on social media by looking at rankings, online visibility and keyword prioritization;
42. Performing a social media audit: conduct thorough analysis on traffic, shares and mentions;

43. Performing banking needs, like paying bills and transferring funds;
44. Performing generic errands for the office, including buying items online, arranging locations for office parties and hiring a cleaning service;
45. Preparing training manuals for new staff members or remote workers;
46. Producing content marketing material, such as infographics, white papers and ebooks;
47. Producing customer care scripts for customer service requests;
48. Producing graphs from your spreadsheets;
49. Proofreading documents and other office materials;
50. Publishing extensive how-to guides and industry-related book reviews;
51. Putting together welcome and goodbye packages for both clients and staff;
52. Reach out to media outlets and influencers to garner profiles by publishers;
53. Receptionist duties: answering calls, leaving voicemails and checking messages;
54. Recording, editing and setting up podcasts and inserting them onto webpage;
55. Recruiting for potential team members and contractors or freelancers;
56. Researching key hashtag conversations of the day and find out if it fits in with company's messages and marketing objectives;
57. Researching on important data, statistics and facts for meetings, presentations or blogs;
58. Run a social media contest or challenge;
59. Running an internal office or challenge so employees can receive bonuses;
60. Scheduling social media posts;
61. Searching for and contacting industry experts or guests to participate in podcasts and webinars;
62. Setting up, monitoring and managing other affiliates and respective links;
63. Speaking with customer service representatives for tech support, banking issues, etc;
64. Splicing videos intros and outros and inputting graphics and music;
65. Starting an in-depth competitor analysis (targeted keywords, ranked content, SEM-RUSH positions);

66. Supporting payment gateway and ticketing systems;
67. Taking care of customer refunds;
68. Training on-site employees, virtual staff members or freelancers;
69. Updating membership sites with content you provide;
70. Uploading files to Social Medias;
71. Writing and sending invoices to clients;
72. Writing and submitting op-ed pieces to newspapers and websites;
73. Writing down minutes from meetings or events and then creating a detailed document;
74. Writing, editing and sharing posts on social networks.

### B.3 Classificação das Tarefas

A seguir foi feita uma separação das tarefas encontradas dentro das classes de serviço inteligentes. Ou seja, para cada tarefa sugerida como uma possibilidade de terceirização para um AHV foi verificado em qual classe de serviço inteligente esse serviço seria enquadrado caso fosse feito por um ASV. O resumo desta classificação pode ser visto na Tabela 17.

Nas próximas seções vamos descrever a classe e listar as tarefas que foram classificadas nela.

Tabela 17 – Total de tarefas encontradas por classe

Classe	Qtd	Pct
Auxílio ao usuário	12	16,22%
Execução de tarefas do usuários	7	9,46%
Filtro e gerenciamento das informações providas de fontes externas	4	5,40%
Gerenciamento da agenda	2	2,70%
Personalização automática do ambiente ao usuário	0	0%
Procura de informações em repositórios	14	18,92%
Substituição do Usuário	35	47,30%

Fonte: Autor.

#### B.3.1 Substituição do Usuário

Aqui entram os serviços que substituem os usuários na execução de alguma tarefa. Os serviços devem não apenas realizar as tarefas mas como também devem personificar o usuário, assim caso durante a execução do serviço seja necessário interagir com outros usuários, estes podem não saber se no momento em que a tarefa é realizada é o próprio usuário ou sistema quem está interagindo com ele. Segue a lista de tarefas classificadas nesta classe:

- 
- Adding tags and images to website and blog posts;
  - Article and blog post creation;
  - Creating images for your social media posts, your podcast episodes and your blog posts;
  - Creating online forms for content submission, customer feedback or inquiries;
  - Database building, entry and updates (sales, contacts, updating email or contact lists on your CRM, etc.);
  - Designing logos, ebook covers, headers, icons and other graphic elements;
  - Engaging with audience: responding to inquiries, sharing relevant information, thanking customers for mentions and purchases and posting promotions;
  - Helping moderate your Forum;
  - Hotel and Flight Booking;
  - Interviewing job applicants and speaking with references;
  - Interviewing previous customers to compose case studies;
  - Manage and update Social Media Accounts;
  - Managing Social Media groups you run;
  - Performing banking needs, like paying bills and transferring funds;
  - Performing generic errands for the office, including buying items online, arranging locations for office parties and hiring a cleaning service;
  - Preparing training manuals for new staff members or remote workers;
  - Producing content marketing material, such as infographics, white papers and ebooks;
  - Producing customer care scripts for customer service requests;
  - Publishing extensive how-to guides and industry-related book reviews;
  - Putting together welcome and goodbye packages for both clients and staff;
  - Recording, editing and setting up podcasts and inserting them onto webpage;
  - Recruiting for potential team members and contractors or freelancers;
  - Run a social media contest or challenge;

- Running an internal office or challenge so employees can receive bonuses;
- Speaking with customer service representatives for tech support, banking issues, etc;
- Splicing videos intros and outros and inputting graphics and music;
- Supporting payment gateway and ticketing systems;
- Taking care of customer refunds;
- Training on-site employees, virtual staff members or freelancers;
- Updating membership sites with content you provide;
- Uploading files to Social Medias;
- Writing and sending invoices to clients;
- Writing and submitting op-ed pieces to newspapers and websites;
- Writing, editing and sharing posts on social networks;
- Composing press releases and newsletters and submitting to news release directories.

### B.3.2 Gerenciamento da agenda

Aqui ficam os serviços responsáveis pela criação e manutenção da agenda do usuário e/ou do grupo incluindo tarefas tais como: agendamento das tarefas, reuniões e lembretes para o usuário. O próprio ambiente pode realizar agendamentos automáticos ou sugerir reuniões e horários aos usuários. Ele pode facilitar a negociação de horários entre os envolvidos. Segue a lista de tarefas classificadas nesta classe:

- Establishing, updating and managing a calendar of important events;
- Scheduling social media posts.

### B.3.3 Procura de informações em repositórios

Aqui ficam os serviços que precisam interagir com repositórios de informações externos ao ambiente para obter informações necessárias ao seu funcionamento ou devido a solicitação do usuário. É desejável que os serviços tenham uma certa iniciativa de quando e aonde devam buscar informações, não sendo sempre inicializados devido a demanda explícita do usuário. Segue a lista de tarefas classificadas nesta classe:

- Conducting a keyword research for website and performing a blog analysis;

- Conducting background, credit and criminal checks on staffers;
- Finding appropriate, high-traffic websites to place advertisements;
- Gathering data / research from your industry, or even from your own content;
- Looking for a content marketing firm or publisher to post branded content;
- Monitoring weekly and monthly Google Analytics reports; observing site traffic;
- Observing the company's competitors on social media by looking at rankings, online visibility and keyword prioritization;
- Performing a social media audit: conduct thorough analysis on traffic, shares and mentions;
- Reach out to media outlets and influencers to garner profiles by publishers;
- Researching key hashtag conversations of the day and find out if it fits in with company's messages and marketing objectives;
- Researching on important data, statistics and facts for meetings, presentations or blogs;
- Searching for and contacting industry experts or guests to participate in podcasts and webinars;
- Setting up, monitoring and managing other affiliates and respective links;
- Starting an in-depth competitor analysis (targeted keywords, ranked content, SEM-RUSH positions).

### B.3.4 Auxílio ao usuário

Aqui ficam os serviços responsáveis por auxiliarem os usuários na execução das suas tarefas. Eles auxiliam o usuário a compreender que tarefas precisam ser executadas e como. Esse auxílio pode ser: fornecendo ferramentas para facilitar o trabalho do usuário, elucidar dúvidas do usuários sobre a tarefa ou sugerir atividades a serem feitas. Segue a lista de tarefas classificadas nesta classe:

- Collecting documents for tax season; perhaps complete business taxes too;
- Coming up with detailed profiles and inserting links to company website;
- Composing documents from handwritten drafts, faxes and dictations;

- Creating, filing and presenting basic reports (reports on weekly tasks, deliverables, sales);
- Developing and delivering slideshow presentations;
- File Management;
- Generating listicles (list articles) on industry-related matters;
- Liaison between you and other team members;
- Producing graphs from your spreadsheets;
- Proofreading documents and other office materials;
- Writing down minutes from meetings or events and then creating a detailed document;
- Converting, merging and splitting .PDF files.

### B.3.5 Execução de tarefas do usuários

Aqui ficam os serviços que assumem a responsabilidade de execução de tarefas pelo usuário. Pode ser via uma solicitação de agendamento de execução da tarefa que o usuário fez ou através da automatização de uma tarefa que antes era do usuário. Segue a lista de tarefas classificadas nesta classe:

- Bookkeeping and payroll duties: calculating hours, adding expenses, updating salaries;
- Creating and sending out greeting cards, invitations, newsletters and thank you notes;
- Developing, updating and optimizing an SEO and web marketing strategy;
- Doing regular backups to prevent data loss;
- Editing audio files by removing background noise and improving volume levels;
- Establishing follow-up emails and autoresponders and edit according to response rate;
- Answering support tickets.

### B.3.6 Filtro e gerenciamento das informações provindas de fontes externas

Aqui ficam os serviços responsável por receber, filtrar e encaminhar os pacotes de comunicação endereçadas aos usuários. Segue a lista de tarefas classificadas nesta classe:

- Checking email, responding to customer inquiries and managing spam;
- Creating new list of email contacts, email newsletters and promotional copy;
- Filter and reply to comments on your blog;
- Receptionist duties: answering calls, leaving voicemails and checking messages.

### B.3.7 Personalização automática do ambiente ao usuário

Aqui ficam os serviços que visam adaptar automaticamente o ambiente as necessidades do usuário. Os serviços devem detectar as características do usuário e reaproveitar cenários de sucesso em outros perfis semelhantes ao do usuário.

Na relação de exemplos de tarefas de um AV não foi possível encontrar exemplos explícitos para essa classe, porém acreditamos que vários dos serviços já oferecem algum nível de personalização nas entregas dos produtos para os clientes e simplesmente isso é considerado como algo normal e portanto acaba não sendo descrito na especificação da tarefa.

## B.4 Conclusão

Com o experimento de classificação das tarefas de um AHV nas classes de serviços inteligentes de um ASV foi possível demonstrar a factabilidade da nossa classificação.

# APÊNDICE C – Uso do LCC como Protocolo de Interação

## C.1 Definição do Problema

Para a implementação de uma Camada de Serviços Inteligentes (CSI) foi escolhido usar a técnica de Sistema Multiagentes (SMA). Com o uso de SMA o comportamento inteligente vai surgir como um resultado das interações dos agentes inteligentes dentro do SMA. Assim faz-se necessário estabelecer um protocolo de interação entre os agentes de formar atingir os seguintes objetivos:

- Permitir que novos serviços (funcionalidades) sejam facilmente adicionados ou removidos da CSI;
- Permitir que os agentes possam descobrir como encontrar e trocar solicitações de serviços com outros agentes.

O problema reside então em descrever o protocolo e as ferramentas necessárias a sua implantação.

## C.2 Metodologia

Para a resolução do problema foi escolhido utilizar a linguagem Lightweight Coordination Calculus (LCC) para a modelagem do protocolo de interação dos agentes dentro da CSI. Segundo (ROBERTSON, 2004) a LCC serve para a criação de modelos de interação entre agentes. Neste modelo de interação os agentes escolhem quais papéis eles podem ou querem interpretar na interação, o modelo prever também como se dar o início bem como o fim de cada interação. Além disso, cada agente é livre para exercer mais de um papel dentro do modelo. Portanto o LCC oferece uma solução bastante flexível que se adapta bem ao nosso caso.

Assim para a criação de uma Prova de Conceito da factibilidade de uso do LCC dentro da nossa solução foram seguidas as seguintes etapas:

1. Criação de um Modelo de Interação;
2. Implementação de uma instância funcional;

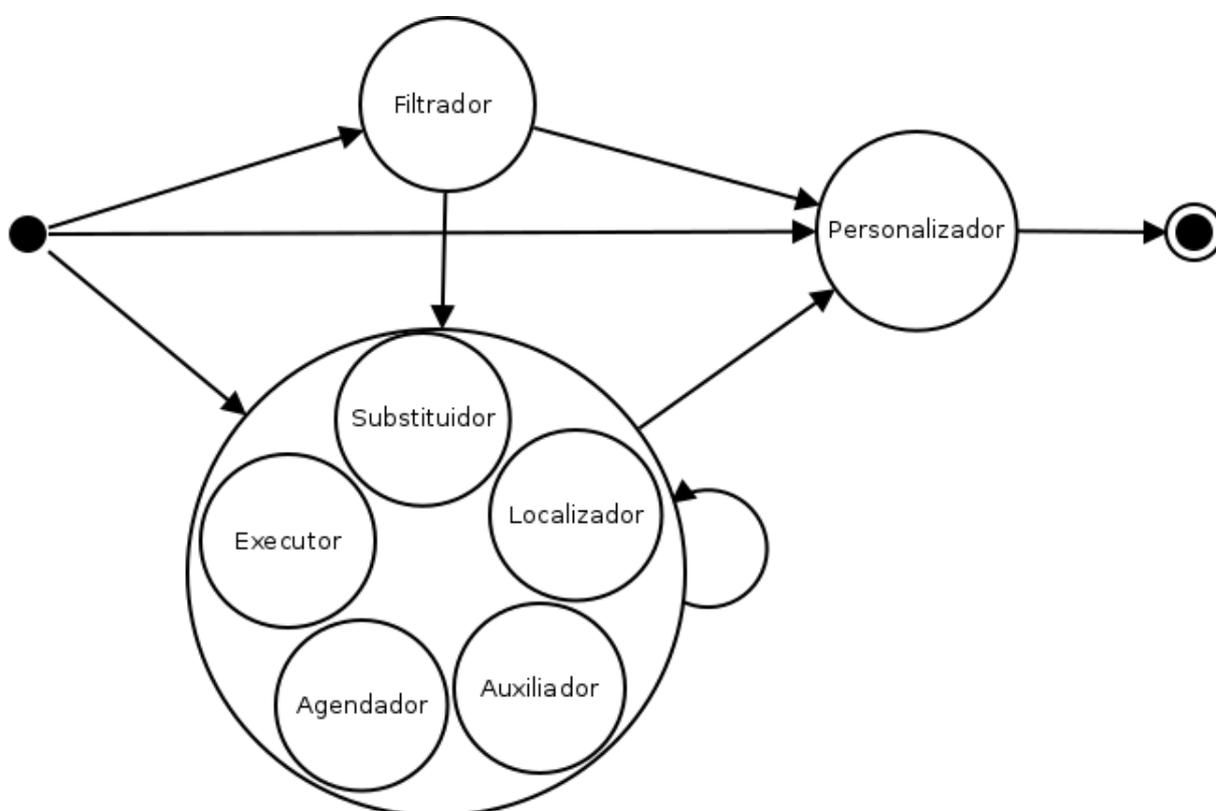
### 3. Testes do Modelo de Interação.

Nas próximas seções descreveremos cada uma destas etapas.

## C.3 Criação de um Modelo de Interação

O modelo de interação a ser criado é baseado no autômato do fluxo de execução de um serviço inteligente apresentado na Figura 37 que foi criado durante o desenvolvimento de um processo para elicitación de requisitos para construção de serviços inteligentes.

Figura 37 – Fluxo de execução de um serviço inteligente.



Fonte: Autor.

Durante a transformação do autômato para um código LCC os estados são traduzidos como papéis dentro do modelo de interação. Assim para simplificar o código LCC os estados Substituidor, Localizador, Auxiliador, Agendador e Executor foram agrupados em um único papel chamado Outro (Other). Resultando no seguinte código:

```

// Papel avatar e o papel inicial da interacao
2 a/avatar, ID0) ::
  // 0 Input sera a entrada do usuario
  null <- getInput(Input) then
  // Temos entao tres possibilidades
6 ((
  
```

```

// Se o Input for uma mensagem ele segue o caminho
// Input -> Filtro -> Outros -> Personalizador -> Output
doTask(Input) => a(filter, ID1) <- isMessage(Input) then
10 taskDone(Output) <= a(filter, ID1) then
doTask(Output) => a(otherMaster, ID2) then
taskDone(Output) <= a(otherMaster, ID2) then
doTask(Output) => a(personalizer, ID5) then
14 taskDone(Output) <= a(personalizer, ID5)
) or (
// Se o Input nao e uma mensagem e temos agentes do tipo Outro
// segue o caminho: Input -> Outros -> Personalizador -> Output
18 doTask(Input) => a(otherMaster, ID2) then
taskDone(Output) <= a(otherMaster, ID2) then
doTask(Output) => a(personalizer, ID5) then
taskDone(Output) <= a(personalizer, ID5)
22 ) or (
// Por fim, se nao e uma mensagem e nao temos agentes do tipo Outro
// segue o caminho: Input -> Personalizador -> Output
doTask(Input) => a(personalizer, ID5) then
26 taskDone(Output) <= a(personalizer, ID5)
)) then
// Retorna o resultado final
null <- displayOutput(Output)
30
// Descricao do papel filtro
a(filter, ID1) ::
// Ao receber uma solicitacao de um agente Avatar
34 // verifica se pode fazer alguma coisa
doTask(Input) <= a(avatar, ID0) <- canDoTask(Input) then
// Se puder, executa a tarefa e retorna o resultado ao agente Avatar
taskDone(Output) => a(avatar, ID0) <- execTask(Input, Output)
38
// Papel auxiliar ao papel Outro, cujo o objetivo e enviar uma
solicitacao
// de tarefa para cada agente com papel Outro de forma encadeada.
a(otherMaster, ID2) ::
42 // Ao receber uma solicitacao do agente Avatar
doTask(Input) <= a(avatar, ID0) then
// Recupera a lista de todos os agentes do tipo Outro
null <- getPeers("other",IDs) then
46 // Faz uma chamada recursiva usando a lista de agentes Outros
a(otherMaster(Input,Output,IDs),ID3) then
// Retorna o resultado para o agente Avatar
taskDone(Output) => a(avatar, ID0)
50
// Chamada recursiva
a(otherMaster(Input,Output,IDs),ID3) ::

```

```
(
54 // Envia solicitacao para o agente Outro identificado por ID
doTask(Input) => a(other, ID) <- IDs = [ID|RIDs] then
// Espera resposta de retorno do agente Outro
taskDone(Output) <= a(other, ID) then
58 // Faz chamada recursiva passando o resultado atual como entrada
// para o proximo agente Outro
a(otherMaster(Output, Output, RIDs), ID3)
) or null <- IDs = []
62
// Descricao do papel Outro
a(other, ID4) ::
// Ao receber um solicitacao, verifica se pode fazer alguma coisa
66 doTask(Input) <= a(otherMaster, ID3) <- canDoTask(Input) then
// Se puder, executa a tarefa e retorna o resultado
taskDone(Output) => a(otherMaster, ID3) <- execTask(Input, Output)

70 // Descricao do papel Personalizador
a(personalizer, ID5) ::
// Ao receber uma solicitacao de um agente Avatar
// verifica se pode fazer alguma coisa
74 doTask(Input) <= a(avatar, ID0) <- canDoTask(Input) then
// Se puder, executa a tarefa e retorna o resultado ao agente Avatar
taskDone(Output) => a(avatar, ID0) <- execTask(Input, Output)
```

### Código 9 – Código LCC Teste do Modelo de Interação

Sobre o código-fonte em LCC apresentado devemos as seguintes observações

- As função `getPeers()` é uma função interna do ambiente LCC e portanto não precisa ser implementada;
- As funções `canDoTask()` e `execTask()` são de responsabilidade do desenvolvedor dos agentes nos papéis Filtro, Outro e Personalizador;
- A função `isMessage()` é de responsabilidade do desenvolvedor dos agentes no papel de Avatar;
- `Input` e `Output` representam o conteúdo das mensagens trocadas entre os agentes e o modelo de interação não define o seu formato, ficando a cargo do projetista do sistema defini-los.

Após a criação do código-fonte em LCC passamos a implementação.

## C.4 Implementação de uma Instância Funcional

Para a implementação da prova de conceito foi escolhido usar o OpenKnowledge Kernel<sup>1</sup> (OKK), pois ele além de implementar o LCC ainda tem um implementado a parte de descoberta de serviços, assim, podemos focar apenas na criação dos agentes e deixar a troca de mensagens e a descoberta dos serviços para o OKK. Para a criação dos serviços/agentes basta então serem codificadas novas classes em Java (que foi a linguagem usada na construção do OKK).

### C.4.1 Definição do Cenário de Teste

Para a implementação foi definido o cenário de teste descrito a seguir.

O usuário enviar uma string para o sistema. O sistema processa essa string e devolve uma nova string. Se a string de entrada começa com o caractere “?” então quer dizer que ela é uma mensagem. Foi definido criar um conjunto de 4 serviços:

- Filtro: Converte toda a string para caixa alta e remove o caractere inicial ‘?’;
- Outro 1: Adiciona a string “ Três Tigres Tristes.” ao final da string;
- Outro 2: Remove espaços em branco no início e no fim da string;
- Personalizador: Troca todos os caracteres “r” por “l” na string (i.e. converte para cebolês).

### C.4.2 Implementação do Input e do Output

No modelo de interação construído o Input e o Output são representações genéricas porém iguais na forma, portanto a sua implementação se deu através de uma única classe que chamamos de `DataLayer`, apresentada a seguir:

```
import java.util.List;
import java.util.ArrayList;
import java.util.Collections;
4 import java.io.Serializable;

public class DataLayer implements Serializable {
    String text = "";
8    List<Integer> ids = Collections.synchronizedList(new ArrayList<
    Integer>());
    boolean message;

    public boolean isMessage() { return message; }
```

<sup>1</sup> Baixado de <<http://groups.inf.ed.ac.uk/OK/>>

```
12     public void setMessage(boolean i) { message = i; }

    public String getText() { return text; }
16
    public void setText(String s) { text = s; }

    public void addID(Integer id) { ids.add(id); }
20
    public boolean containsID(Integer id) {
        return (ids.contains(id));
    }
24 }
```

Destacamos o seguinte ponto sobre a implementação do DataLayer: Ao não utilizar um tipo básico para a implementação mas sim uma classe mais complexa comprovamos a possibilidade de implementar tanto o Input e o Output da forma que for conveniente para a CSI.

### C.4.3 Implementação dos Serviços

Para que os serviços possam ser utilizados no OKK são necessários dois passos: Implementar as funções que o modelo de interação associa com o papel e se registrar como um executor do papel específico dentro do modelo. Esses dois passos foram separados em duas classes para cada serviço. Abaixo apresentamos os códigos-fonte:

#### C.4.3.1 Papel Avatar

```
public class AvatarOKC extends OKCFacadeImpl {
    public boolean isMessage(Argument In) {
        if (In.getValue() instanceof DataLayer) {
4         DataLayer Input = (DataLayer) In.getValue();
            return Input.isMessage();
        } else {
            return false;
8         }
    }

    public boolean getInput(Argument In) {
12     String message = JOptionPane.showInputDialog("Digite uma frase:",
        "?Frases de Teste");
        if (message == null) return false;
        DataLayer Input = new DataLayer();
16     if (message.charAt(0) == '?') {
            Input.setMessage(true);
        }
    }
}
```

```

        Input.setText(message.substring(1));
    } else {
20     Input.setText(message);
        Input.setMessage(false);
    }
    In.setValue(Input);
24     return true;
}

public boolean displayOutput(Argument Out) {
28     DataLayer Output = (DataLayer) Out.getValue();
        JOptionPane.showMessageDialog(null, Output.getText(), "Reply",
            JOptionPane.INFORMATION_MESSAGE);
        return true;
32 }
}

public class PeerAvatar extends Peer {
36     public void init(String[] args) {
        super.init(args);
        manager.storeComponent(createOKC("avatar",
            "org.openk.intelligencelayer.okc.AvatarOKC", null));
40     InteractionTask task = new InteractionTask(this, "avatar",
        AcceptPolicy.ACCEPT_ONE, "intelligencelayer", false);
        String[] q = { "intelligencelayer" };
        task.attempt(q);
44 }

    public static void main(String args[]) {
        PeerAvatar peer = new PeerAvatar();
48     peer.init(args);
    }
}

```

#### C.4.3.2 Papel Filter

```

// Descricao do Papel
2 public class FilterOKC extends OKCFacadeImpl {
    public boolean canDoTask(Argument I) { return true; }

    public boolean execTask (Argument I, Argument O) {
6     DataLayer Input = (DataLayer) I.getValue();
        DataLayer Output = new DataLayer();
        Output.setText(Input.getText().toUpperCase());
        Output.addID(new Integer(0));
10     O.setValue(Output);
        return true;

```

```

    }
  }
14 // Registro e Execucao do Papel
  public class PeerFilter extends Peer {
    public void init(String[] args) {
18      super.init(args);
      manager.storeComponent(createOKC("filter",
          "org.openk.intelligencelayer.okc.FilterOKC", null));
      InteractionTask task = new InteractionTask(this, "filter",
22          AcceptPolicy.ACCEPT_ONE, "intelligencelayer", false);
      String[] q = { "intelligencelayer" };
      task.attempt(q);
    }
26 public static void main(String args[]) {
    PeerFilter peer = new PeerFilter();
    peer.init(args);
  }
30 }

```

#### C.4.3.3 Outro 1

```

// Descricao do Papel
2 public class OtherOKC extends OKCFacadeImpl {
  public boolean canDoTask(Argument I) {
    if (I == null) return false;
    DataLayer Input = (DataLayer) I.getValue();
6    return ! Input.containsID(new Integer(1));
  }

  public boolean execTask (Argument I, Argument O) {
10    if (this.canDoTask(I)) {
      DataLayer Input = (DataLayer) I.getValue();
      DataLayer Output = new DataLayer();
      Output.setText(Input.getText().concat(" Tres Tigres Tristes."));
14    Output.addID(new Integer(1));
      O.setValue(Output);
    } else {
      O.setValue(I.getValue());
18    }
    return true;
  }
  }
22 // Registro e Execucao do Papel
  public class PeerOther extends Peer {
    public void init(String[] args) {

```

```

26     super.init(args);
        manager.storeComponent(createOKC("other",
            "org.openk.intelligencelayer.okc.OtherOKC", null));
        InteractionTask task = new InteractionTask(this, "other",
30         AcceptPolicy.ACCEPT_ONE, "intelligencelayer", false);
        String[] q = { "intelligencelayer" };
        task.attempt(q);
    }
34     public static void main(String args[]) {
        PeerOther peer = new PeerOther();
        peer.init(args);
    }
38 }

```

#### C.4.3.4 Outro 2

```

// Descricao do Papel
2 public class Other2OKC extends OKCFacadeImpl {
    public boolean canDoTask(Argument I) {
        if (I == null) return false;
        DataLayer Input = (DataLayer) I.getValue();
6     return ! Input.containsID(new Integer(10));
    }
    public boolean execTask (Argument I, Argument O) {
        if (this.canDoTask(I)) {
10         DataLayer Input = (DataLayer) I.getValue();
            DataLayer Output = new DataLayer();
            Output.setText(Input.getText().trim());
            Output.addID(new Integer(10));
14         O.setValue(Output);
        } else {
            O.setValue(I.getValue());
        }
18     return true;
    }
}

22 // Registro e Execucao do Papel
    public class PeerOther2 extends Peer {
        public void init(String[] args) {
            super.init(args);
26         manager.storeComponent(createOKC("other2",
            "org.openk.intelligencelayer.okc.Other2OKC", null));
            InteractionTask task = new InteractionTask(this, "other",
                AcceptPolicy.ACCEPT_ONE, "intelligencelayer", false);
30         String[] q = { "intelligencelayer" };
            task.attempt(q);

```

```

    }
    public static void main(String args[]) {
34      PeerOther2 peer = new PeerOther2();
        peer.init(args);
    }
}

```

#### C.4.3.5 Personalizador

```

// Descricao do Papel
public class PersonalizerOKC extends OKCFacadeImpl {
3  public boolean canDoTask(Argument In) {
    return true;
  }
  public boolean execTask (Argument In, Argument Out) {
7    DataLayer Input = (DataLayer) In.getValue();
    DataLayer Output = new DataLayer();
    Output.setText(Input.getText().replace('r', 'l').replace('R', 'L'));
    //ceboles
    Output.addID(new Integer(2));
11   Out.setValue(Output);
    return true;
  }
}
}
15 // Registro e Execucao do Papel
public class PeerPersonalizer extends Peer {
  public void init(String[] args) {
    super.init(args);
19   manager.storeComponent(createOKC("personalizer",
        "org.openk.intelligencelayer.okc.PersonalizerOKC", null));
    InteractionTask task = new InteractionTask(this, "personalizer",
        AcceptPolicy.ACCEPT_ONE, "intelligencelayer", false);
23   String[] q = { "intelligencelayer" };
    task.attempt(q);
  }
  public static void main(String args[]) {
27   PeerPersonalizer peer = new PeerPersonalizer();
    peer.init(args);
  }
}
}

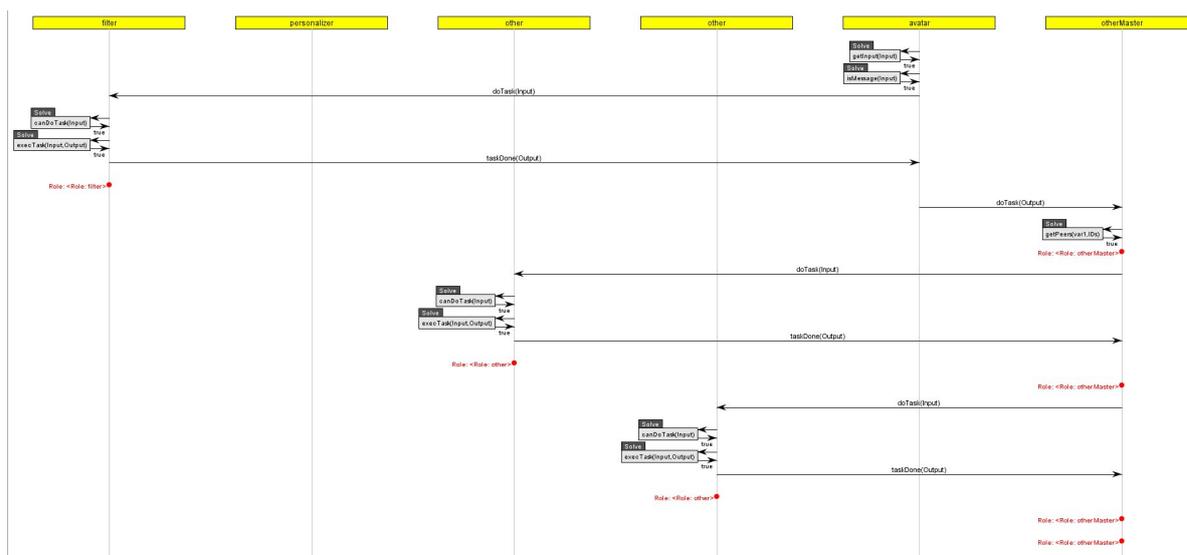
```

O papel OtherMaster por ser um papel auxiliar usado para a criação e não ter a exigência no modelo da criação de implementar uma função não foi mostrado o código. A sua criação consiste apenas em declarar a classe sem nenhum método e o registro e execução desse papel é semelhante ao que é feito para os outros papéis.

## C.5 Testes do Modelo de Interação

Após desenvolver o modelo de interação e desenvolver as classes necessárias a descrição e execução dos papéis do modelo foi feita então com sucesso um teste da execução do modelo de interação. Na Figura 111 mostramos uma tela de debug do OKK mostrando um pedaço da troca de mensagens durante a execução do modelo de interação.

Figura 38 – Tela de debug do OKK.



Fonte: Autor.

## C.6 Conclusões

O uso do OpenKnowledge Kernel (OKK) além de preencher as requisitos elencados no início deste relatório traz as seguintes vantagens:

- Possibilidade de distribuir numa rede de computadores os agentes;
- Entrada e saída dos agentes da CSI pode ser feita a quente sem necessidade de reiniciar o funcionamento dela;
- Como o OKK permite a utilização de mais de um modelo de interação no mesmo ambiente, tem-se a vantagem que o modelo aqui criado pode ser implementado como o modelo padrão da CSI e conforme a necessidade for surgindo novos modelos de interação podem ser acrescentados para atingir necessidades específicas.

# APÊNDICE D – Projeto de Mineração de Dados Educacionais UAB

## D.1 Introdução

Devido ao aumento da capacidade de processamento dos computadores e a melhoria substancial das ferramentas para realização de cursos online houve um aumento da quantidade de informações registradas pelas ferramentas sobre a interação delas com os seus usuários (alunos, professores e administradores) e a interação entre os próprios usuários. Esse aumento da quantidade de informação disponível possibilitou o avanço das áreas de Learning Analytics (LA) e Educacional Data Mining (EDM). Para atingir uma eficiência melhor as áreas de EDM e LA precisam de uma arquitetura que possa trabalhar com as diferenças das bases de dados existentes nos ambientes virtuais de aprendizagem além de melhorar o processo de coleta dos dados.

A arquitetura aqui proposta consegue lidar com diferentes bases de dados e melhorar o processo de coleta ao diminuir o volume de dados transferidos e encurtar o tempo necessário para a execução da fase de coleta dos dados.

### D.1.1 Educacional Data Mining - EDM

Uma descrição da EDM contendo os principais tópicos e objetivos pode ser encontrada em (BAKER; ISOTANI; CARVALHO, 2011). A EDM é definida como a área de pesquisa cujo foco principal é o desenvolvimento de métodos para explorar conjuntos de dados coletados em ambientes educacionais. O objetivo da EDM é conseguir novas informações sobre os relacionamentos entre os dados educacionais de forma a produzir novos conhecimentos ou descobertas científicas, tais como um melhor entendimento do comportamento dos estudantes ou um melhor entendimento do processo de ensino-aprendizagem em ambientes virtuais. Para realizar a análise de grandes quantidades de dados a EDM vem desenvolvendo e usando técnicas de vários subcampos da Ciência da Computação para a exploração dos dados adquiridos em ambientes educacionais.

### D.1.2 Learning Analytics - LA

Segundo [Elias and Lias 2011] a LA é um campo emergente no qual ferramentas sofisticadas de análise são utilizadas para melhorar a aprendizagem e a educação. O campo toma emprestado várias técnicas de outros campos de estudo tais como: Business

Intelligence, Web Analytics, EDM e Action Analytics. LA busca utilizar a capacidade de modelagem existente para prever o comportamento, agir de acordo com as previsões e então realimentar o processo de ensino e aprendizagem de forma a esta sempre melhorando as previsões e o próprio processo de ensino aprendizagem. Ou seja, a LA objetiva melhorar os sistemas de aprendizagem através de adaptações baseadas nas evidências encontradas pelas pesquisas e experimentos realizados na área. Apesar de ser uma área nova a UNESCO já se interessou pela área e publicou recomendações para os pesquisadores conforme mostrado em [Shum 2012].

## D.2 Contextualização do Problema

Ambas as áreas de EDM e LA utilizam técnicas de mineração de dados como ferramentas para as suas pesquisas. Portanto, ambas sofrem dos mesmos problemas durante a fase de coleta dos dados.

### D.2.1 Contextualização em EDM

Em EDM, segundo [Romero et al. 2008], a aplicação de técnicas de mineração de dados segue o mesmo processo padrão cíclico já utilizado em mineração de dados:

1. Coleta dos dados;
2. Preprocessamento dos dados;
3. Aplicação das técnicas de mineração de dados;
4. Interpretação, avaliação e publicação dos resultados;
5. Retorno ao início do ciclo.

Em seu trabalho [Romero et al. 2008], dá um exemplo da aplicação das técnicas de EDM utilizando o Moodle como o Ambiente Virtual de Aprendizagem. Seu exemplo é ilustrado na Figura 39.

O trabalho aqui detalhado é um aprimoramento para as primeiras partes do círculo de mineração de dados: as fases de coleta dos dados e preprocessamento.

### D.2.2 Contextualização em LA

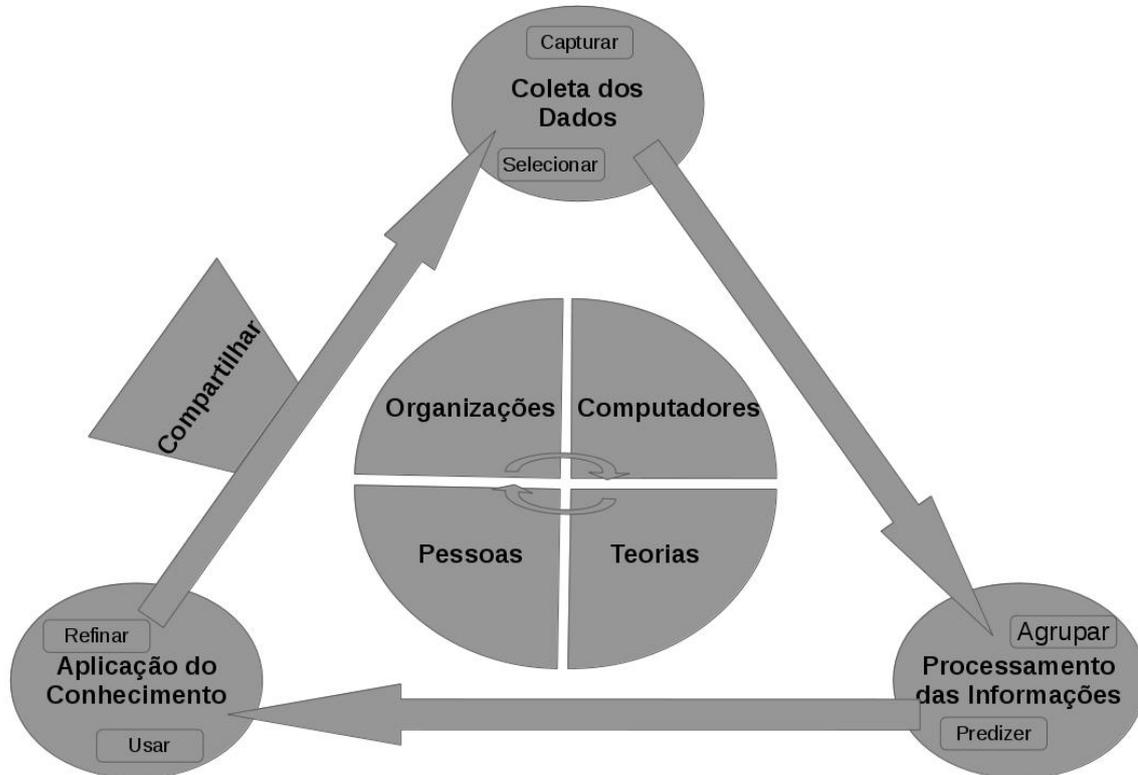
(ELIAS, 2011) utilizando de vários frameworks e modelos (Knowledge Continuum, Five Steps of Analytics, Web Analytics Objectives e Collective Applications Model) constrói e apresenta o modelo do processo de Learning Analytics mostrado na Figura 40.

Figura 39 – Mineração dos dados do Moodle.



Fonte: (ROMERO; VENTURA; GARCÍA, 2008). Traduzido pelo autor.

Figura 40 – Modelo para Learning Analytics.



Fonte: (ELIAS, 2011). Traduzido pelo autor.

Assim como na área de EDM na LA o modelo prever um processo cíclico de melhoria contínua. No caso do trabalho aqui descrito vai ser proposto uma melhoria para as fases de seleção (select) e captura (capture) do processo de LA.

## D.3 Especificação do Problema

O problema que se pretende resolver neste trabalho é o de otimizar a fase de ETL (Extract-Transform-Load) do processo de mineração de dados em diferentes ambientes educacionais para o uso da técnica de processamento de cestas de produtos.

O processo de mineração dos dados educacionais tem várias etapas sendo uma delas o ETL. Nesta etapa o processo recebe como entrada os dados brutos (os bancos de dados e registros de eventos) e produz como resultado os dados aonde será feito o processo de mineração de dados (as cestas). Por exemplo, é nessa fase que se verifica se existe alguma inconsistência nos dados e retira-se essas inconsistências tais como:

- Mesma informação armazenada como dados diferentes (ex: horas em formato am/pm e 24h);
- Mesma informação armazenada com formatos diferentes (campo data pode ser: um número ou uma string).

Na fase de ETL também é comum fazer os cálculos (somatórios, médias, etc.) necessários para se gerar os dados que se tem interesse.

Tradicionalmente durante a execução da fase de ETL os dados brutos são coletados em um único local de onde a ferramenta de ETL faz o trabalho. No entanto, num ambiente de Big Data esta forma não é eficiente pois demanda uma grande quantidade de espaço e largura de banda para a coleta dos dados brutos. Acrescentando ainda que os trabalhos de mineração de dados educacionais nem sempre contam com ambientes homogêneos tem-se um nível a mais de complexidade. Outra camada de complexidade que pode aparecer é quando se trata da questão da proteção da privacidade, como os dados são copiados dos seus locais originais para um único local (que pode ou não pertencer aos donos do local original) essa questão surge como mais um ponto a ser levado em consideração.

Vamos discutir em mais detalhes duas questões principais dos problemas da abordagem tradicional (armazenamento e processo administrativo) e ao final desta seção vamos elencar as características que seriam desejáveis numa solução a ser proposta.

### D.3.1 O Problema do Espaço de Armazenamento

O crescimento da educação a distância na Internet tem como produto a massificação de cursos online e ambientes virtuais de ensino e aprendizagem diversos. Cada um desses

ambientes tem diversas ferramentas e cada uma delas gera uma quantidade significativa de informações sobre o aluno e o processo de ensino-aprendizagem. Tais informações podem ser agrupadas e mineradas em busca de novos conhecimentos que venham a melhorar o próprio desenvolvimento da educação a distância. Como base para a nossa discussão sobre espaço de armazenamento vamos utilizar o Moodle como exemplo pois se trata de uma plataforma de educação a distância bastante popular.

Então analisando primeiramente a questão do registro de eventos temos no Moodle uma tabela chamada logs. Esta tabela armazena todas as ações que os usuários do Moodle realizaram dentro dele. Na Figura 41 apresentamos uma descrição da tabela de log do Moodle na sua versão 3.1.3.

Figura 41 – Descrição da tabela de logs do Moodle 3.1.3.

#	Nome	Tipo	#	Nome	Tipo	#	Nome	Tipo
1	<b>id</b>	bigint(10)	8	<b>crud</b>	varchar(1)	15	<b>relateduserid</b>	bigint(10)
2	<b>eventname</b>	varchar(255)	9	<b>edulevel</b>	tinyint(1)	16	<b>anonymous</b>	tinyint(1)
3	<b>component</b>	varchar(100)	10	<b>contextid</b>	bigint(10)	17	<b>other</b>	longtext
4	<b>action</b>	varchar(100)	11	<b>contextlevel</b>	bigint(10)	18	<b>timecreated</b>	bigint(10)
5	<b>target</b>	varchar(100)	12	<b>contextinstanceid</b>	bigint(10)	19	<b>origin</b>	varchar(10)
6	<b>objecttable</b>	varchar(50)	13	<b>userid</b>	bigint(10)	20	<b>ip</b>	varchar(45)
7	<b>objectid</b>	bigint(10)	14	<b>courseid</b>	bigint(10)	21	<b>realuserid</b>	bigint(10)

Fonte: Autor.

Usando esta descrição como base e calculando o valor mínimo (melhor caso) de um registro nesta tabela (campos varchar ocupando apenas 1 byte) teríamos o seguinte cálculo: 10 campos bigint, 8 campos varchar e 2 tinyint totalizando  $10*8+8*1+2*1 = 90$  bytes por registro, considerando todos os campos varchar no máximo (pior caso) e ignorando o campo longtext teríamos um tamanho por registro de 743 bytes. Supondo então a existência de 100 usuários ativos com 2 horas de atividade por dia por usuário e cada usuário fazendo uma ação a cada 1 minuto, teríamos então aproximadamente 1MB de logs por dia no melhor cenário e 8,5MB no pior. Apenas no Brasil temos mais de 100 instituições participantes da UAB, considerando apenas 100 AVAs daria aproximadamente 103MB por dia no melhor cenário e 850MB no pior. Em um ano teríamos aproximadamente 36GB de dados no melhor cenário e 299GB no pior.

Acontece porém, que temos interesse em dados e informações que estão fora do registro dos eventos (logs) tais como: quais mensagens de um usuário que foram respondidas, a nota do aluno, impacto da contribuição do aluno nos fóruns, etc. Essas informações estão espalhadas pelas várias tabelas do Moodle (assim como acontece em outros AVAs), a solução seria então copiar toda a base de dados do Moodle, porem uma instalação nova do Moodle tem por volta de 200 tabelas e essas outras tabelas somadas ocupam tanto ou

mais espaço que a tabela de log. Ou seja, teríamos que dobrar a nossa estimativa feita anteriormente.

Outro ponto da discussão é que estamos lidando com uma base de dados que vai aumentando conforme o tempo sendo que essa base de dados já vem sendo formada a uma quantidade razoável de anos. Tomando como exemplo a UAB, temos um histórico de mais de 10 anos visto que o projeto UAB foi criado em 2005. E temos IES participantes da UAB que já usavam Moodle e cursos a distância antes da criação da UAB. Lembrando também que no projeto UAB não participam instituições privadas. Portanto, podemos dizer que o potencial volume de informações que poderíamos ter para uma mineração de dados se encaixa no perfil de Big Data.

Por fim, outro problema é a transferência de todos esses dados para um único local. A transferência desse volume de dados para um único ponto faz com que seja necessário uma boa largura de banda e o gerenciamento do processo de transferência para que o link do local de armazenamento não seja saturado durante a transferência dos dados.

### D.3.2 O Problema Administrativo

Mesmo o problema do espaço para armazenamento sendo relevante ele pode ser resolvido através da aquisição de espaço de armazenamento e poder computacional suficiente para guardar e realizar o processamento dos dados em um formato que possa ser utilizado na mineração de dados. Como temos hoje em dia várias empresas que fornecem esses dois tipos de recursos (espaço de armazenamento e poder computacional) o problema poderia então ser reduzido a uma questão de ter recurso financeiro suficiente para obter os outros recursos necessários.

Porém, a principal dificuldade em reunir os dados em um único local é que esses dados são provenientes de diferentes instituições. Cada instituição com autonomia de decisão e regulamentos próprios além das regulações e leis que devem ser observadas por todas elas. Como esses dados são referentes a pessoas tem-se ainda questões de privacidade e segurança da informação em adição as outras dificuldades já relatadas.

Portanto é difícil que as instituições deixem terceiros terem acesso as bases de dados dos seus ambientes virtuais. Essa é uma resistência natural quando entra em questão diversas informações sensíveis tanto sobre os alunos como sobre os professores e administradores do ambiente. Se ter acesso já é difícil, elas entregarem uma cópia da sua base de dados é ainda mais difícil visto que uma vez que os dados estejam fora da instituição ela não tem como garantir o uso que será feito dos dados.

Negociar um acordo entre as instituições que resolva todos esses problemas seria trabalhoso e demorado.

### D.3.3 Características Desejáveis da Solução

Assim uma solução boa para o problema aqui descrito deveria ter as seguintes características:

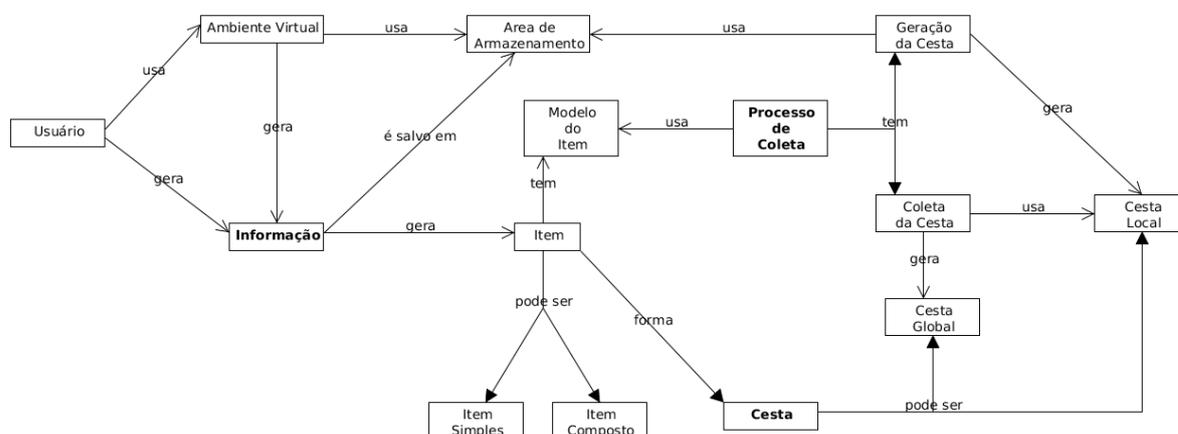
- Reduzir as demandas de espaço para armazenamento e largura de banda em relação ao processo normal;
- Poder trabalhar com diferentes formatos de dados e diferentes formas de armazenamento desses dados;
- Garantir a privacidade dos dados;
- Facilitar o ingresso de novos participantes no processo de coleta de dados.

Neste trabalho apresentamos um modelo de arquitetura para a execução da fase ETL em um processo de mineração de dados educacionais num ambiente de BigData que apresenta essas características identificadas.

## D.4 Modelo de Referência

A seguir é descrito o modelo de referência para a solução proposta. Segundo (MACKENZIE et al., 2006) um modelo de referência serve para entender os relacionamentos mais significativos entre as entidades da solução. O modelo de referência deve consistir de um conjunto mínimo de conceitos unificantes, axiomas e relações dentro do domínio de um determinado problema, além disso, dever ser também independente de padrões, tecnologias, implementações ou outros detalhes mais concretos.

Figura 42 – Modelo de referência.



Fonte: Autor.

Para a representação do modelo de referência foi escolhido o uso de mapas conceituais. Assim, apresentamos na Figura 42 o nosso modelo de referência e em seguida fazemos um detalhamento dos conceitos presentes no mapa conceitual.

#### D.4.1 Informação

Um dos conceitos principais do nosso modelo de referência é o conceito de Informação. Para a definição de Informação no contexto desse trabalho foi utilizado o Knowledge Continuum conforme utilizado por (BAKER, 2007) apud (ELIAS, 2011) e descrito em (ROWLEY, 2007).

**Axioma 1** Dado é um fato observável dentro do Ambiente Virtual. Ele pode ser gerado tanto pelo Usuário como pelo próprio ambiente. Exemplo: Acessar um fórum, ter nota 5 num trabalho, ser aprovado num curso, etc.

**Axioma 2** Uma Informação é o resultado da fusão de um Dado com o seu significado. Para isso é necessário responder a pelo menos um dos seguintes tipos de perguntas: Quem? O que? Quando? Onde? e Quantos?. Exemplo: João acessou o fórum duas vezes, José tem nota 5 no trabalho e Joaquim foi aprovado no curso.

**Axioma 3** Uma Informação não pode ser uma resposta a perguntas do tipo: Porque? e Como?.

As Informações são geradas tanto pelos Usuários como pelo próprio Ambiente Virtual e são enviadas para a Área de Armazenamento afim de serem preservadas.

#### D.4.2 Ambiente Virtual

O Ambiente Virtual é o local aonde além de acontecer o processo de ensino-aprendizagem acontece também as interações entre os usuários e entre o usuário e o ambiente. As interações entre usuários engloba não apenas a relação aluno-professor e sim todos os possíveis relacionamentos, por exemplo: professor-tutor, tutor-aluno, aluno-administração, etc.

Para o seu funcionamento adequado o Ambiente Virtual faz uso de uma Área de Armazenamento.

#### D.4.3 Usuário

Usuário são todos os participantes do Ambiente Virtual, como por exemplo: professores, administradores, alunos, robôs de chat, etc.

#### D.4.4 Área de Armazenamento

Na Área de Armazenamento ficam os dados e informações gerados durante o funcionamento normal do Ambiente Virtual. Algumas informações podem ser geradas através da interação de um usuário, já para outras pode ser necessário uma maior complexidade para que elas sejam geradas. Por exemplo, o horário de login de um usuário é uma informação gerada através de um ato do Usuário (o ato de logar no sistema). Por outro lado, a nota média de um grupo requer uma consulta a todos os usuários participantes do grupo bem como a realização dos cálculos necessários a obtenção do valor da média.

#### D.4.5 Cesta

Este trabalho é focado na técnica de mineração de dados baseado em cesta de compras. Ela é uma técnica para análise de dados exploratória aonde temos como objetivo procurar padrões interessantes. Através então da análise e estudo desses padrões novos conhecimentos podem ser descobertos.

Assim a Cesta aqui vai conter as informações de um determinado aluno em um determinado curso. O mesmo aluno em dois cursos diferentes vai significar a criação de duas Cestas.

**Axioma 4** Cada Cesta é única para a relação aluno-curso.

A Cesta é composta por Itens e todas as Cestas tem o mesmo tamanho.

**Axioma 5** Cada Item aparece em todas as Cestas.

**Axioma 6** Cada Item aparece apenas uma vez em cada Cesta.

Para a obter as cestas o sistema deve executar um Processo de Coleta das cestas.

#### D.4.6 Item

Um Item é o componente básico da Cesta. Cada Item representa uma Informação na Cesta. O item é classificado como um Item Simples ou um Item Composto dependendo da forma como ele é obtido. Para a criação de um Item deve ser seguido o seu Modelo do Item.

#### D.4.7 Item Simples

Um Item é considerado um Item Simples se para obter o seu valor é necessário apenas uma consulta da sua existência na Área de Armazenamento. Exemplo: Quantidade de vezes que determinado usuário acessou um fórum.

**Axioma 7** Um Item Simples é um item que não usa nenhum outro Item na sua estrutura.

#### D.4.8 Item Composto

Um Item é considerado um Item Composto se para obter o seu valor é necessário a utilização de algum processo de calculo ou combinação de outros dois Itens. Exemplo: Porcentagem de posts do usuário que foram respondidas.

**Axioma 8** Cada Item Composto pode ser separado em dois ou mais itens e uma ou mais operações.

**Axioma 9** Um Item Composto pode utilizar outro Item Composto na sua estrutura.

#### D.4.9 Modelo do Item

O Modelo do Item contém a especificação de como um determinado Item é criado. Ele é composto por:

- Um nome único que identifica o modelo;
- Uma descrição do Item a ser construído pelo modelo. A descrição tem como objetivo deixar claro aos interessados no processo de mineração o que aquele Item significa;
- Instruções de como obter o item. As instruções podem variar dependendo se o local onde elas são executadas é diferente, porém devem sempre retornar o mesmo Item (mesmo que por ações diferentes em locais diferentes).

#### D.4.10 Processo de Coleta

A solução proposta se inspira no cenário de um único coletor na época de colheita tendo de coletar Cestas em vários locais e reuni-las em um único ponto, seguindo para isso um Processo de Coleta.

O Processo de Coleta deve então especificar como o trabalho é dividido entre os vários locais e como proceder em cada um dos locais. Ele deve também tratar de eventuais conflitos que apareçam entre os processos nos diversos locais de coleta bem como de conflitos durante o armazenamento das cestas em um ponto único.

O Processo de Coleta portanto é dividido em dois subprocessos: o processo de Geração das Cestas e o processo de Coleta das Cestas.

#### D.4.11 Geração das Cestas

O processo de Geração das Cestas é análogo ao processo de plantação e coleta executado em uma fazenda. Ou seja, ele executa as ações necessárias diretamente em cada um dos Ambientes Virtuais que tem as Informações que vão gerar os Itens que por sua vez vão compor as Cestas.

As Cestas geradas nessa etapa são as Cestas Locais.

#### D.4.12 Cesta Local

Uma Cesta Local refere-se ao conjunto de Itens que foram coletados localmente em um Ambiente para uma relação aluno-curso. Posteriormente as Cestas Locais vão ser coletadas e reunidas em um único ponto.

#### D.4.13 Coleta das Cestas

O processo de Coleta das Cestas é análogo ao processo de envio dos produtos de uma fazenda para um grande armazém central. O processo deve especificar como as Cestas Locais são recuperadas e qual é o processo para a criação da Cesta Global. Restrições (volume, velocidade, tempo, etc) durante a recuperação das Cestas Locais devem ser tratados e resolvidos.

Durante a criação da Cesta Global o processo de coleta pode ainda criar e acrescentar nas Cestas algum Item Composto cujo cálculo envolvam informações de mais de um Ambiente Virtual, como por exemplo: uso da ferramenta de chat pelo Ambiente Virtual em relação ao uso de chats em todos os Ambientes Virtuais participantes do experimento.

#### D.4.14 Cesta Global

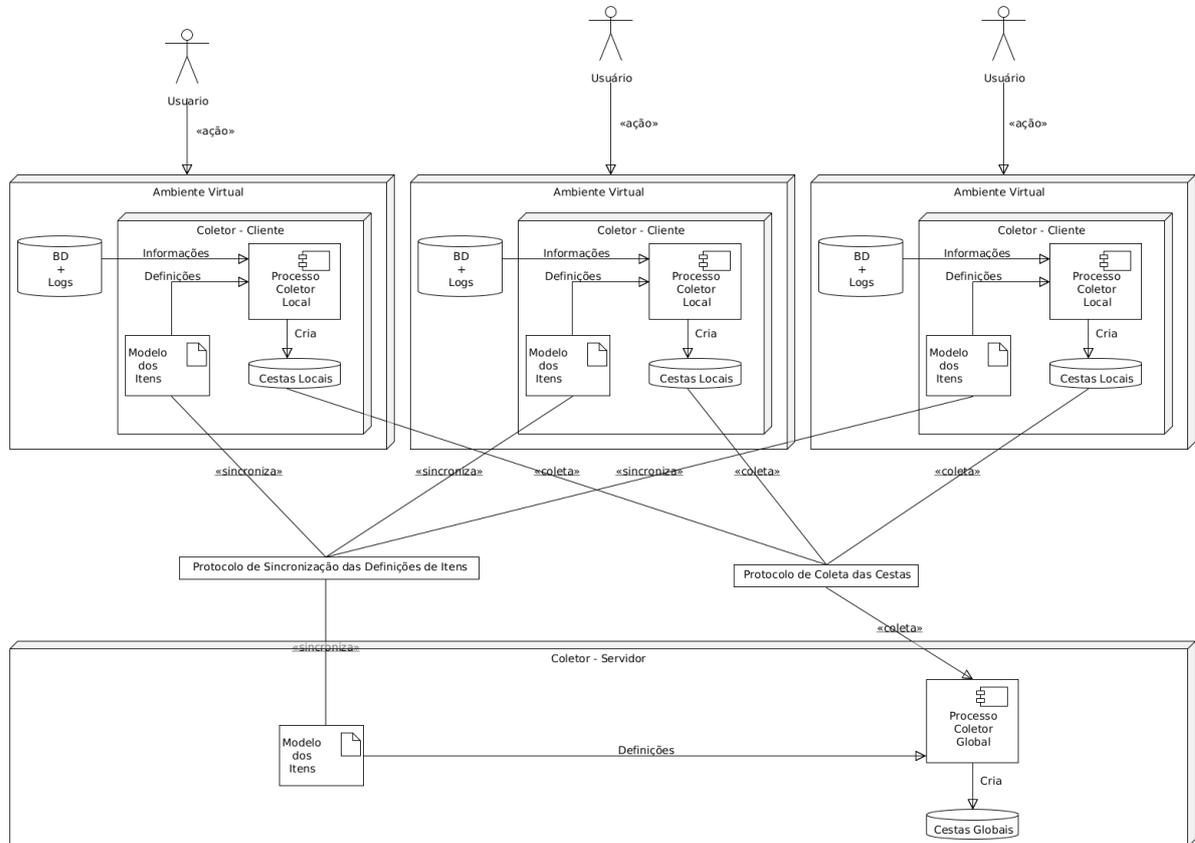
A Cesta Global é o objetivo de todo o processo descrito neste trabalho. Ela é a junção de todas as Cestas Locais de todos os Ambientes Virtuais que façam parte do experimento de mineração de dados mais os Itens Compostos calculados durante o processo de Coleta das Cestas.

### D.5 Arquitetura de Referência

Usando então o Modelo de Referência mais o estilo arquitetural cliente-servidor foi criada a Arquitetura de Referência apresentada na Figura 43 aonde cada ambiente de onde os dados serão coletados será um cliente e o local final aonde as os itens serão armazenados para o processo de mineração é o servidor.

A seguir descrevemos os elementos da arquitetura e por fim as vantagens que ela apresenta.

Figura 43 – Arquitetura de referência.



Fonte: Autor.

### D.5.1 Ambiente Virtual, Cesta Locais, Cestas Globais e Usuário

Iguais ao descrito no Modelo de Referência.

### D.5.2 BD + Logs

O Ambiente Virtual deve armazenar os dados necessários ao seu funcionamento em um Banco de Dados (BD). Além do BD outro ponto de interesse para a mineração de dados em ambientes educacionais é o Registro de Eventos (Log) do Ambiente Virtual. Em alguns sistemas o Log fica dentro do mesmo BD do ambiente em outros o Log fica separado e ainda tem os casos aonde o Log pode ficar em vários lugares diferentes. Qualquer que seja o sistema, o Coletor precisa ter acesso tanto ao BD como ao Log do Ambiente Virtual para poder coletar a maior quantidade possível de informações.

### D.5.3 Coletor – Cliente

O Coletor Cliente é o software que será executado dentro do Ambiente Virtual que vai receber as informações do sistema para gerar as Cestas Locais do ambiente aonde ele está instalado, para posterior envio para o Coletor Servidor. Outra responsabilidade do Coletor Cliente é garantir que o Modelo dos Itens esteja atualizado e sincronizado durante o processo de ETF da mineração de dados.

### D.5.4 Processo Coletor Local

O Processo Coletor Local é responsável por utilizar as definições guardadas no Modelo dos Itens para processar as informações do Ambiente Virtual com o objetivo de criar as Cestas Locais.

### D.5.5 Modelo dos Itens

O Modelo dos Itens é um repositório compartilhado entre os Coletores Cliente e o Coletor Servidor das definições dos Itens que fazem parte das cestas que serão geradas. A sua atualização e sincronização é feita através de um Protocolo de Sincronização das Definições de Itens. Ele deve identificar quais itens podem ser gerados pelo Coletor Cliente e quais podem ser gerados pelo Coletor Servidor.

### D.5.6 Protocolo de Sincronização das Definições de Itens

Este protocolo especifica os procedimentos para criação, alteração, remoção e distribuição das definições de itens que estão salvas no repositório Modelo dos Itens.

### D.5.7 Protocolo de Coleta das Cestas

O Protocolo de Coleta das Cestas deve especificar como as Cestas Locais vão ser coletadas pelo Coletor Servidor. Ele deve tratar questões tais como: prioridades, horários de coleta e particionamento ou não das cestas.

### D.5.8 Coletor – Servidor

O Coletor Servidor é responsável por coletar todas as Cestas Locais dos Ambientes Virtuais e gerar as Cestas Globais. Para isso, ele executa o Protocolo de Coleta das Cestas.

### D.5.9 Processo Coletor Global

O Processo Coletor Global é responsável por organizar e executar a coleta das Cestas Locais dos Coletores Clientes e fazer um último processamento nos itens (usando o

Modelo dos Itens) antes de armazenar os resultados.

Este processamento extra dos itens é necessário no caso de itens que para o seu cálculo seja necessário levar em consideração mais de um Ambiente Virtual. Por exemplo, caso um dos itens seja a idade média dos alunos em todos os ambientes virtuais, então no servidor ao receber cada uma das médias e quantidade de aluno de cada ambiente poderia então realizar o cálculo da média global dos ambientes antes de armazenar a informação.

### D.5.10 Vantagens da Arquitetura Proposta

Na arquitetura proposta cada cliente processa localmente os itens da cesta dentro do escopo do seu Ambiente Virtual. Isto permite o desenvolvimento de melhorias no processo de ETL usado durante a mineração de dados educacionais. A seguir apresentamos essas melhorias.

#### D.5.10.1 Processamento paralelo dos itens

Como os itens são calculados independentemente em cada cliente temos que o processo como um todo fica naturalmente paralelizado. Assim, temos um ganho de desempenho na aquisição das cestas visto que a ocorrência de atraso em um Ambiente Virtual não acarreta em atrasos nas aquisições dos outros Ambientes Virtuais.

#### D.5.10.2 Economia de espaço de armazenamento e largura de banda

Em vez de reunir todos os dados de todos os ambientes em um único local antes de realizar o processo de ETL, apenas os itens da cesta de mineração de dados é que são transmitidos para o servidor. Isto permite uma economia de tamanho no armazenamento e largura de banda necessários em uma grande escala.

#### D.5.10.3 Melhor privacidade dos dados

Como apenas o item da cesta é transmitido apenas as informações necessárias a mineração de dados são transmitidas reduzindo assim a possibilidade de informações sensíveis serem transferidas. Outro ponto é que os clientes podem fazer o processo de anonimização dos dados e o servidor não vai precisar se preocupar com isso.

#### D.5.10.4 Maior tenacidade

Possibilidade maior de conseguir dados mesmo de servidores remotos que estejam fisicamente em locais com pouca largura de banda, por exemplo: Ambientes Virtuais de Aprendizagem (AVA) em cidades do interior ou mesmo em escolas particulares e/ou públicas aonde o acesso a Internet é estressado devido a quantidade de usuários, mas

dentro da rede interna a largura de banda é suficiente para o uso de um AVA interno a rede.

#### D.5.10.5 Fácil de expandir

Tanto os clientes como o servidor tem acesso a um repositório aonde ficam as definições dos itens da cesta. Ao se acrescentar um novo item ao repositório essa informação é compartilhada com todos os clientes e com o servidor.

No repositório de definição dos itens da cesta ficam armazenadas informações sobre o que é o item da cesta (nome, tipo básico e descrição) e a sua forma de cálculo (caso seja necessário). Cada cliente é responsável por implementar localmente as extrações dos itens definidos no repositório. Assim pode-se tratar ambientes e formas diferentes de armazenamento através de implementações diferentes dos clientes, sendo que clientes iguais podem usar implementações iguais.

Por exemplo, digamos que um dos itens a ser utilizado nas cestas do processo de mineração de dados seja o item “Total de emails”. Sendo que no ambiente virtual de aprendizagem A esse dado fica na tabela users, no ambiente B esse dado fica na tabela students e no ambiente C para se pegar essa informação é necessário fazer um JOIN nas tabelas people e people\_emails. Assim com três implementações para os ambientes A, B e C podemos resolver o problema de formato de dados e formas de armazenamento diferente enquanto preservamos o sentido do item, ou seja, preservamos a informação mesmo ela estando representada por dados diferentes em ambientes diferentes.

## D.6 Implementação de uma Instância da Arquitetura

Seguindo a descrição da proposta deste trabalho apresentamos o desenvolvimento de uma aplicação que é uma instância da Arquitetura de Referência apresentada anteriormente. Outras implementações da Arquitetura de Referência são possíveis e a aplicação aqui descrita é apenas uma as muitas possibilidades de implementá-la.

A aplicação foi desenvolvida para ser utilizada para a coleta de dados nos Moodles existentes dentro do âmbito do projeto UAB.

A aplicação desenvolvida é dividida em duas partes: um plugin a ser instalado nos Moodles para coletar os dados e um software servidor que vai receber e armazenar as cestas em um local único.

Devido a restrição de tempo para o desenvolvimento da aplicação foram tomadas as seguintes decisões para simplificar a implementação:

- O Modelo do Item foi implementado optando-se por colocar diretamente dentro do

código do plugin as definições e modelos dos itens a serem coletados;

- O Protocolo de Sincronização das Definições dos Itens foi implementado definindo que as atualizações de itens (inclusão, exclusão e alteração) implicam em uma atualização da versão do plugin. Assim a sincronização se dar ao atualizar todos os pontos de coleta para usarem a mesma versão do plugin e do servidor.

### D.6.1 Casos de Uso

Inicialmente foram identificados os seguintes atores com função relevante na execução da aplicação:

- Administrador do Moodle;
- Gerente / Professor / Tutor;
- O próprio plugin.

Com base nesses atores foram identificados os principais casos de uso conforme Figura 44. A seguir fazemos uma descrição deles.

#### D.6.1.1 Instala Plugin

Um administrador do Moodle baixa o plugin e copia ele para dentro do diretório de instalação do Moodle e executar o procedimento de instalação de plugins do Moodle. Se for a primeira vez que é instalado o plugin no Moodle o administrador terá acesso ao caso de uso “Configura opções do plugin” ao final da instalação.

#### D.6.1.2 Configura Opções do Plugin

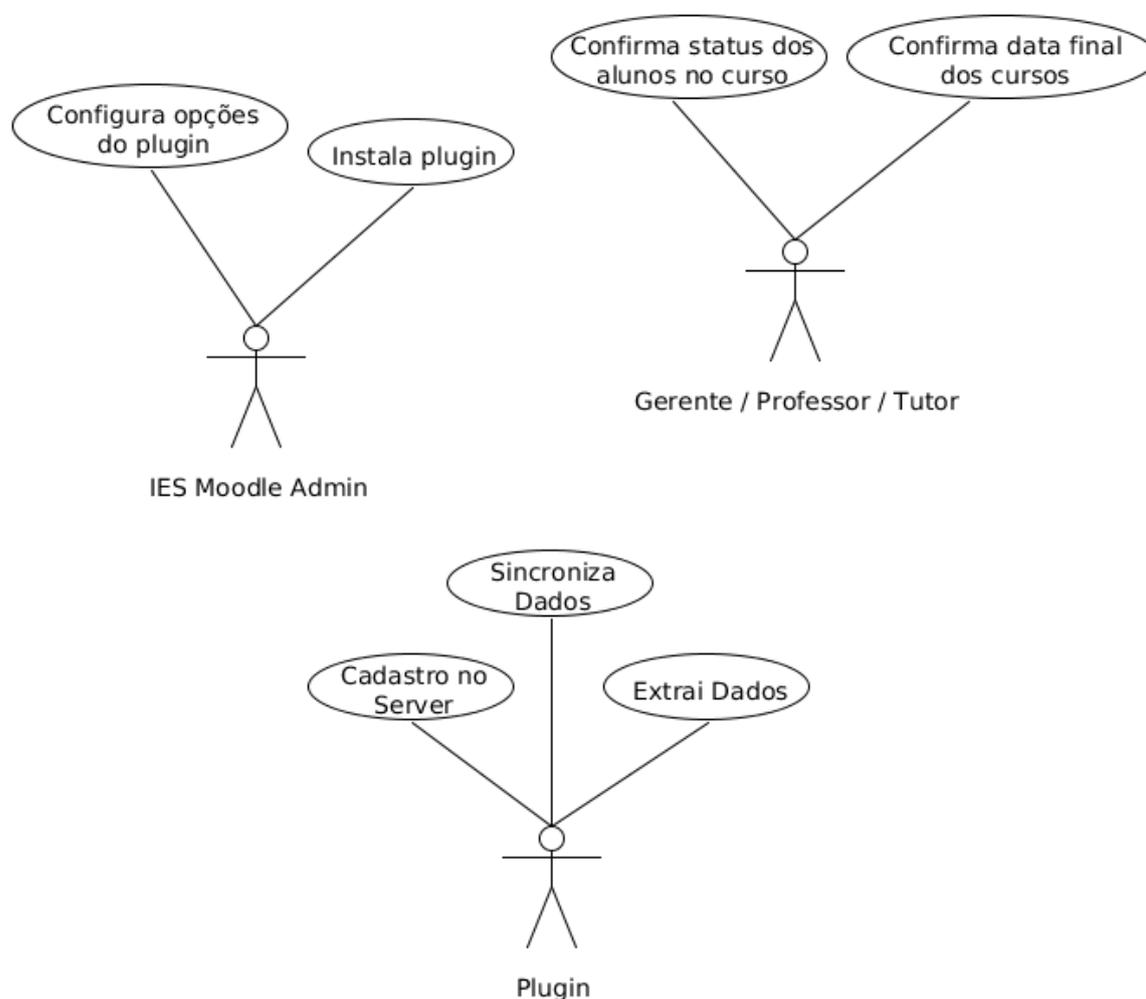
Após a instalação do plugin um administrador do Moodle precisa configurar algumas opções dele para que ele possa executar o seu trabalho corretamente. Para isso ele acessa o formulário de configuração do plugin. Os nomes e a descrição dos itens do formulário são detalhados na Tabela 18.

#### D.6.1.3 Confirma Status dos Alunos no Curso

Um Tutor / Professor / Gerente acessa uma lista dos alunos do curso e para cada aluno ele deve indicar um dos seguintes status:

- Aluno aprovado;
- Aluno reprovado;

Figura 44 – Principais Casos de Uso.



Fonte: Autor.

Tabela 18 – Configurações do plugin

Nome	Descrição
URL do servidor	Uma URL para o servidor que vai coletar as cestas.
Chave de registro no servidor	Uma chave única que identifica esse Moodle no servidor.
IES	A Instituição Superior de Ensino responsável pelos cursos do Moodle.
Início do Job	Horário em que o plugin pode iniciar os trabalhos necessários a coleta dos itens das cestas locais e transferência para o servidor.
Tempo máximo de execução	Período máximo de tempo que o plugin pode ficar em execução, após esse período o plugin congela as suas atividades e na próxima execução retorna do ponto de onde parou.
Papéis que são alunos	Informa quais dos papéis dentro do Moodle são usados por alunos.

Fonte: Autor.

- Aluno desistiu;
- Não é aluno.

#### D.6.1.4 Confirma Data Final do Curso

Um Tutor / Professor / Gerente acessa um formulário para informar a data final de um curso do Moodle. Isto é necessário pois o Moodle não tem uma data de encerramento do curso.

#### D.6.1.5 Cadastro no Servidor

O Plugin ao detectar que não tem uma chave de registro entra em contato com o servidor e solicita a sua chave de registro. O servidor então verifica se ele já tem uma chave para o Moodle do plugin. Caso já tenha ele envia a chave, caso não tenha ele gera uma nova chave e a envia ao plugin.

#### D.6.1.6 Extrai Dados

Após um curso ficar marcado como pronto para processamento (por um Tutor / Professor / Gerente) o Plugin vai na sua próxima janela de atividade executar as ações necessárias para a extração dos itens e das cestas referentes aquele curso. Caso não seja possível terminar toda a extração em uma única janela de execução o processo será retornado de onde parou na próxima janela de execução.

#### D.6.1.7 Sincroniza Dados

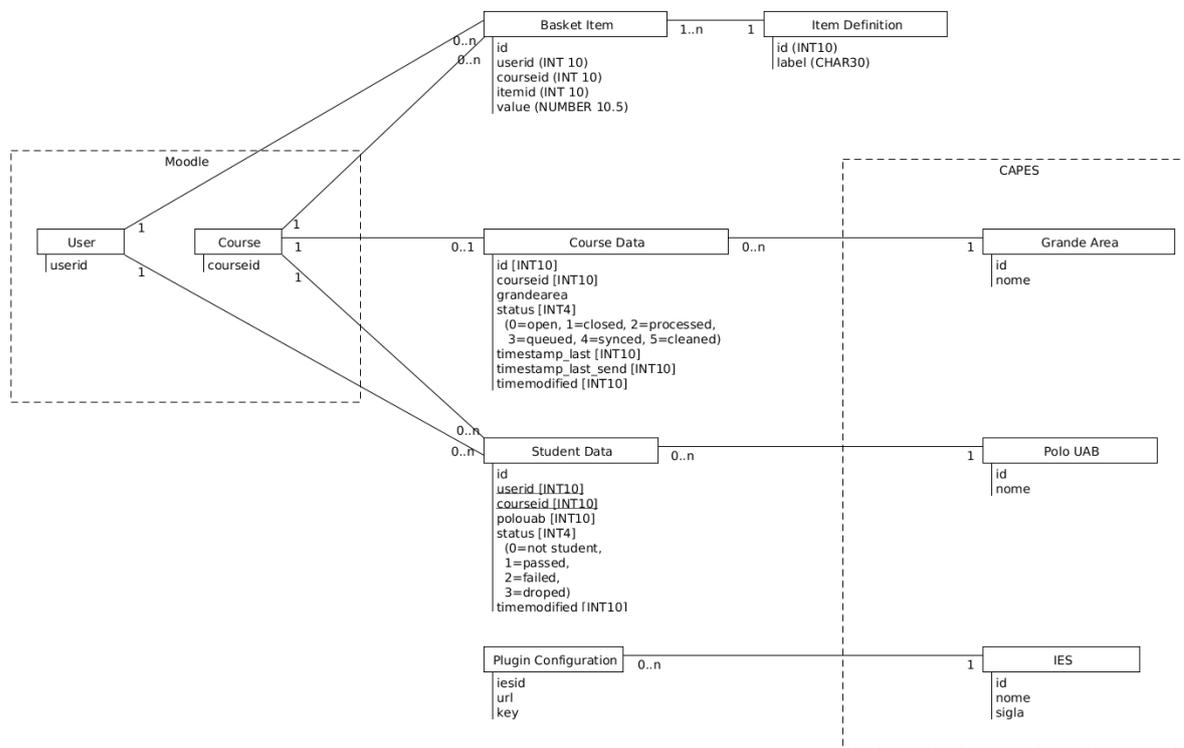
Após preparar as cestas locais o plugin entra em contato com o servidor e informa que está pronto para transmitir as cestas. As cestas são agrupadas pelo curso ao qual pertencem para poderem serem enviadas. Caso não seja possível enviar todas as cestas de todos os cursos o envio é retomado de onde parou na próxima janela de execução do plugin.

### D.6.2 Modelo Entidade Relacionamento

Após descrito os casos de uso do plugin foi feita a construção das tabelas do plugin que serão criadas dentro do Moodle para que plugin possa funcionar. O próprio Moodle já prover os meios dos seus plugins criarem as tabelas necessárias ao seu funcionamento ficando então a cargo do programador especificar como são as tabelas.

O modelo Entidade-Relacionamento do plugin é apresentado na Figura 45. Nele podemos notar duas áreas pontilhadas chamadas Moodle e CAPES. A área Moodle representa tabelas já existentes dentro do Moodle que o plugin vai utilizar no seu funcionamento.

Figura 45 – Modelo ER do plugin.

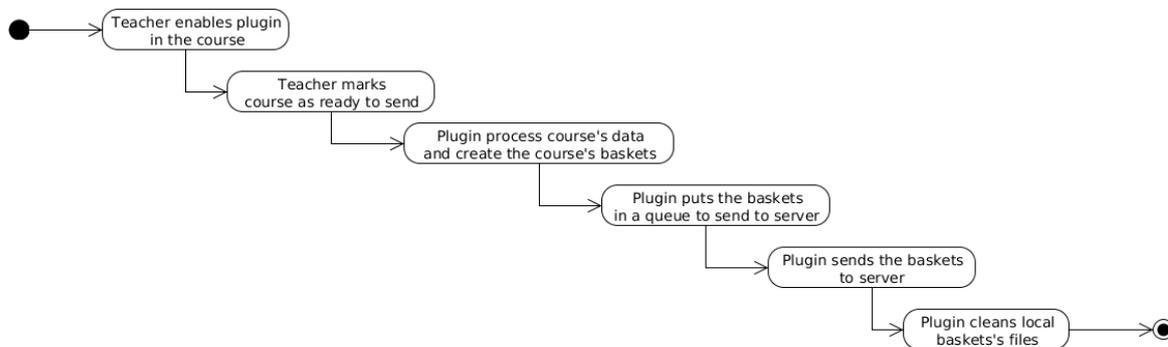


Fonte: Autor.

Já na área CAPES estão informações que o plugin vai usar mas que são da base de dados da CAPES, elas permitiram que uma vez que a cestas globais estejam salvas no servidor fazer uma expansão dessas cestas levando em consideração os dados e informações da base de dados da CAPES.

### D.6.3 Diagrama de Atividades do Plugin

Figura 46 – Diagrama de Atividades do Plugin



Fonte: Autor.

O plugin executa o processamento para cada curso em que ele foi habilitado, assim os administradores e professores do Moodle podem escolher quais cursos vão ser minerados ou não. Para cada curso o plugin segue o diagrama de atividades mostrado na Figura 46.

#### D.6.4 Servidor

A parte do Servidor é simples de implementar bastando construir os seguintes pontos:

- Comunicação entre cliente/servidor;
- Operações CRUD para o gerenciamento dos Moodles a serem coletados;
- Armazenamento das cestas.

Para a comunicação entre cliente e servidor foi utilizado o protocolo HTTP pois já é bastante difundido e existem várias bibliotecas disponíveis para uso.

O gerenciamento dos Moodles clientes guarda a URL de cada Moodle. Portanto podemos ter vários Moodles da mesma IES que não será um problema.

O Armazenamento das cestas é feito diretamente no sistema de arquivos do servidor seguindo a hierarquia: no topo ficam as IES, dentro de cada IES separa por Moodle cliente. Esta organização visa facilitar futuros processos de mineração de dados que envolvam apenas algumas IES.

### D.7 Resultados Encontrados

Foram coletados dos ambientes do CED/UFAM um total de 148 itens de informação. No entanto ao tentar executar as técnicas de mineração com a totalidade dos itens não conseguimos obter sucesso devido a limitação da capacidade de CPUs e memória RAM disponível durante o experimento. Assim para poder apresentar algum resultado foi escolhido utilizar um subconjunto dos itens que foram coletados. A construção desse subconjunto foi feita ao escolhermos utilizar os itens que foram originados da informação de especialistas juntamente com a todos os itens recolhidos do log do Moodle que são de ações executados por alunos e tenham relação com as principais ferramentas utilizadas no CED/UFAM (fórum, tarefas, chat, questões e wiki).

A Tabela 19 apresenta os itens utilizados bem como os seus significados.

#### D.7.1 Normalização dos dados

Para poderem ser utilizada a técnica de mineração de regras de associação é necessário que os dados sejam normalizados para que não tenhamos valores contínuos mas

Tabela 19 – Itens minerados

Item	Descrição
avg_days_bef_due	Média de dias em que o aluno entrega as tarefas antes do prazo das tarefas acabar.
avg_freq_access	Média da frequência de acessos que o usuário teve no ambiente.
c_assignment{upload}	Quantidade de vezes que o aluno enviou uma tarefa.
c_assignment{view all}	Quantidade de vezes que o aluno visualizou uma lista de todas as tarefas de um curso.
c_assignment{view}	Quantidade de vezes que o aluno visualizou uma tarefa.
c_chat{talk}	Quantidade de vezes que o aluno enviou mensagens nos chats.
c_chat{view all}	Quantidade de vezes que o aluno visualizou uma lista de todos os chats de um curso.
c_chat{view}	Quantidade de vezes que o aluno visualizou um chat.
c_course{view}	Quantidade de vezes que o aluno acessou a página principal do curso.
c_forum{add discussion}	Quantidade de vezes que o usuário criou um tópico de discussão usando a ferramenta fórum.
c_forum{view discussion}	Quantidade de vezes que o aluno visualizou uma lista de tópicos de discussão no curso.
c_forum{view forums}	Quantidade de vezes que o aluno visualizou uma lista de todos os fóruns do curso.
c_forum{view forum}	Quantidade de vezes que o aluno visualizou um fórum.
c_quiz{attempt}	Quantidade de vezes que o aluno fez um quiz.
c_quiz{review}	Quantidade de vezes que o aluno revisou um quiz.
c_user{view all}	Quantidade de vezes que o aluno visualizou a lista de participantes do curso.
c_user{view}	Quantidade de vezes que o aluno visualizou o perfil de um aluno.
c_wiki{edit}	Quantidade de edições que o aluno fez nas wikis do curso.
c_wiki{view all}	Quantidade de vezes que o aluno visualizou uma lista com todos os wikis do curso.
c_wiki{view}	Quantidade de vezes que o aluno acessou os wikis do curso.
grandearea	Área Acadêmica do Curso UAB.
moodleversion	Versão do Moodle.
n_access	Quantidade total de acessos ao curso.
n_assign	Quantidade de tarefas feitas.
n_days_first_access	Quantidade de dias entre o início do curso e o primeiro acesso do aluno.
n_msg_ans	Quantidade de mensagens enviadas para outros participantes durante o curso.
n_msg_ans_teacher	Quantidade de mensagens enviadas para o professor durante o curso.
n_msg_ask	Quantidade de mensagens recebidas de outros participantes durante o curso.
n_msg_ask_teacher	Quantidade de mensagens recebidas do professor durante o curso.
n_post_ans_forum	Quantidade de postagens feitas nos fóruns pelo usuário em resposta a um usuário.
n_post_ans_stu	Quantidade de postagens feitas como comentário em resposta a um post de um aluno.
n_post_ask_stu	Quantidade de postagens que outros alunos fizeram como comentário em resposta a um post do usuário.
n_post_forum	Quantidade total de posts do usuário.
n_post_rev_forum	Quantidade total de revisões ou modificações que foram feitas em um post já feito.
polouab	Polo UAB.
status	Status do aluno (APROVADO/REPROVADO).

Fonte: Autor.

sim discretos. Com isso o valor da informação (com exceção de status, polouab, grandearea e moodleversion) foram normalizados usando usando a distribuição de em quartil de cada informação dentro de uma turma. Assim os valores possível para as informações são: Q1 (quartil1), Q2 (quartil 2), Q3 (quartil 3) e Q4 (quartil 4).

### D.7.2 Técnicas Utilizadas

Foram utilizadas duas técnicas a mineração de regras de associação de classe para o item status e a mineração de regras de associação esporádica. O objetivo desse experimento é o demonstrar os tipos de conhecimentos que podem ser potencialmente descobertos na UAB.

Regras de associação de classe são regras aonde a parte direita da regra tem apenas um componente. Esse tipo de regra pode ser utilizado para classificação.

Regras de associação esporádicas são regras que descrevem situações que aparecem raramente (em relação ao conjunto total) é que portanto podem ser utilizadas para descoberta de padrões sobre reprovação, desde que a taxa de reprovação seja baixa (menos que 30%, 20% do total).

### D.7.3 Recomendações para o CED/UFAM

De posse das informação obtidas através do processo de mineração recomendamos as seguintes ações:

1. O método de avaliação deveria dar um peso maior para a participação nos fóruns do que em manter uma média de acesso ao ambiente;
2. Aumentar a quantidade de atividades de acompanhamento dos alunos aonde o foco seja incentivar a entrega dos trabalhos;
3. Para os alunos da área de GA60200006 (ADMINISTRAÇÃO) focar o acompanhamento nos alunos que tem uma taxa de acesso ao ambiente baixa;
4. Incentivar uma maior comunicação dos tutores/professores com os alunos através da troca de mensagens entre eles;
5. Recompensar os alunos que entregarem as tarefas com antecedência, objetivando aumentar a quantidade de alunos que entregam as tarefas com antecedência;
6. Realizar maiores pesquisas sobre os alunos da área de GA60200006 (ADMINISTRAÇÃO) com objetivo de descobrir o motivo da taxa de aprovação ser maior nessa área;

7. Monitorar os alunos que entregam tarefas com antecedência mas que tem pouco acesso ao curso ou se comunicam pouco com outros alunos ou professores. Objetivando incentivar a entregar de um trabalho quando o prazo estiver vencendo ou já vencido;
8. Utilizar os fóruns para realizar lembrete de entrega de trabalhos. Talvez na forma de uma assinatura do tutor/professor que seja inserida junto com os comentários do tutor/professor;
9. Monitorar os alunos que tem uma frequência média de acesso ao ambiente e uma troca de mensagens na média com os alunos e professores. Objetivando incentivar a entregar de um trabalho quando o prazo estiver vencendo ou já vencido;
10. Monitorar alunos da área GA20100000 (BIOLOGIA GERAL) com pouca comunicação com o professor. Objetivando incentivar a entregar de um trabalho quando o prazo estiver vencendo ou já vencido;
11. Monitorar alunos com participação alto nos fóruns e que tem uma taxa média de troca de mensagens com outros alunos ou o professor. Objetivando incentivar a entregar de um trabalho quando o prazo estiver vencendo ou já vencido;
12. Monitorar alunos que respondem outros alunos no fórum na taxa média da turma. Objetivando incentivar a entregar de um trabalho quando o prazo estiver vencendo ou já vencido.