**Integração de sistemas cognitivo e robótico por meio de uma
ontologia para modelar a percepção do ambiente**

Helio Azevedo

Tese de Doutorado do Programa de Pós-Graduação em Ciências de
Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Helio Azevedo

Integração de sistemas cognitivo e robótico por meio de uma ontologia para modelar a percepção do ambiente

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Roseli Aparecida Francelin Romero

USP – São Carlos
Agosto de 2018

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

A994i Azevedo, Helio
Integração de sistema cognitivo e robótico por
meio de uma ontologia para modelar a percepção do
ambiente / Helio Azevedo; orientador Roseli
Aparecida Francelin Romero. -- São Carlos, 2018.
168 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2018.

1. Interação Humano-Robô. 2. Robótica Social. 3.
Ciência Cognitiva. 4. Ontologia Robótica. I.
Aparecida Francelin Romero, Roseli , orient. II.
Título.

Helio Azevedo

Integration of cognitive and robotic systems through an
ontology to model the perception of the environment

Doctoral dissertation submitted to the Institute of
Mathematics and Computer Sciences – ICMC-USP, in
partial fulfillment of the requirements for the degree of
the Doctorate Program in Computer Science and
Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and
Computational Mathematics

Advisor: Profa. Dra. Roseli Aparecida
Francelin Romero

USP – São Carlos
August 2018

*Este trabalho é dedicado a minha esposa Glaucia pela compreensão e incentivo,
ao meu filho Felipe pelo bom humor e entusiasmo,
a minha orientadora Roseli pela paciência e apoio, enfim,
a todos que direta ou indiretamente contribuíram para sua conclusão.*

AGRADECIMENTOS

O esforço necessário para a conclusão de um doutorado não é pequeno. Disciplinas, artigos, busca por recursos, idas e vindas da pesquisa, publicações, mudança de cidade e distanciamento da família são somente algumas das parcelas que compõe o cálculo desse esforço. Não existem atalhos nesse percurso, mas becos sem saída sempre podem surgir e, apesar do conhecimento agregado, acabam gerando atrasos indesejáveis. Considerando esse custo é fácil compreender que a pesquisa não pode ser realizada sem o apoio de outras pessoas ou instituições. Nesse sentido, tenho muito a agradecer à todos que me apoiaram e tornaram essa empreitada viável.

Inicialmente, agradeço à Profa. Dra. Roseli, pela confiança, incentivo e paciência e a todos do Laboratório de Aprendizado de Robôs (LAR) que contribuíram com sugestões durante o desenvolvimento. Em particular, agradeço ao colega José Pedro Ribeiro Belo, que contribuiu diretamente nesta tese com a elaboração de um simulador robótico utilizado durante seu desenvolvimento.

Vale citar o apoio recebido do Núcleo de Robótica e Visão Computacional (NRVC), em particular ao Dr. Josué Junior Guimarães Ramos pelo apoio e contribuições constantes, e à direção do Centro de Tecnologia Renato Archer (CTI) que avalizou meu afastamento para realização do doutorado.

Gostaria também de agradecer a todas as pessoas que estiveram ao meu lado nesse desafio, incluindo minha família e colegas de outros laboratórios.

Por fim, agradeço ao CNPq, à CAPES/MEC, à FAPESP (2017/01687-0) e ao INCT (subvenções CNPq 465755/2014-3 e FAPESP 2014/50851-0) por recursos dispendidos durante a elaboração desta tese.

“Grande coisa é haver recebido do céu uma partícula da sabedoria, o dom de achar as relações das coisas, a faculdade de as comparar e o talento de concluir!”
(Machado de Assis, ”Memórias Póstumas de Brás Cubas“)

RESUMO

AZEVEDO, H. **Integração de sistemas cognitivo e robótico por meio de uma ontologia para modelar a percepção do ambiente**. 2018. 168 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

A disseminação do uso de robôs na sociedade moderna é uma realidade. Do começo restrito às operações fabris como pintura e soldagem, até o início de seu uso nas residências, apenas algumas décadas se passaram. A robótica social é uma área de pesquisa que visa desenvolver modelos para que a interação direta de robôs com seres humanos ocorra de forma natural. Um dos fatores que compromete a rápida evolução da robótica social é a dificuldade em integrar sistemas cognitivos e robóticos, principalmente devido ao volume e complexidade da informação produzida por um mundo caótico repleto de dados sensoriais. Além disso, a existência de múltiplas configurações de robôs, com arquiteturas e interfaces distintas, dificulta a verificação e repetibilidade dos experimentos realizados pelos diversos grupos de pesquisa. Esta tese contribui para a evolução da robótica social ao definir uma arquitetura, denominada *Cognitive Model Development Environment* (CMDE) que simplifica a conexão entre sistemas cognitivos e robóticos. Essa conexão é formalizada com uma ontologia, denominada *OntPercept*, que modela a percepção do ambiente a partir de informações sensoriais captadas pelos sensores presentes no agente robótico. Nos últimos anos, diversas ontologias foram propostas para aplicações robóticas, mas elas não são genéricas o suficiente para atender completamente às necessidades das áreas de robótica e automação. A formalização oferecida pela *OntPercept* facilita o desenvolvimento, a reprodução e a comparação de experimentos associados a robótica social. A validação do sistema proposto ocorre com suporte do simulador *Robot House Simulator* (RHS), que fornece um ambiente onde, o agente robótico e o personagem humano podem interagir socialmente com níveis crescentes de processamento cognitivo. A proposta da CMDE viabiliza a utilização de qualquer sistema cognitivo, em particular, o experimento elaborado para validação desta pesquisa utiliza Soar como arquitetura cognitiva.

Em conjunto, os elementos: arquitetura CMDE, ontologia *OntPercept* e simulador RHS, todos disponibilizados livremente no GitHub, estabelecem um ambiente completo que propiciam o desenvolvimento de experimentos envolvendo sistemas cognitivos dirigidos para a área de robótica social.

Palavras-chave: Interação Humano-Robô, Robótica Social, Ciência Cognitiva, Ontologia Robótica.

ABSTRACT

AZEVEDO, H. **Integration of cognitive and robotic systems through an ontology to model the perception of the environment**. 2018. 168 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

The use of robots in modern society is a reality. From the beginning restricted to the manufacturing operations like painting and welding, until the beginning of its use in the residences, only a few decades have passed. Social robotics is an area that aims to develop models so that the direct interaction of robots with humans occurs naturally. One of the factors that compromises the rapid evolution of social robotics is the difficulty in integrating cognitive and robotic systems, mainly due to the volume and complexity of the information produced by a chaotic world full of sensory data. In addition, the existence of multiple configurations of robots, with different architectures and interfaces, makes it difficult to verify and repeat the experiments performed by the different research groups. This research contributes to the evolution of social robotics by defining an architecture, called Cognitive Model Development Environment (CMDE), which simplifies the connection between cognitive and robotic systems. This connection is formalized with an ontology, called *OntPercept*, which models the perception of the environment from the sensory information captured by the sensors present in the robotic agent. In recent years, several ontologies have been proposed for robotic applications, but they are not generic enough to fully address the needs of robotics and automation. The formalization offered by *OntPercept* facilitates the development, reproduction and comparison of experiments associated with social robotics. The validation of the proposed system occurs with support of the Robot House Simulator (RHS), which provides an environment where the robotic agent and the human character can interact socially with increasing levels of cognitive processing. All together, the elements: CMDE architecture, *OntPercept* ontology and RHS simulator, all freely available in GitHub, establish a complete environment that allows the development of experiments involving cognitive systems directed to the area of social robotics.

Keywords: Human-Robot Interaction, Social Robotics, Cognitive Science, Robotic Ontology.

LISTA DE ILUSTRAÇÕES

Figura 1 – <i>Cognitive Model Development Environment (CMDE)</i>	27
Figura 2 – Expressividade das linguagens	33
Figura 3 – Comparação: (a) língua humana (b) artificial	40
Figura 4 – Voo ondulado e direto de mariposa seguindo pluma de feromônio	42
Figura 5 – Robô móvel controlado por mariposa e sistema de dutos para sucção de odor	45
Figura 6 – Mecanorreceptores da pele humana e estímulos associados	48
Figura 7 – Uso de feedback visual e tátil para movimentar maçaneta de porta	50
Figura 8 – Robôs cobertos por peles artificiais: iCub, Kaspar e NAO	50
Figura 9 – Uso de visão em agricultura	53
Figura 10 – Taxonomia para métricas envolvendo HRI	60
Figura 11 – Psicologia X psicologia cognitiva X ciência cognitiva.	63
Figura 12 – Taxonomia de arquiteturas cognitivas	68
Figura 13 – Sistema SIGVerse	70
Figura 14 – <i>OntPercept</i> e ontologias superiores	76
Figura 15 – Tela principal da ferramenta Protégé representando a <i>OntPercept</i>	84
Figura 16 – Relacionamentos da <i>OntPercept</i> com ontologias de nível superior	85
Figura 17 – Relacionamentos associados a uma instância da classe <i>RobotSmell</i>	87
Figura 18 – Detalhes das propriedades de uma instância da classe <i>RobotSmell</i>	88
Figura 19 – Visão geral dos nós de processamento associados ao <i>OntPercept</i>	88
Figura 20 – Visão geral das APIs associadas ao <i>OntPercept</i>	90
Figura 21 – Diagrama de classes para API de interface com a <i>OntPercept</i>	91
Figura 22 – Página de interface da <i>triple store</i> Fuseki	97
Figura 23 – Módulos do simulador RHS	98
Figura 24 – Visão geral do ambiente disponibilizado pelo RHS	101
Figura 25 – Avatar humano e agente robótico	102
Figura 26 – Tela da ferramenta “VisualSoar”	104
Figura 27 – Mapa de dados do Soar	105
Figura 28 – Detalhe dos atributos presentes na percepção do ambiente	105
Figura 29 – Diagrama de classe do programa “CogArc”	106
Figura 30 – Diagrama de sequência para a execução do pedido: “ <i>get cracker</i> ”	108
Figura 31 – Tela do simulador durante a execução do experimento	109
Figura 32 – Cognitive Model Development Environment - CMDE	110
Figura 33 – Diagrama de classes da API <i>OntPercept</i>	142

Figura 34 – Diagrama detalhado de classes do programa “CogArc” 144

LISTA DE QUADROS

Quadro 1 – Classificação de ontologias	34
Quadro 2 – A Internet como referência para ontologias	38
Quadro 3 – Diretórios para ontologias	39
Quadro 4 – Classificação das técnicas de localização de odor - baseado em <i>taxis</i>	43
Quadro 5 – Domínios de aplicações x atributos de HRI	60
Quadro 6 – Propriedades do conceito <i>RobotPerceptionEvent</i>	77
Quadro 7 – Propriedades associadas ao conceito <i>RobotTaste</i>	78
Quadro 8 – Propriedades associadas ao conceito <i>RobotSmell</i>	79
Quadro 9 – Propriedades associadas ao conceito <i>RobotHear</i>	80
Quadro 10 – Propriedades associadas ao conceito <i>RobotTouch</i>	81
Quadro 11 – Propriedades associadas ao conceito <i>Object</i>	82
Quadro 12 – Variações de terminologia utilizadas pela comunidade	83
Quadro 13 – Comandos disponibilizados pelo RHS	100
Quadro 14 – Regras de produção Soar	107
Quadro 15 – Ativação do programa “CogArc” via terminal Linux	111

LISTA DE CÓDIGOS-FONTE

Código-fonte 1 – Exemplos de elementos em linguagem Manchester	86
Código-fonte 2 – Exemplo da inserção de objeto na <i>triple store</i>	90
Código-fonte 3 – Código da <i>OntPercept</i> em linguagem Manchester	145
Código-fonte 4 – Código de proposição de operadores para o Soar	160

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CLR	<i>Common Language Runtime</i>
CMDE	<i>Cognitive Model Development Environment</i>
COGAT	<i>Cognitive Atlas Ontology</i>
CST	<i>Cognitive Systems Toolkit</i>
CTI	Centro de Tecnologia Renato Archer
DARPA	<i>Defense Advanced Research Projects Agency</i>
DL	<i>Description Logic</i>
DVS	<i>Dynamic Vision Sensor</i>
EAP	<i>Electroactive Polymers</i>
fMRI	<i>Functional Magnetic Ressonance Imaging</i>
FPGA	<i>Field Programmable Gate Array</i>
GPL	<i>General Public License</i>
GRBFN	<i>geometric RBF networks</i>
HMM	<i>Hidden Markov Model</i>
HPLC	<i>High-Performance Liquid Chromatography</i>
HRI	<i>Human-Robot Interaction</i>
ICMC	Instituto de Ciências Matemáticas e de Computação
IEC	<i>International Electrotechnical Commission</i>
IHC	Interação Humano-Computador
IRI	<i>Internationalized Resource Identifier</i>
ISE	<i>Ion Selective Electrode</i>
ISO	<i>International Organization for Standardization</i>
JNI	<i>Java Native Interface</i>
JVM	<i>Java Virtual Machine</i>
LAR	Laboratório de Aprendizado de Robôs
LDA	<i>Linear Discriminant Analysis</i>
LPO	Lógica de Primeira Ordem
MECA	<i>Multi-purpose Enhanced Cognitive Architecture</i>
NRVC	Núcleo de Robótica e Visão Computacional
ORA	<i>Ontologies for Robotics and Automation</i>

ORA-WG	<i>Ontologies for Robotics and Automation Working Group</i>
ORO	<i>Open Robots Ontology</i>
OWL	<i>Web Ontology Language</i>
OWRL	<i>Object World Representation Language</i>
QAF	<i>Quaternionic Atomic Function</i>
QTC	<i>Quantum Tunneling Composite</i>
RDF	<i>Resource Description Framework</i>
REST	<i>Representational State Transfer</i>
RGB-D	<i>Red Blue Green and Depth information</i>
RHS	<i>Robot House Simulator</i>
ROS	<i>Robot Operating System</i>
RoSta	<i>Robot Standards and Reference architecture</i>
SAR	<i>Socially Assistive Robotics</i>
SI	<i>Système International d'unités</i>
SIR	<i>Socially Interactive Robots</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
SML	<i>Soar Markup Language</i>
SOA	<i>Service Oriented Architecture</i>
SPARQL	<i>Simple Protocol and RDF Query Language</i>
SQuaRE	<i>Software product Quality Requirements and Evaluation</i>
SUMO	<i>Suggested Upper Merged Ontology</i>
UGV	<i>Unmanned Ground Vehicle</i>
UML	<i>Unified Modeling Language</i>
URI	<i>Uniform Resource Identifier</i>
USP	<i>Universidade de São Paulo</i>
WTEC	<i>World Technology Evaluation Center</i>

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Objetivos	28
1.2	Contribuições, limitações e estrutura do documento	29
2	ONTOLOGIAS ROBÓTICAS E SENTIDOS DOS ROBÔS	31
2.1	Ontologia robótica	32
2.1.1	<i>Trabalhos relacionados</i>	34
2.1.2	<i>Ontologia e internet</i>	37
2.2	Percepção do ambiente	38
2.2.1	<i>Sentido do robô: paladar</i>	39
2.2.2	<i>Sentido do robô: olfato</i>	41
2.2.3	<i>Sentido do robô: audição</i>	46
2.2.4	<i>Sentido do robô: tato</i>	48
2.2.5	<i>Sentido do robô: visão</i>	51
2.3	Considerações finais	54
3	HRI E COGNIÇÃO	55
3.1	<i>Human-robot interaction</i>	56
3.1.1	<i>Classificação: autonomia</i>	57
3.1.2	<i>Classificação: atributo da aplicação</i>	58
3.1.3	<i>Métricas para HRI</i>	59
3.2	Ciência cognitiva	62
3.2.1	<i>Tópicos presentes na ciência cognitiva</i>	63
3.3	Arquiteturas cognitivas	67
3.4	Considerações finais	72
4	DESENVOLVIMENTO	73
4.1	Metodologia	74
4.2	O projeto da <i>OntPercept</i>	75
4.2.1	<i>Definição dos conceitos</i>	76
4.2.1.1	<i>Conceito RobotPerceptionEvent</i>	77
4.2.1.2	<i>Conceito RobotTaste</i>	77
4.2.1.3	<i>Conceito RobotSmell</i>	79
4.2.1.4	<i>Conceito RobotHear</i>	80

4.2.1.5	<i>Conceito RobotTouch</i>	81
4.2.1.6	<i>Conceito RobotVision</i>	81
4.2.1.7	<i>Conceito Object</i>	81
4.2.2	Construindo a ontologia	82
4.3	Provendo interfaces para <i>OntPercept</i>	87
4.3.1	API <i>ontPerceptCSharp</i>	89
4.3.2	API <i>ontPerceptJava</i>	92
4.4	Considerações finais	92
5	VERIFICAÇÃO & VALIDAÇÃO	95
5.1	Simulador <i>Robot House Simulator</i> (RHS)	96
5.1.1	<i>Características do RHS</i>	98
5.1.2	<i>Comandos de atuação sobre o humanoide</i>	99
5.1.3	<i>Elementos do ambiente de simulação</i>	100
5.1.4	<i>Percepção do ambiente</i>	102
5.2	Modelando arquitetura cognitiva	103
5.3	Descrição do experimento	107
5.4	Considerações finais	110
6	CONCLUSÃO	113
6.1	Trabalhos futuros	115
6.2	Divulgação dos resultados	116
REFERÊNCIAS		119
GLOSSÁRIO		137
APÊNDICE A DETALHES DE MODELOS DO SISTEMA		142
A.1	Diagrama detalhado de classes da API <i>ontPercept</i>	142
A.2	Diagrama detalhado de classes do programa “CogArc”	144
A.3	<i>ontPercept</i> representado em linguagem Manchester	145
A.4	Soar: proposição de operadores	160
ANEXO A CRÍTICAS À WEB SEMÂNTICA		167

INTRODUÇÃO

Atualmente, os robôs são capazes de movimentos com precisão milimétrica executando tarefas repetitivas operando em [ambientes estruturados](#) onde os objetos estão em locais conhecidos e previsíveis.

Assim, não é de surpreender que robôs sejam mais usados em operações fabris, como pintura e soldagem, em vez de operações onde, a diversidade de ações, o contato direto com os seres humanos e alterações do ambiente façam parte dos requisitos do sistema. Apesar de sua complexidade, a demanda por utilização de agentes robóticos em ambientes distintos da manufatura é uma necessidade da sociedade moderna. O relatório *World Robotics 2015* ([IFR, 2015](#)) indica que as vendas globais de robôs de serviço para uso privado irão aumentar para cerca de 35 milhões de unidades no período de 2015-2018 movimentando mais de US\$ 12 bilhões.

Essa tendência de aumento na utilização de robôs de serviço requer como contrapartida um crescimento nas pesquisas envolvendo interações entre humanos e robôs (*Human-Robot Interaction* (HRI)), em particular na subárea definida por Fong ([FONG; NOURBAKHSH; DAUTENHAHN, 2003](#)) como *Socially Interactive Robots* (SIR) ou “Robótica Social”, termo utilizado nesta tese.

Uma área básica para a consolidação da robótica social é a ciência cognitiva. Essa área possui grupos de pesquisadores motivados e envolvidos diretamente na construção de modelos que representem a complexa dinâmica presente no relacionamento e atuação humana. Vale ressaltar que essa não é uma tarefa restrita a roboticistas, pelo contrário, psicólogos, linguistas e filósofos também participam deste grande desafio.

Um dos fatores que dificultam as pesquisas em robótica social é a existência de múltiplas plataformas de robôs, muitas das quais proprietárias, fato que minimiza a troca de conhecimentos e habilidades entre os pesquisadores. Além da questão da diversidade de robôs, temos também inúmeros arcabouços de programação com arquiteturas e interfaces distintas. Esses fatores acabam subtraindo recursos e postergando a obtenção de resultados.

Outro ponto crítico é a falta de disponibilidade de ambientes de referência para desenvolver, avaliar e comparar o desempenho de uma aplicação específica. Uma interessante proposta apresentada no relatório “*A Roadmap for U.S. Robotics: From Internet to Robotics*” (ROBOTICS, 2013) propõe uma cidade de robôs (*Robot City*) elaborada a partir de um ambiente urbano comum em que todos os habitantes fazem parte do experimento e ajudam no processo de avaliação, bem como, na definição de requisitos adequados para aplicação cotidiana.

Até a presente data, a literatura não registra um ambiente com as características imaginadas para o *Robot City*. Um cenário semelhante foi criado pela Google para testar seus carros autônomos (AUSTIN, 2017). Neste caso, foi construída uma cidade no deserto da Califórnia, onde são desenvolvidos cenários que desafiam os carros autônomos a responderem de forma aceitável a situações elaboradas em ambientes compostos por cones, sinalização, manequins, outros carros e até mesmo pedestres humanos participantes do experimento. Outros cenários para testes englobam os criados pela *Defense Advanced Research Projects Agency* (DARPA) para o *Grand Challenge* (DARPA, 2014) e *The Robotics challenge* (DARPA, 2015). Entretanto, todos eles apresentam um custo elevado e inacessível para a maioria dos grupos de pesquisa da área.

Essas questões representam um desafio no processo de acelerar o desenvolvimento da robótica social envolvendo a verificação e repetibilidade dos experimentos. Segundo Chrysostomou et al. (CHRYSTOMOU *et al.*, 2017), as métricas atuais, configurações de cenário, corpora e base de dados aplicados em experimentos HRI são adaptadas a casos individuais e, portanto, não são adequados para comparar e obter métricas de resultados em laboratórios e sistemas.

Uma estratégia alternativa associada a ambientes de referência é o uso de simuladores em vez de cidades emuladas. Uma cidade emulada duplica de forma controlada um ambiente real, permitindo a realização de experimentos fidedignos com riscos limitados embora com alto custo. Outra estratégia envolve o uso de uma cidade simulada que disponibiliza um ambiente semelhante a uma cidade emulada, mas totalmente implementada em um sistema computacional com um custo muito inferior.

No Laboratório de Aprendizado de Robôs (LAR), do Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP), existe uma linha de pesquisa que visa a elaboração de um sistema direcionado para atividades de robótica social envolvendo ciência cognitiva, HRI e ontologia robótica. Esta tese contribui diretamente com essa linha de pesquisa disponibilizando uma arquitetura, denominada *Cognitive Model Development Environment* (CMDE). O objetivo do CMDE é acelerar o processo de desenvolvimento e reduzir o esforço na fase de validação definindo uma interface clara entre o sistema cognitivo e os sensores e atuadores presentes no agente robótico.

A Figura 1 apresenta a arquitetura CMDE onde podemos identificar dois nós de processamento, o primeiro, denominado “Sistema Cognitivo”, recebe informações de percepção, processa

e atua no ambiente realimentando o sistema robótico com instruções. O segundo nó, controla os sensores e atuadores presentes no agente robótico, seja ele real ou virtual. Como exemplos de instanciação de agente robótico podemos citar: humanoide, recepcionista virtual e simuladores. Na [Figura 1](#), esse nó é representado pelo “Simulador Robótico”, que simula um robô humanoide atuando em [ambientes não estruturados](#). Naturalmente, após a exaustão dos testes com simulador este nó de processamento é substituído pelo controle dos sensores e atuadores presentes em um “robô real” apresentado na cor azul clara nessa figura.

Figura 1 – *Cognitive Model Development Environment (CMDE)*

Fonte: Elaborada pelo autor.

Outro elemento presente na [Figura 1](#) é a interface de comunicação entre o “Sistema Cognitivo” e o nó de processamento responsável pelo controle do agente robótico. Esse interface deve formalmente descrever os conceitos necessários para representar a percepção do ambiente realizada pelos sensores do agente robótico. Um caminho para atender a essa demanda é através da utilização de ontologias.

1.1 Objetivos

O objetivo principal desta pesquisa é contribuir para a área de robótica social formalizando a comunicação entre o sistema cognitivo e o sistema robótico presentes na arquitetura CMDE. Essa formalização simplifica a utilização de arquiteturas robóticas, catalisando o aumento de sua utilização na área de robótica.

A formalização proposta é alcançada com a definição de uma ontologia¹, denominada *OntPercept*. Nossa visão é da percepção mapeada em diferentes camadas de abstração, sendo a mais baixa associada aos dados brutos obtidos dos sensores e a mais alta, associada a representação da percepção pelo sistema que implementa a cognição. A ontologia *OntPercept* se posiciona nos níveis superiores estabelecendo uma interface de alto nível para a implementação de modelos cognitivos.

Essa abordagem facilita testes e análises comparativas entre diversas arquiteturas cognitivas contribuindo para uma inserção natural de agentes robóticos na sociedade. Ao longo dos últimos anos, diversas ontologias foram propostas para aplicações robóticas. Entretanto, segundo [Prestes et al. \(2013\)](#), essas ontologias são muito específicas para atender completamente as necessidades das áreas de robótica e automação.

A ontologia *OntPercept* é aderente a norma 1872-2015 - *IEEE Standard Ontologies for Robotics and Automation* ([IEEE, 2015](#)). A utilização da IEEE 1872-2015 como ontologia base, recomenda que todos os termos e exemplos sejam apresentados no idioma inglês como forma de viabilizar a ampla divulgação dos resultados alcançados. As vantagens da definição da ontologia *OntPercept* consistem em:

- estabelecer uma interface padronizada entre o sistema cognitivo e robótico. Essa interface reduz o esforço do especialista em ciência cognitiva em questões como: movimento, visão, controle de graus de liberdade, etc. Essas questões desviam o pesquisador do foco principal, ou seja: capacitar o robô a interagir socialmente através da construção de modelos que representem as atividades a serem executadas no ambiente alvo;
- facilitar o desenvolvimento de simuladores robóticos pela definição de características mínimas para representar a percepção do ambiente;
- redução do esforço no desenvolvimento de sistemas envolvendo ciência cognitiva e robótica social;
- viabilizar a construção de ambientes de referência para arquiteturas cognitivas permitindo a reprodução de experimentos realizados em laboratórios distintos;
- facilitar o acesso ao imenso volume de informação disponível na Web, com o uso da tecnologia de [Web Semântica](#). No final de 2016, esse volume foi estimado como 1 bilhão de *websites* com tráfego de 1.1 *zettabytes*² por ano ([COLLINS, 2016](#)). Ignorar essa massa

¹ O tópico ontologia é aprofundado na [Seção 2.1](#).

² 1 zettabyte = 1 bilhão de terabytes.

de informação representa um grave erro. A **Web Semântica** viabiliza estruturar e acessar esses dados com o apoio de ontologias. Nesse cenário, a utilização de ontologias robóticas representa uma estratégia fundamental para viabilizar a inserção de sistemas robóticos em **ambientes não estruturados** constituídos por inúmeros objetos “desconhecidos” pelo agente robótico mas, facilmente identificáveis na Web.

Vale ressaltar que no levantamento bibliográfico apresentado na **Seção 2.1** não foi constatada uma ontologia com essas características na literatura.

1.2 Contribuições, limitações e estrutura do documento

Esta tese apresenta duas contribuições originais para a área de robótica social. A primeira consiste na definição da ontologia *OntPercept*, que estabelece um protocolo para a comunicação entre os sistemas cognitivo e robótico definidos pela arquitetura *Cognitive Model Development Environment* (CMDE).

A segunda consiste na definição dos requisitos do aplicativo *Robot House Simulator* (RHS), um simulador capaz de modelar aspectos necessários para robótica social. A definição dos requisitos viabilizou o desenvolvimento do simulador no escopo de uma dissertação de mestrado (BELO, 2018b) realizada no LAR/ICMC/USP – São Carlos. A orientação dessa dissertação foi realizada pela Professora Roseli Ap. F. Romero, sendo minha responsabilidade atuar como coorientador. Essa intersecção garantiu uma evolução consistente das atividades na medida que integrou pesquisadores no desenvolvimento de um ambiente complexo em evolução no laboratório.

Esta tese representa um ciclo completo de desenvolvimento que envolve desde a elaboração dos requisitos de arquitetura e ontologia, seu modelamento, desenvolvimento e, finalmente, a criação de ambiente de validação com o apoio do simulador RHS. Todos os itens de configuração associados a esse processo foram liberados de acordo com a licença GNU *General Public License* (GPL) v3.0 (Free Software Foundation, 2018) no repositório GitHub: <<https://github.com/helioaz/ontPerception>>.

Esses resultados representam os primeiros passos na elaboração de um ambiente para desenvolvimento de sistemas cognitivos com aplicação em robótica social. Nesse contexto, as principais limitações são representadas por etapas ainda em desenvolvimento associadas a este ambiente:

- a ontologia proposta é unidirecional, ou seja, explora os sentidos, mas não abrange as ações a serem executadas pelo agente robótico. A extensão de ontologia para representar tarefas e atividades moldadas às capacidades disponíveis nos robôs representa um passo importante na consolidação do *OntPercept*;

- a validação da ontologia foi realizada utilizando a arquitetura cognitiva Soar (LAIRD, 2012; LAIRD *et al.*, 2017). A elaboração de interfaces para outras arquiteturas cognitivas, como ACT-R (ACT, 2015), por exemplo, representa um passo importante na consolidação dos resultados;
- acréscimo de outros sensores, além dos associados aos sentidos clássicos: visão, paladar, olfato, audição e tato.

Esta tese está organizada do seguinte modo. No [Capítulo 2](#), é apresentado o estado da arte das pesquisas em ontologia e sistemas de sensoriamento para agentes robóticos. Esse capítulo oferece a base para o estabelecimento dos conceitos presentes na ontologia. No [Capítulo 3](#), são exploradas áreas correlatas à evolução da pesquisa, a saber: HRI, Ciência Cognitiva e Ontologia. Esse capítulo posiciona a pesquisa no contexto da robótica social e das necessidades para viabilizar a interação do agente robótico com seres humanos. A definição clara dos objetivos, associada ao embasamento teórico construído nos Capítulos 2 e 3 viabilizam o desenvolvimento realizado no [Capítulo 4](#). A [verificação](#) e [validação](#) dos resultados alcançados são detalhadas no [Capítulo 5](#). Finalmente, no [Capítulo 6](#), a conclusão é apresentada com os principais resultados e futuras evoluções.

ONTOLOGIAS ROBÓTICAS E SENTIDOS DOS ROBÔS

Este capítulo apresenta a revisão de duas áreas básicas diretamente utilizadas na realização desta pesquisa: percepção sensorial do ambiente por um agente robótico e ontologia robótica.

A percepção sensorial do ambiente envolve capturar, processar e organizar dados coletados por sensores visando criar um modelo para representar o ambiente. Esse modelo representa a base para que o sistema cognitivo infira a ação mais apropriada para cada situação.

A organização formal das informações sensoriais capturadas no ambiente é realizada através da ontologia. É importante distinguir ontologia de taxionomia. A taxionomia utiliza hierarquias, comumente representadas por estrutura em árvore, para classificar objetos, como por exemplo, tipo, família, gênero, espécie, etc. Entretanto, não formaliza relacionamentos nem estabelece rigidez formal na classificação inviabilizando realizar inferências sobre a informação disponível. Como o objetivo final desta tese é a utilização de ontologia em agentes robóticos, o foco de nossa revisão foi direcionado para ontologias voltadas ao domínio da robótica, esse grupo é identificado neste tese como “ontologia robótica”.

2.1 Ontologia robótica

Uma ontologia descreve formalmente objetos e seus relacionamentos em um domínio de conhecimento. [Bernus et al. \(2009\)](#) fornece uma definição comumente aceita para o termo:

Uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada. “Explícita” se refere ao fato de que todos os elementos de uma ontologia são explicitamente definidos, ao passo que “formal” significa que a especificação da ontologia é dada em uma linguagem com sintaxe formal e semântica resultando em uma descrição interpretável e executável por uma máquina. Finalmente, “compartilhada” capta o aspecto que uma ontologia representa conhecimento consensual que foi acordado por um grupo de pessoas, normalmente como resultado de um processo social.

A elaboração de uma ontologia envolve aspectos que determinam sua representação, ferramentas e infraestrutura necessária para seu desenvolvimento. Esses aspectos são apresentados nos próximos itens.

- **Engenharia Ontológica:** trata de metodologias e processos para elaboração de ontologias. *Methontology* ([FERNÁNDEZ; GÓMEZ-PÉREZ; JURISTO, 1997](#)) representa um exemplo de metodologia para construção de ontologias pelo reuso de outras ontologias em seu estado natural, ou por processos de reengenharia. Outro exemplo de metodologia de apoio é a *OntoClean* ([GUARINO, 2009](#)) que investiga a adequação e coerência lógica dos relacionamentos estabelecidos entre os conceitos da ontologia.
- **Infraestrutura:** a escalabilidade da tecnologia ontológica é crucial para a sua utilização comercial. De seu início, na década de 90, que demandava a gerência de algumas centenas de entidades até a atual necessidade de gerenciar bilhões de triplas *Resource Description Framework* (RDF) somente alguns anos transcorreram ([BERNUS et al., 2009](#)). Este crescimento passou a exigir alta escalabilidade na infraestrutura ontológica, em particular mecanismos de armazenamento e recuperação de informação. Como exemplo, [Hertel, Broekstra e Stuckenschmidt \(2009\)](#) apresentam diferentes modelos elaborados para manipular dados RDF com alta escalabilidade.
- **Representação da Ontologia:** de acordo com a definição de [Bernus et al. \(2009\)](#), a linguagem para representar a ontologia necessita ser explícita e formal. Uma primeira abordagem para essa representação formal foi o uso de lógica de predicados, em particular, a lógica de primeira ordem ([BAADER; HORROCKS; SATTLER, 2009](#)). Na mesma linha de representação lógica, F-logic (Frame Logic) habilita a associação de conceitos com elementos do modelo de orientação a objetos ([ANGELE; KIFER; LAUSEN, 2009](#)). Lembrando que compartilhamento de informação representa um conceito central em ontologias, outra estratégia é o mapeamento de conceitos em representações utilizadas pela Web que, atualmente, é o meio mais utilizado para difusão de informação e conhecimento.

A **RDF** (**RDF-TEAM, 2016**) apresenta um modelo leve para tráfego de informações que permite o uso de *Uniform Resource Identifier* (URI) para mapear relacionamento entre nós presentes na Web, essa estrutura é definida como uma “tripla **RDF**”. Outra tecnologia utilizada é a *Web Ontology Language* (OWL) (**ANTONIOU; HARMELEN, 2009**), construída sobre **RDF** para fornecer uma linguagem ontológica mais expressiva que permite inferências adicionais, como igualdade e desigualdade, restrições de cardinalidade, existência de objetos entre outros. A **OWL** é preferencialmente utilizada para representar informação na *Web Semântica* (**ANTONIOU et al., 2012**). Segundo **Pease (2011)**, a expressividade da linguagem utilizada para representar o conhecimento está diretamente associada a rigidez de sua representação, contudo, quanto maior a expressividade maior o custo de processamento associado (**Figura 2**).

Figura 2 – Expressividade das linguagens

Fonte: Adaptada de **Pease (2011, p. 33)**.

O amadurecimento da representação, da engenharia e da infraestrutura permitiu uma explosão no desenvolvimento de ontologias (**BATEMAN, 2016**). A necessidade de organizar conceitos e relacionamentos acabou disseminando o uso de ontologias em diversas áreas, como por exemplo: linguística, geografia, conhecimento, negócios, software, cognição (**POLDRACK, 2016**) e biologia e medicina (**BIOPORTAL, 2016**). Essa explosão gerou uma busca por definições de hierarquias e tipos para classificar as ontologias criadas.

Naturalmente, a diversidade de definições existentes associada à atualidade da área acabam por gerar mais de um tipo de classificação dependendo da visão do autor da proposta. O **Quadro 1** apresenta algumas destas classificações (**GRIGONYTE, 2010**):

Quadro 1 – Classificação de ontologias

Autor	Ano	Critério de classificação
Van Heijst	1997	de acordo com o problema que a ontologia se propõe a representar: terminológico, informação e conhecimento.
Guarino	1998	nível de generalidade da ontologia.
Goméz-Péres	2004	riqueza da estrutura interna: leves (não incluem axiomas) e pesadas (incluem axiomas).
Fensel	2004	generalidade e papel da ontologia na construção de sistemas de conhecimento: domínio, metadados, genérica, representação e tarefa.

Fonte: Elaborada pelo autor.

Nesta pesquisa será adotada a classificação, proposta por [Guarino \(1998\)](#), que tipifica as ontologias em:

- **Nível superior (*top-level*):** descrevem conceitos muito gerais (como espaço, tempo, matéria, objeto, evento, ação, etc.) independentes de um problema ou domínio particular. A ideia é, ao menos em teoria, alcançar um nível superior unificado, ou seja, ontologias para grandes comunidades de usuários.
- **Domínio:** especializam definições das ontologias de nível superior construindo conceitos de um domínio genérico, como medicina ou automóveis.
- **Tarefas:** descrevem tarefas ou atividades genéricas, como diagnosticar ou vender, especializando os termos introduzidos na ontologia de nível superior.
- **Aplicação:** estritamente relacionadas a um domínio ou tarefa específica, correspondendo a funções desempenhadas por entidades de domínio durante a execução de uma atividade determinada.

A escolha da classificação proposta por Guarino ocorre principalmente por ela ser adotada pela norma [IEEE \(2015\)](#), utilizada como base no desenvolvimento desta tese. O objetivo final desta pesquisa é a elaboração de ontologia que reflita os sentidos presentes em um robô, assim, nossa ontologia é do tipo “Aplicação”.

2.1.1 Trabalhos relacionados

Em Robótica, o uso de ontologia contribui de diversas formas ([HAIDEGGER et al., 2013a](#)):

- fornece um conjunto consistente de terminologia (vocabulário do domínio), conceitos, definições, relações e axiomas;
- habilita a descrição procedural de tarefas complexas;
- fornece um repositório de conhecimento que pode ser compartilhado por diversos robôs;
- produz novas relações através da análise de dados gerados com o uso de ontologias.

Considerando a área de robótica, o projeto RoboEarth ([ROBOEARTH-TEAM, 2015](#)), financiado pela União Europeia de 2010 à 2014, tem como objetivo construir uma internet para robôs através de plataforma aberta que possibilite qualquer agente robótico com acesso a rede a gerar, compartilhar e reutilizar dados. Um dos resultados foi a disponibilização de uma ontologia representada em [OWL](#) e construída com conceitos base extraídos da base OpenCyc ([CYCORP, 2016](#)). Todas as partículas de informação presentes no RoboEarth são “semanticamente anotadas”, ou seja, são associadas com uma ontologia e descritas por uma linguagem lógica e formal.

O trabalho de [Riazuelo et al. \(2015\)](#) associa a base de conhecimento presente no RoboEarth com um sistema visual SLAM para disponibilizar uma percepção precisa e contínua do ambiente. Um robô Pioneer P3-DX é utilizado para construir e explorar mapas semânticos necessários para executar ações cotidianas com o uso do serviço de nuvem disponibilizado pelo RoboEarth. O robô realiza suas tarefas baseado em *action recipes* disponibilizadas pela base de conhecimento presente na nuvem. Essas “receitas” especificam de forma abstrata quais ações devem ser seguidas para completar uma tarefa selecionada de forma independente do robô e do ambiente. Naturalmente, após o download a receita deve ser customizada para as características do robô e do ambiente.

Na mesma linha da *action recipes*, [Ji et al. \(2012\)](#) apresenta uma estrutura genérica para a geração automatizada de ações em ambientes dinâmicos utilizando ontologia semântica descrita em [OWL](#). A proposta é construir uma ontologia, utilizando como base o sistema KnowRob ([KNOWROB, 2016](#)), para o planejamento de tarefas em um cenário real de robôs de serviço. Um robô, denominado *Care-O-Bot*, é usado como plataforma robótica e o arcabouço *Robot Operating System* (ROS) como plataforma de software.

Como exemplo de ontologia voltada para um domínio específico temos a APIbot ([ZESHAN et al., 2014](#)) direcionada para robôs móveis autônomos. Essa ontologia foi validada em uma aplicação para controle de incêndio com a utilização de sensores presentes em um robô e conectados via rede sem fio com um servidor externo. O servidor fornece um conjunto de serviços aderentes a arquitetura *Service Oriented Architecture* (SOA) com o apoio de *Web Services*.

Uma relevante iniciativa normativa é a definição de ontologia para robótica e automação como um padrão IEEE. Em 2012, a *IEEE Robotics and Automation (R&A) Society* criou um grupo de trabalho denominado *Ontologies for Robotics and Automation Working Group* (ORAWG) ([PRESTES et al., 2013](#); [SCHLENOFF et al., 2012](#)) que compreende dois subgrupos: um para a área de Automação e outro para Serviços ([HAIDEGGER et al., 2013b](#)). Os esforços desse grupo para atender as necessidades dos domínios de robótica e automação produziram em 2015 a norma *1872-2015 IEEE Standard Ontologies for Robotics and Automation* ([IEEE, 2015](#)). Essa norma define quatro ontologias ([FIORINI et al., 2015](#)):

CORA: ontologia núcleo da norma direcionada para Robótica e Automação (R&A). Esta

ontologia representa o nível mais alto de abstração sob a qual outros grupos podem desenvolver ontologias específicas.

CORAX: conceitos comuns presentes em R&A;

RPARTS: conceitos que representam partes do robô;

POS: aspectos gerais sobre posição e orientação.

Apesar dos parâmetros da revisão sistema estabelecerem como período de 2012 à 2016, é natural que algumas referências importantes extrapolem esse parâmetro. Infelizmente, os trabalhos associados não evoluíram a ponto de gerar referências recentes sobre o tema. Nesse contexto dois trabalhos merecem destaque¹.

O primeiro trabalho envolve ação financiada pela União Europeia, denominado *Robot Standards and Reference architecture* (RoSta) ([ROSTA-TEAM, 2009](#)). O grupo RoSta criado em 2007 trabalhou durante dois anos produzindo um conjunto de documentos com foco em arquiteturas de referência para robótica de serviços. Os documentos gerados concentram o estado da arte, na época, sobre manipulação móvel e robôs de serviço:

Deliverable 1: criação de um glossário/ontologia;

Deliverable 2: especificação de uma arquitetura de referência;

Deliverable 3: especificação de middleware;

Deliverable 4: formulação de benchmarks (para componentes, métodos, middleware e arquiteturas).

O projeto RoSta (2007-2009) viabilizou o projeto RoboEarth (2010-2014) ([ROBOEARTH-TEAM, 2015](#)) citado anteriormente. Na sequência do RoboEarth a União Europeia financiou o projeto RoboHow (2012-2016) ([ROBOHOW-TEAM, 2016](#)) que tem como objetivo: “habilitar robôs para competentemente executar atividades cotidianas realizadas por humanos em seu ambiente de trabalho ou residência”.

Como segundo trabalho não capturado pela revisão temos [Lemaignan et al. \(2010\)](#) que disponibiliza ao agente robótico um modelo do contexto, papel e relacionamento de objetos presentes no ambiente. Além disso, permite inferir intenções e crenças dos outros agentes. A plataforma aberta desenvolvida com esse objetivo é denominada *Open Robots Ontology* (ORO). Essa plataforma oferece um biblioteca de conhecimento de alto nível, um conjunto de funções cognitivas além de uma abordagem geral para processamento de conhecimento. O conhecimento associado ao ambiente é representado em lógica de primeira ordem, como tripla **RDF** ([RDF-TEAM, 2016](#)). As triplas **RDF** são armazenadas com o apoio de Jena ([JENA-TEAM, 2018](#)), um arcabouço Java para construção de aplicações apoiadas pela tecnologia de web semântica. A ferramenta Jena é utilizada em conjunto com o raciocinador Pellet ([STARDOG-UNION,](#)

¹ Esses trabalhos não foram contabilizados nos gráficos da revisão por alterarem a janela da pesquisa comprometendo o resultado apresentado.

2016) para garantir a classificação contínua do conhecimento. Durante o período de execução, declarações são permanentemente adicionadas, removidas ou consultadas pelos componentes do robô.

Como contraponto ao uso de lógica de primeira ordem para representar os axiomas da ontologia, temos o trabalho de [Raskin e Taylor \(2015\)](#) que discorre sobre as limitações de seu uso para representar a semântica presente em linguagem natural. Segundo os autores, o problema está associado ao receio que a representação de uma semântica compreensiva inviabilize teste computadorizado de formalidade. Como resultado, o contexto da informação não é representado completamente. Essa limitação pode ser estendida para sensores presentes em um agente robótico onde somente a análise semântica de todos os sensores permite estabelecer uma imagem real do ambiente.

Nosso estudo ressalta a diversidade de ontologias robóticas disponíveis. Esta diversidade é esperada, uma vez que o objetivo da ontologia é representar entidades presentes em um mundo que é intrinsecamente complexo. Por outro lado, não foram localizadas referências associadas a ontologias robóticas que modelem diretamente a percepção do ambiente pelo agente robótico. Vale ressaltar que esse resultado representa uma motivação a mais para a elaboração desta tese.

Apesar da variedade de pesquisas, as abordagens utilizadas para desenvolver, representar e testar as ontologias produzidas acabam revelando algumas preferências, ao menos no período pesquisado. Para representar a ontologia temos OWL/RDF ([WEB-ONTOLOGY-WG, 2016](#)), a construção é modelada com o uso de Protégé ([PROTÉGÉ, 2016](#)), Jena ([JENA-TEAM, 2018](#)) é utilizada para busca e processamento, Pellete ([STARDOG-UNION, 2016](#)) para processamento de raciocínio e o arcabouço ROS ([ROS-TEAM, 2015](#)) como apoio na implementação da interface com robôs.

Por outro lado, o caminho da normatização criando ontologias base contribui para minimizar essa “explosão ontológica” além de prover uma estrutura consistente para operações de mapeamento de conceitos e união de ontologias.

2.1.2 Ontologia e internet

Apesar de não ser resultado de busca nas bases de referência bibliográficas a riqueza de informação presente na internet sobre ontologia demanda um olhar mais detalhado. Essa conexão com a internet não é coincidência, um dos catalisadores para novas pesquisas e surgimento de centenas de ontologias foi a necessidade de compartilhar informação na internet com o apoio da [Web Semântica](#).

O [Quadro 2](#) apresenta uma pequena mostra de ontologias descritas na internet e associadas com esta pesquisa.

Como afirmado anteriormente, temos centenas de páginas que descrevem ontologias para as mais diferentes áreas de atuação: mobiliário, eletrônica, geografia, medicina, robótica,

Quadro 2 – A Internet como referência para ontologias

Ontologia	Tipo	Domínio
BioPortal	Medicina	Uma das mais completas base de dados sobre ontologias para área médica. No escopo desta pesquisa merece destaque a <i>Cognitive Atlas Ontology</i> (COGAT) que caracteriza o pensamento corrente da área médica em ciência cognitiva (BIOPORTAL, 2016).
CogPowiki	Medicina	Apresenta os termos básicos para a elaboração de ontologia de paradigmas cognitivos (COGPOWIKI, 2016).
CORA	Robótica	Ontologia para Robótica e Automação elaborada como norma pelo grupo <i>Ontologies for Robotics and Automation</i> (ORA) da IEEE (IEEE-RAS/SC-ORA, 2016)
Cyc	Geral	A base de conhecimento Cyc disponibiliza ontologia com mais de 600.000 termos atômicos, moldados por um conjunto de cerca de 7 milhões de axiomas elaborados manualmente (CYCORP, 2016).
próxima SUMO	Geral	A ontologia base <i>Suggested Upper Merged Ontology</i> (SUMO) e suas ontologias de domínio formam atualmente a maior base pública disponível gratuitamente. Sua utilização envolve pesquisas e aplicações em busca, linguística e raciocínio. (PEASE, 2016)
DOLCE	Linguística	Um ontologia descritiva para engenharia linguística e cognitiva (LOA-GROUP, 2016)

Fonte: Elaborada pelo autor.

educação, biologia, economia, componentes de industriais, alimentação, esportes, governo, justiça, linguística, militar, cognição, etc. O crescimento de ontologias de domínio com o apoio de *Semantic Web* acabou por revelar um gargalo em sua utilização: a necessidade de mapeamento, fusão e integração de ontologias. Essas demandas são áreas de pesquisa em ebulição que transcendem o escopo desta pesquisa. Atualmente, uma estratégia para gerenciar esse volume de informações é a organização das ontologias por serviços de diretórios organizados em segmentos de interesse. O Quadro 3 apresenta alguns desses diretórios.

2.2 Percepção do ambiente

No contexto desta tese, a percepção do ambiente é interpretada como o conjunto mínimo de informações, necessária e suficiente, para viabilizar a operação de um agente robótico de forma a atender as demandas de um agente humano. As informações de percepção do ambiente são processadas pelo sistema cognitivo que delibera as ações pertinentes.

Uma tendência em pesquisa robótica é incorporar diferentes capacidades sensoriais inspiradas em soluções presentes na natureza. A proposta é permitir que os robôs mimetizem as capacidades humanas de comunicação e exploração ao receberem as mesmas competências presentes nos sentidos humanos, ou seja, ver, ouvir, degustar, cheirar e tocar. Nos agentes robóticos, essa percepção ocorre através da transdução da luz, estímulos inerciais, mecânicos e bioquímicos em sinais elétricos utilizando uma variedade de sensores: fotorreceptores para

visão, mecanorreceptores para audição e quimiorreceptores para o paladar e olfato, além de termorreceptores para temperatura e nociceptores para dor e detecção de danos.

Os próximos itens desta seção apresentam uma breve visão das pesquisas associadas a integração de sistemas robóticos com sensores associados a cada um dos sentidos. O [Capítulo 4](#) utiliza esta visão para elaborar os conceitos presentes na ontologia *OntPercept*. A estratégia adotada consiste em modelar, com o apoio de ontologia e com um alto nível de cognição, as informações de percepção obtidas a partir dos dados brutos gerados pelos sensores.

2.2.1 Sentido do robô: paladar

O paladar permite reconhecer os gostos de substâncias colocadas sobre a língua². Na língua temos as papilas gustativas que identificam substâncias do sabor e enviam a informação ao cérebro. Existem 5 sabores básicos: amargo, ácido, salgado, doce e [umami](#). A língua é a responsável por sentir grande parte dos sabores porque concentra os quimiorreceptores responsáveis

² A língua não é o único órgão associado ao paladar. Papilas presentes no céu da boca também contribuem no processo.

Quadro 3 – Diretórios para ontologias

Diretório	Descrição
Portal	Portal elaborado por John Bateman que apresenta lista de ontologias organizadas por motivação e por domínio de aplicação (BATEMAN, 2016).
DAML	Apresenta lista de ontologias aderentes a tecnologia DAML desenvolvida pelo DARPA para Web Semântica (DAML, 2016).
OBO Foundry	Conjunto de ontologias de referência interoperáveis em biologia e biomedicina. Os participantes aderem e contribuem para o desenvolvimento de um conjunto evolutivo de princípios assumindo as seguintes diretrizes: uso aberto, desenvolvimento colaborativo, conteúdo não-sobrepuesto, sintaxe e relações padronizadas (OBO, 2016).
KnowRob	O sistema de processamento de conhecimento (<i>Knowledge Processing for Robots-KnowRob</i> (KNOWROB, 2016) é composto por um conjunto de ontologias elaboradas com o apoio da linguagem OWL . O sistema contém quatro componentes principais: a base de conhecimento enciclopédica; modelos de ação; instâncias de objetos, ações e eventos; além das classes computacionais. As classes computacionais permitem a consulta de fontes de informação externas ou combinar informações da base para criar novas declarações. Essas ontologias são acessíveis como um pacote disponibilizado no site do arcabouço ROS (ROS-TEAM, 2015).
W3C Sensors	Apresenta 17 ontologias centradas em sensores e em observadores aderentes aos padrões da W3C(W3C-SENSORES, 2016). Note que, “sensores” observam estímulos e produzem valores que são processados por “observadores” que agregam informações de tempo e localização, por exemplo.
Protégé	Contém um conjunto de ontologias desenvolvidas com o arcabouço <i>Protégé</i> sendo 81 descritas em OWL , 19 com <i>Frame-based</i> e 8 em outros formatos (PROTÉGÉ, 2016).

Fonte: Elaborada pelo autor.

pela “identificação” dos vários sabores (Figura 3a).

Figura 3 – Comparação: (a) língua humana (b) artificial

(a) Fonte: CECIERJ (2017)

(b) Fonte: Insent (2017)

O reconhecimento do sabor pelas papilas gustativas é realizado por duas famílias de receptores: receptores acoplados à proteína G (G-protein-coupled receptors–GPCRs) e **canal iônico**. Substâncias doces e **umami** são absorvidas por uma família de três GPCRs: T1R1, T1R2 e T1R3. Compostos amargos por GPCRs do tipo T2Rs; ácidos por canais iônicos receptores de transiente de potencial, ou PKD1L3-PKD2L1; e finalmente, substâncias salgadas por canais iônicos de sódio epitelial.

Em geral, cada um dos sabores básicos de paladar possui um significado distinto para os seres humanos. Acidez é um sabor produzido por alguns ácidos e pode ser considerado um sinal de alimentos decompostos e impróprios para consumo humano. A salinidade, causada por materiais iônicos incluindo cloreto de sódio (NaCl) e de potássio (KCl), indica um bom equilíbrio eletrolítico nos alimentos. Doçura, evocada principalmente por açúcares, é um indicador de fontes de nutrientes. O amargor, por sua vez, produzida por alcaloides, pode ser considerada como um sinal de um material venenoso e ajuda a impedir que os seres humanos ingiram tais materiais. Umami, evocado principalmente por aminoácidos, também mostra que abundantes nutrientes estão presentes. Por estas razões, os limiares do paladar nos seres humanos diferem com a tipo do sabor e diminuem na seguinte ordem: *salinidade* > *umami* > *azedo* > *amargor* (KOBAYASHI; IKEZAKI, 2013).

Mecanismos para avaliação de paladar encontram aplicações em diversos campos, como indústria de alimentos, bebidas e farmacêuticas. Tradicionalmente, são utilizados dois métodos para avaliar o sabor dos produtos: testes sensoriais e análises químicas.

Os testes sensoriais são realizados por degustadores que avaliam os produtos pelo sabor. No entanto, os resultados obtidos a partir de testes sensoriais são suscetíveis a fatores físicos e psicológicos, bem como às preferências individuais, muitas vezes subjetivas.

Em contraste, as análises químicas, como a cromatografia líquida de alta eficiência (*High-Performance Liquid Chromatography* (HPLC)), indicam com precisão as concentrações de

substâncias químicas em uma amostra. Infelizmente, uma avaliação de sabor com base em dados obtidos a partir de análise química é complexa devido ao grande número de substâncias presentes e do tempo requerido para a obtenção das concentrações individuais, além disso, interações entre as substâncias podem gerar efeitos sinérgicos ou de supressão no paladar (KOBAYASHI; IKEZAKI, 2013). Como exemplo, o acréscimo de substância **umami** (glutamato monossódico, por exemplo) aumenta a sensação sensorial de sabor dos alimentos criando um efeito sinérgico que mascara a concentração real dos sabores.

Algumas abordagens têm sido propostas para detectar substâncias gustativas imitando o mecanismo dos organismos vivos; uma delas é a língua eletrônica. Uma língua eletrônica é um sistema que processa dados, usando análise multivariada ou redes neurais artificiais, obtidas por várias classes de sensores, como por exemplo, eletrodo seletivo de íons (*Ion Selective Electrode* (ISE)). Assim, é necessário um processo de aprendizagem utilizando **analitos** com sabor conhecido antes de se avaliar as amostras.

Uma crítica levantada por Kobayashi e Ikezaki (2013) em relação a abordagem de eletrodo seletivo de íons é que o treinamento e avaliação de resultados exige experiência e conhecimento considerável dos operadores da ferramenta. Como contrapartida, ele propõe a utilização de sensores de membrana artificial à base de lípidos que disponibiliza uma detecção de sabores de forma semelhante a de seres humanos. Essa abordagem simula o comportamento dos sensores presentes na língua utilizando elementos lipídio/polímero que medem diferença de potencial nas amostras. Como exemplo de medidor que utiliza essa abordagem temos o equipamento TS-5000Z da *Intelligent Sensor Technology, Inc.* (INSENT, 2017) que oferece até oito sensores, cada um voltado para um sabor específico (Figura 3b). Segundo o fabricante, “o modelo permite a quantificação do sabor e a classificação das amostras medidas com o uso de reconhecimento de padrões”. Em sua versão atual, o sistema permite uma avaliação objetiva do sabor de alimentos e de bebidas como aminoácidos, água mineral, leite, molho de soja, vinho, saquê, café e pasta de soja.

O trabalho de Lanzani *et al.* (2014) também utiliza conceitos orgânicos na elaboração de sensores, contudo, ao invés de emular sistemas orgânicos, o objetivo é viabilizar uma interação direta com sistemas vivos utilizando bioeletrônica através de transdutores implementados através de semicondutores orgânicos. Essa classe de dispositivo apresenta biocompatibilidade e adaptabilidade superiores na interação com tecidos vivos.

As informações obtidas sobre o sentido paladar modelado em sistemas robóticos viabilizaram a definição do conceito *RobotTaste* (Subseção 4.2.1.2) presente na ontologia *OntPercept*.

2.2.2 Sentido do robô: olfato

O sentido do olfato processa elementos químicos com o objetivo de extrair informações do meio ambiente. Essa função biológica é essencial para a sobrevivência e reprodução de

animais, como exemplo, a mariposa macho utiliza o olfato para localizar a fêmea seguindo sua trilha de feromônio alternando entre voo direto e senoidal conforme pode ser observado na [Figura 4](#). Nos seres humanos o processamento do olfato está presente no sistema límbico que é também responsável pelo nosso comportamento, humor e memória. Como exemplo dessa conexão, é comum a recordação de eventos da infância a partir do reconhecimento de odores identificados nos primeiros anos de vida.

Figura 4 – Voo ondulado e direto de mariposa seguindo pluma de feromônio

Fonte: Adaptada de [López et al. \(2011, p. 454\)](#).

Em robótica, os estímulos baseados em [odorantes](#) são utilizados como subsídio para a navegação em aplicações de monitoração ambiental, detecção vazamentos de substâncias químicas, inspeção de aterros e operações de busca e salvamento ([MARJOVI; MARQUES, 2013](#)), ([ISHIDA; WADA; MATSUKURA, 2012](#)).

O processo de identificação da fonte do odor pelos animais é dividido em três fases: localizar vestígios químicos de interesse, percorrer a nuvem de odor e alcançar a fonte. O foco da pesquisa na área envolve o modelamento matemático da nuvem de odor, algoritmos e sensores para identificação do odorante, definir estratégia para reconhecimento do odor e algoritmo para seguir a nuvem e identificar a(s) fonte(s) de odor.

Considerando o sistema olfativo, os animais usam o olfato para diversas tarefas como inspeção, reconhecimento, acasalamento e caça. Ao ser utilizada em robôs, a percepção do ambiente através do olfato (identificação das moléculas de odorantes e sua concentração) permite, por exemplo, que ele siga trilhas de odor até atingir uma zona saturada viabilizando aplicações importantes como localizar vazamentos de gases tóxicos ou a origem de um incêndio ([VILLARREAL; OLAGUE; GORDILLO, 2016](#)).

O movimento de um organismo em resposta a um estímulo é conhecido como [taxis](#). Quando consideramos o sentido do olfato o [taxis](#) pode ser classificado conforme apresentado no [Quadro 4](#).

Considerando a quimiotaxia, existem duas estratégias para detectar a direção dos sinais químicos: *klinotaxis* e osmotropotaxia. Na *klinotaxis* o organismo possui sensores de odorantes

Quadro 4 – Classificação das técnicas de localização de odor - baseado em *taxis*.

<i>Taxis</i>	Descrição
quimiotaxia	Estímulo derivado do gradiente de concentração de substâncias odorantes presentes no ambiente. Os sensores para captura de odorantes voláteis variam dos muito simples como do tipo metal óxido que podem seletivamente detectar gases como hidrogênio, vapor de água, metano e monóxido de carbono até aos mais complexos como espectômetro de massa e cromatógrafos que podem analisar complexas composições químicas.
anemotaxia	Estímulo pela direção do vento com a utilização de anemômetros.
fluxotaxia	Uma técnica utilizada em robótica para localizar a fonte química com o uso dos princípios de dinâmica dos fluidos (ZARZHITSKY et al., 2005).

Fonte: Elaborada pelo autor.

não pareados. A detecção do estímulo é realizada por amostragem sequencial de um gradiente espacial. Um exemplo é observado em larvas de *Drosophila*, que movem suas cabeças lateralmente e usam seus órgãos olfatórios (presentes na região dorsal) para amostrar espacialmente o gradiente. Na estratégia da osmotropotaxia, encontrada por exemplo em moscas adultas, a amostragem do gradiente de concentração é simultânea utilizando órgãos olfatórios separados espacialmente ([ANDO; KANZAKI, 2015](#)).

A distinção entre ambientes abertos e fechados representa um importante fator na propagação da nuvem de odor. Quando não existe fluxo de ar, a propagação química ocorre por difusão de forma radial. Quando o fluxo de ar está presente, a propagação é realizada por [advecção](#) de forma laminar. Naturalmente, a utilização da anemotaxia é necessária em ambiente abertos, uma vez que o componente vetorial da advecção é muito maior do que o da difusão, por outro lado, quando em ambientes fechados o fluxo de ar é quase imperceptível. Vale ressaltar que a difusão de moléculas químicas no ar ou na água é um processo extremamente lento. Por exemplo, um valor típico da constante de difusão de uma molécula pequena, como o metano, no ar é $2 \times 10^{-5} m^2/s$, com isso, a extensão de difusão típica em uma hora é de apenas 0,5m ([ISHIDA; WADA; MATSUKURA, 2012](#)).

Uma vez estabelecidas as características do processo olfativo, podemos evoluir para as estratégias de tratamento dos sinais coletados. Segundo [Villarreal, Olague e Gordillo \(2016\)](#), a utilização de redes neurais para simular os neurônios sensoriais olfativos remonta ao final de 1999. Em particular, esses autores exploraram o uso de programação genética para rastrear e localizar fontes de odor, especialmente quando outros sensores são limitados ou indisponíveis. O objetivo é projetar algoritmos, baseados apenas no olfato (quimiotaxia), que possam seguir caminhos diretos em ambientes fechados, chegando rapidamente a fonte do odor. O algoritmo considera modelos matemáticos para a propagação do odor apoiando um sistema de percepção implementado em um veículo terrestre não tripulado (*Unmanned Ground Vehicle* (UGV)) que busca solução ótima para realizar a tarefa em ambientes com correntes de ar imperceptíveis para os seres humanos. O UGV utiliza um nariz bioinspirado com capacidade de determinar a

direção da fonte utilizando um par de narinas dividido por um septo. No processo de inalação, o nariz artificial concentra as moléculas de odor perto do sistema sensorial, e na exalação o nariz [dessatura](#) os sensores. Segundo os autores, a utilização de programação genética viabiliza uma solução geral que se adéque aos diferentes tempos de dessaturação, as diferenças físicas e mecânicas e ao tempo de processamento da medição.

[Ullah e Virk \(2012\)](#) apresenta um robô simples, denominado SOMR-1, que possui 20 cm diâmetro, duas rodas motorizadas, dois rodízios, quatro sensores IR, quatro sensores de olfato posicionados a 90 graus e um sistema WIFI de comunicação com um controlador externo. Com um algoritmo muito simples, que de forma cíclica direciona o robô no sentido do sensor que detém a maior concentração, o robô é capaz de localizar uma fonte de odor distante de 1,5m em cerca de 60s.

Em um trabalho que agrega biologia e redes neurais, [Arena et al. \(2013\)](#) utiliza informações de fisiologia e bioquímica disponíveis sobre o “[corpo de cogumelo](#)” presente no cérebro de insetos para desenvolver uma arquitetura neural olfativa. O modelo implementado é uma rede neural multi-camadas representando em escala o “corpo de cogumelo” mapeando desde a captura de odores pelos lóbulos antenais até seu processamento pela região denominada *calyx*, além de considerar capacidade de aprendizado e memória com o uso da técnica *Spike-Timing Dependent Plasticity –STDP*.

[Ando e Kanzaki \(2015\)](#) realizam pesquisa biológica envolvendo o estudo da funcionalidade do sistema olfativo em insetos. O desenvolvimento envolve a “simbiose” entre um robô móvel e uma mariposa macho do bicho da seda. Na implementação, a mariposa caminha sobre um globo sintético instalado sobre o robô cujo movimento controla os motores das rodas ([Figura 5](#)). O objetivo é estudar as características funcionais do sistema olfativo da mariposa macho quando excitado pelo feromônio da fêmea. O trabalho demonstra que a direção selecionada envolve não somente estímulos olfativos, mas também estímulos visuais e realimentação do movimento realizado. Além disso, a mariposa utiliza duas estratégias para seguir a nuvem de feromônio: osmotropotaxia utilizando órgãos olfatórios bilaterais e zigzague quando o odor não é percebido.

Considerando a fase de classificação de odores, o artigo dos pesquisadores [López-Serrano et al. \(2012\)](#) defende a tese que estrutura temporal do sinal induzido pelo odor também contribui para sua identificação. Dessa forma, as estratégias clássicas de classificação de odor, como *k-nearest neighbors* e análise do discriminante linear (*Linear Discriminant Analysis (LDA)*) devem também considerar o momento da detecção do odor.

Os trabalhos anteriores focam principalmente em seguir uma nuvem de odor identificada. O passo anterior, que envolve a localização inicial da nuvem de odor, representa o foco principal da pesquisa de [Marjovi e Marques \(2013\)](#) com o uso de enxame de robôs. Após representar matematicamente as concentrações e dispersão dos odores considerando velocidade e direção do vento, o autor conclui que a topologia ótima de posicionamento é em linha, em direção

Figura 5 – Robô móvel controlado por mariposa e sistema de dutos para sucção de odor

Fonte: [Ando e Kanzaki \(2015\)](#).

transversal ao vento, com igual distância entre cada par de sensores vizinhos. Além disso, a distância entre pares vizinhos depende principalmente da velocidade do vento, ou seja, quanto maior a velocidade do vento mais próximos os sensores devem ser posicionados. Para garantir essa formação é apresentado um conjunto de forças de controle de atração e repulsão polarizadas pelo vento de modo que o comportamento emergente de trajetória dos robôs convirja para a formação ótima.

Apesar dos avanços promissores, [Ishida, Wada e Matsukura \(2012\)](#) apresentam alguns desafios presentes na detecção de odores, como por exemplo:

- as nuvens de odores se movimentam e difundem através de um meio (ar, água) no espaço 3D, assim a migração de robôs terrestres para aéreos e submersos é uma questão que deve ser modelada;
- uso de cenários complexos para validação como por exemplo: o fluxo de ar também é afetado por variações de temperatura, obstáculos e janelas onde incide a luz solar em espaços fechados e
- desenvolvimento de sensores rápidos com recuperação eficiente. Como exemplo, nos animais o tempo de resposta é da ordem de 0,1s, por outro lado, os típicos sensores de metal óxido oferecem tempo de resposta de 5s com dessaturação de 1 minuto.

Além disso, os autores também acrescentam que os algoritmos baseados em processos biológicos mimetizam o comportamento dos animais, sem atingir o mesmo nível de desempenho; isto é, a fonte de odor não é determinada com precisão ou demanda muito tempo para ser alcançada.

Finalmente, um importante aspecto associado ao senso de olfato é determinar a forma apropriada para representar os odores, note que, a representação completa dos odores demandaria uma classe para cada monocelula presente na natureza. Naturalmente, quando consideramos o

processamento desse volume de informação, essa abordagem é inviável. Como contrapartida, [Dravnieks \(1985\)](#) elaborou um atlas com odores monomoleculares (como exemplo: Acetofenona, Frutona, Ácido 4-pentenóico) estabelecendo um perfil com dimensão 146, cada um correspondendo a um rótulo como: floral, terra, pútrida, alho, queimado entre outros.

Em trabalho mais recente, [Castro, Ramanathan e Chennubhotla \(2013\)](#) utilizam como base o atlas de Dravnieks para representar o perfil de odor em um espaço multidimensional com 10 grupos ($W1..W10$) através da técnica de redução da dimensionalidade original de 146. Esses grupos são rotulados como: *Fragrant, Wood, Fruity, Decayed, Chemical, Minty, Sweet, Popcorn, Pungent, Lemon*. O autor observa que esses rótulos são apenas para fins de concisão, pois cada eixo é, na verdade, “um meta-descritor que consiste em uma combinação linear de descritores mais elementares oriundos do atlas de Dravnieks”. Essa classificação de odores foi adotada nesta pesquisa conforme detalhado no [Capítulo 4](#) na definição do conceito *RobotSmell* (Subseção [4.2.1.3](#)) presente na ontologia *OntPercept* desenvolvida nesta tese.

2.2.3 Sentido do robô: audição

Os mecanismos que regem o sentido da audição são similares aos presentes no olfato:

- dependentes do meio (ar ou água) para se difundir;
- nos mamíferos está presente aos pares (direito e esquerdo), viabilizando determinar a fonte do sinal;
- técnicas distintas de análise para ambiente interno e externo.

As pesquisas envolvendo audição levam uma pequena vantagem sobre o olfato devido aos microfones existirem desde o século dezenove resultando assim, em uma tecnologia muito mais amadurecida. Além disso, os microfones não necessitam de dessaturação prévia antes de novas aquisições. Mesmo assim, muitas das técnicas presentes no olfato são também utilizadas na audição.

A utilização do sentido de audição nos agentes robóticos ganhou impulso com a necessidade de interação com humanos através da compreensão da fala. O uso da linguagem em sistemas robóticos pode ser observado em trabalhos de medicina ([NEUSTEIN, 2014](#)), reabilitação ([ROSATI et al., 2012](#)), ([ARDIANSYAH, 2016](#)) e robôs de serviço ([GUAN; HE; ZHANG, 2014](#); [WANG; YANG; SUN, 2016](#)).

O foco desta revisão é sobre mecanismos de percepção e reconhecimento de sons pelo agente robótico. Esse direcionamento exclui os casos de captura para posterior identificação por humanos, como por exemplo: transformar diferença de potencial presentes em músculos (batimento cardíaco) em sinais sonoros durante cirurgias, sonares para pessoas com deficiência visual, utilização de informação auditiva como complemento na definição da trajetória de instrumentos cirúrgicos como agulha ou cateter.

Uma das maiores dificuldades presentes na análise da fala é a presença de “ruídos” no sinal, como por exemplo, alarmes, veículos, equipamentos e animais (MARTINSON; BROCK, 2013). Vale ressaltar que os ruídos não devem ser simplesmente rejeitados, todas as fontes de som representam informações preciosas na interação do agente robótico com o ambiente. A pesquisa descrita nesta tese viabiliza a análise dos “ruídos” através da representação de todas as classes de sinais audíveis em um mesmo quadro de captura.

A percepção humana adquire mais rapidamente informação de estímulos sonoros do que estímulos de toque ou visuais. Essa característica determina que a captura e processamento de sinais sonoros seja mais rápida em comparação com outros estímulos (VIDAL; ZARRICUETA; CHEEIN, 2015). Ao ouvir um ruído, o ser humano reage imediatamente, rotacionando a cabeça em busca de informação visual para determinar sua origem. O colículo superior é a estrutura presente no cérebro humano responsável pela fusão dos dados audição e visão viabilizando a reação apropriada ao estímulo sonoro (RAVULAKOLLU; ERWIN; BURN, 2011).

A mesma estratégia é adotada em agentes robóticos que refinam seu comportamento utilizando informações auditivas do meio ambiente. O trabalho de Park *et al.* (2016) utiliza múltiplas modalidades sensoriais para detectar anomalias na execução de tarefas pelo robô. Em particular, o pesquisador explora sinais sonoros inesperados (grito de um humano, som de contato com objetos e alarmes, por exemplo) para identificar anomalias na execução através de processamento realizado via *Hidden Markov Model* (HMM) pela comparação de observações multimodais de execuções não-anômalas.

Em abordagem semelhante, Akolkar *et al.* (2015) utiliza estímulos sonoros para detectar *saliências* áudio-visuais. A arquitetura proposta utiliza sinais sonoros como gatilho para captura instantânea de imagens de colisões, muitas vezes despercebidas devido a latência intrínseca presente na captura de imagem como uma sequência periódica de quadros.

O microfone é o sensor mais utilizado para captura de sinais audíveis, contudo não é o único. Além de microfones, técnicas adicionais podem ser utilizadas para detecção das ondas sonoras, como por exemplo: sensores de fluxo em polímero que imitam a morfologia dos feixes ciliares mecanosensoriais presentes em células ciliadas biológicas (ASADNIA *et al.*, 2016) ou sonares (HA *et al.*, 2015).

Quando consideramos robótica social, uma das utilizações mais frequentes de sinais sonoros envolve a comunicação com humanos através da fala. Uma pesquisa associada a essa questão envolve identificar a posição da fonte em um ambiente com diversas pessoas utilizando um modelo híbrido “probabilístico/determinístico” para associar os sons às pessoas presentes no ambiente (ALAMEDA-PINEDA; HORAUD, 2015). Outro trabalho controla um robô com comandos sonoros capturados por dois microfones (MAGASSOUBA; BERTIN; CHAUMETTE, 2015), e finalmente técnicas de reconhecimento de voz podem ser utilizadas em robôs de resgate para identificar as vítimas (SANTHOSH, 2014).

As informações obtidas sobre o sentido audição modelado em sistemas robóticos viabilizaram a definição do conceito *RobotHear* (Subseção 4.2.1.4).

2.2.4 Sentido do robô: tato

Em seres humanos, o senso de tato é associado com a percepção mediada pelo contato entre a pele e um objeto. Diversos são os estímulos associados ao tato: químico ou pontiagudo (sensação de dor), temperatura, umidade e mecânicos como vibração, pressão, toque e deslizamento. A Figura 6 apresenta os corpúsculos associados a cada classe de sensação mecânica presentes na pele humana.

Figura 6 – Mecanorreceptores da pele humana e estímulos associados

Fonte: Lanzani *et al.* (2014).

Várias abstrações dessa percepção foram modeladas por dispositivos tecnológicos como telas sensíveis ao toque, telas flexíveis (Organic User Interfaces) (HOLMAN *et al.*, 2013), sensores tácteis robóticos, etc. Os sensores utilizados nessas abstrações utilizam uma ou mais das seguintes tecnologias: piezoresistivos, capacitivo, piezoelétrico, óticos (MARZBAN; PACKIRISAMY; DARGAHI, 2015), barométricos, acústico, *Quantum Tunneling Composite* (QTC), dentre outros. Uma visão detalhada sobre transdutores para capturar percepção tátil, seus méritos e limitações, é apresentada por Kappassov, Corrales e Perdereau (2015).

A instalação dos sensores nos sistemas robóticos é dependente da utilização desejada, como exemplo, em atividades de “agarrar” envolvendo manipuladores, os sensores são preferencialmente instalados nas mãos, em geral nas pontas dos dedos ou em regiões próximas como, palma da mão e antebraço. Poucos trabalhos posicionam os sensores na parte inferior do corpo, como pernas ou sola do pé. Como contraponto, Sullivan *et al.* (2012), desenvolveu sensores para atuarem como vibrissas (bigodes) presentes em mamíferos como, gatos e ratos. Esses sensores viabilizam a classificação precisa da textura de três superfícies diferentes com o uso de um classificador Naive Bayes.

Cada sentido oferece desafios distintos envolvendo a captura e o processamento das informações do ambiente. Quando consideramos o sentido do tato, dois desafios merecem destaque: multiplicidade de atributos e posicionamento dos sensores. A multiplicidade de atributos envolve a necessidade de utilização de sensores distintos para obter temperatura, pressão, escorregamento e textura. O posicionamento de sensores envolve a distribuição e o volume de sensores na área de contato desejada. Teoricamente, o tato deveria estar presente em cada milímetro da “pele” do agente robótico, demandando uma complexa rede de suprimento de energia, transmissão e processamento de dados (YOUSSEFI *et al.*, 2015). Considerando as dificuldades tecnológicas e financeiras dessa solução, a instalação dos sensores táteis acaba sendo moldada pela aplicação. Em aplicações industriais e médicas, os sensores são posicionados nas extremidades das garras de manipulação, nos humanoides na ponta dos dedos e, finalmente, em robôs passíveis de movimentação, nas rodas, esteiras ou na planta dos pés.

O primeiro passo na captura de informações de tato envolve a definição do sensor a ser utilizado. Sadun, Jalani e Sukor (2016) apresenta um conjunto de sensores de força para utilização em garras ou mão robótica. Esses sensores são baseados em sistemas microeletromecânicos (MEMS) utilizando polímeros eletroativos (*Electroactive Polymers* (EAP)). Os polímeros eletroativos são capazes de se adaptar a superfícies de diferentes formas sendo atraentes em aplicações que envolvam sensores vestíveis e interfaces com tecidos moles (WANG *et al.*, 2016). Uma alternativa aos polímeros eletroativos envolve a utilização de nanotubos de carbono como sensores de força. Segundo Hyo Seung Han *et al.* (2016), os nanotubos representam uma alternativa com menor custo e desempenho superior, infelizmente, ainda em estágio inicial de desenvolvimento.

Uma vez que as informações táteis sejam capturadas, inicia-se a fase de processamento. Em geral mais de uma classe de informação é obtida demandando técnicas de fusão e modelagem para aumentar a confiabilidade dos dados. Gil *et al.* (2016) pesquisam a manipulação de objetos elásticos combinando dados táteis e *Red Blue Green and Depth information* (RGB-D) para determinar o movimento dos dedos quando a geometria do objeto muda como resultado da deformação causada pela pressão.

As aplicações envolvendo o sentido do tato englobam, dentre outras as áreas: industrial (SPIERS *et al.*, 2016), médica (CUPPONE *et al.*, 2016), interação com humanos (JAMALI *et al.*, 2015) e identificação do terreno em robótica móvel (HOFFMANN; ŠTĚPÁNOVÁ; REINSTEIN, 2014). Em um exemplo concreto, Winiarski, Banachowicz e Seredyński (2015) utilizam o robô KUKA LWR4 para abrir uma porta com feedback visual utilizado para planejar a aproximação e feedback tátil para detectar o contato das garras com a maçaneta (Figura 7).

A relevância da pesquisa háptica pode ser comprovada através do projeto europeu ROBOSKIN (BILLARD *et al.*, 2013) que persistiu de 2009 até 2012. O ROBOSKIN teve como objetivo desenvolver e demonstrar novas capacidades robóticas com o uso de feedback tátil fornecido por uma “pele” cobrindo grandes áreas do corpo do robô. Os recursos foram

Figura 7 – Uso de feedback visual e tátil para movimentar maçaneta de porta

Fonte: Winiarski, Banachowicz e Seredyński (2015).

direcionados tanto no desenvolvimento de sensores quanto na criação de modelos para representar e integrar dados táteis com as arquiteturas cognitivas existentes. A Figura 8 apresenta um resultado do ROBOSKIN que disponibilizou procedimentos específicos para cobrir robôs de diferentes formas com malhas táteis.

Figura 8 – Robôs cobertos por peles artificiais: iCub, Kaspar e NAO

Fonte: Billard *et al.* (2013).

Mesmo apresentando múltiplas linhas de pesquisa, algumas embrionárias, a háptica obteve maturidade suficiente para ser explorada como produto. O “BioTac Toccare” da SynTouch (SYNTOUCH, 2018) permite testar características táteis de materiais utilizados em linhas de produção. O “Grip System” da Tekscan (TEKSCAN, 2018) e o “Finger tps” da PPS Inc. (PPS-INC., 2018) realizam medições e avaliam vetores de força em operações de agarrar objetos com sensores adaptáveis aos dedos do usuário. O BarrettHand da Barrett Technology (BARRETT, 2018) é um manipulador com três garras capaz de segurar objetos de diferentes tamanhos, formas e orientações.

As informações obtidas sobre o sentido tato modelado em sistemas robóticos viabilizaram a definição do conceito *RobotTouch* (Subseção 4.2.1.5).

2.2.5 Sentido do robô: visão

Desde sua criação no final do século XIX a câmera fotográfica têm “congelado” informações visuais permitindo sua preservação e posterior análise. Mesmo considerando toda essa herança histórica, o processamento de imagens ainda representa um desafio tanto do ponto de vista computacional quanto de modelamento. Precisão (resolução/amostragem e quantização), oclusão, influências do ambiente, cenas em movimento e tempo de resposta representam somente algumas das questões em aberto.

Este cenário constitui um catalisador na busca por novas técnicas de captura e processamento [Bayro-Corrochano et al. \(2016\)](#), por exemplo, critica a utilização de redes neurais associadas ao cálculo vetorial e álgebra linear como parte da solução para processamento de imagens. Como contrapartida, considerando que a representação do mundo por animais é realizada em termos geométricos, propõe a utilização de álgebra geométrica associada com função atômica quaterniônica (*Quaternionic Atomic Function* (QAF)) e redes geométricas RBF (*geometric RBF networks* (GRBFN)) para a modelagem do problema.

O processo de análise visual inicia com a recepção de informações através de sensores. Câmeras ópticas tradicionais e câmaras RGB-D representam algumas das tecnologias utilizadas para captura de imagens. Segundo [Lachat et al. \(2015\)](#), na última década as câmeras RGB-D têm sido largamente utilizadas em aplicações robóticas. Em 2010, o advento da câmera de baixo custo, Kinect v1 da Microsoft, promoveu o uso da tecnologia viabilizando aquisição de dados 3D. Em seu estudo, o autor recomenda tempo de aquecimento de 30 minutos para Kinect v2, bem como calibrações para aferir a profundidade apresentada pelo sensor.

Para ambientes sem luz e turvos, [Bai et al. \(2015\)](#) apresentam um sensor baseado em campos elétricos. A estratégia é semelhante à utilizada por alguns peixes que geram um campo elétrico para detecção de objetos próximos. O algoritmo apresentado oferece precisão menor que 1cm para posição e 5 graus para orientação, além de estimar o tamanho do objeto com precisão de 10%.

Uma interessante tecnologia, denominada *Dynamic Vision Sensor* (DVS), é explorada por [Mueggler, Huber e Scaramuzza \(2014\)](#) para captura, em alta velocidade, de imagens produzidas por um quadricóptero em manobras de desvio de obstáculos. Ao contrário de uma câmera CMOS padrão, um DVS não produz imagens completas a uma taxa de quadros fixa. Na verdade, o DVS transmite somente mudanças de brilho dos *pixels* no instante em que ocorrem, com precisão de microssegundos, criando um pipeline de percepção cuja latência é insignificante em comparação à dinâmica do robô.

Independente da tecnologia utilizada, após a captura, as informações necessitam ser processadas para viabilizar seu uso. Em geral as informações são compostas por matriz de *pixels* e, eventualmente, informação de distância. Vale ressaltar que as imagens contêm ruídos gerados por poeira, chuva ou neve quando atuando na superfície; iluminação deficiente em

minas, túneis ou a noite; além de turbidez quando embaixo d'água. Esse complexo conjunto de informações demanda um processamento rápido realizado por hardware e/ou algoritmos que procuram identificar pontos de interesse específicos no ambiente. Como exemplo, [Dias e Osorio \(2015\)](#) apresentam uma estrutura de hardware baseada em *Field Programmable Gate Array* (FPGA) que aumenta o desempenho da navegação visual de veículos autônomos processando taxas de quadros de 43FPS (*frames-per-second*). Na verdade, a quebra da barreira de 30FPS associada ao olho humano, para atender aplicações que necessitam de centenas de Hertz é necessária em diversas áreas como: sensoramento da dinâmica de vibrações, inspeção visual tridimensional e aplicações biomédicas ([GU; ISHII, 2016](#)).

Complementando o processamento, métodos de estabilização de vídeo são necessários para minimizar vibrações inerentes as instalações das câmaras. Em geral, os métodos de estabilização são realizados off-line com ferramenta de pós-processamento de vídeos pré-gravados. Por outro lado, [Liu et al. \(2016\)](#) propõe uma técnica para estabilização de vídeo online com latência de um quadro usando o modelo de movimento *MeshFlow*. Outra estratégia envolve reconstrução de imagem para recuperar a imagem original a partir de imagens produzidas em um ambiente ruidoso. [Khilar, Chitrakala e SelvamParvathy \(2013\)](#) apresentam um estudo detalhado de técnicas utilizadas para reconstrução de imagens 3D como: técnica de visualização, método de triangulação, sistema de captura cinética múltipla e manipulação de informação a partir de retalhos.

Como forma de aumentar a confiabilidade dos resultados, informações de outras fontes (câmeras adicionais, GPS, Lidar, Sonar e etc) são utilizadas em conjunto em técnicas de fusão de dados, como exemplo, os algoritmos para *Simultaneous Localization and Mapping* (SLAM). [Brunetto et al. \(2015\)](#) propõem uma arcabouço para robôs móveis que funde estimativas de posicionamento obtidas a partir de medidas visuais e inerciais, com o objetivo de ampliar cenários passíveis de processamento por SLAM visual móvel. O arcabouço implementa uma matriz de filtros de Kalman onde a seleção cuidadosa das variáveis de estado e o pré-processamento das medições do sensor inercial resultam em um processo de fusão de dados que, segundo os autores, se apresenta simples e eficaz.

Com o objetivo de estimar a capacidade de travessia de terrenos acidentados, [Sock et al. \(2016\)](#) apresentam uma abordagem de fusão que elabora um mapa de grade probabilístico em 2D utilizando informações de um sensor 3D-LIDAR e uma câmera. O algoritmo constrói dois mapas probabilísticos, um para cada sensor, que posteriormente são fundidos com o uso da regra de Bayes para melhorar o desempenho da detecção.

Na prática, as estratégias de processamento exploradas nos parágrafos anteriores são utilizadas em aplicações que envolvem: robótica móvel, indústria, serviço, medicina e agricultura. As competições DARPA foram catalisadoras de resultados importantes em veículos terrestres autônomos. Exemplo de pesquisa neste contexto envolve o uso de odometria visual como um método alternativo para determinar o deslocamento do veículo ([ESCALERA et al., 2016](#))

ou navegação “fora de estrada” com o apoio da fusão de visão estéreo e laser para detecção de obstáculos (HUSSEIN *et al.*, 2016). Esses resultados, inicialmente aplicados em veículos, acabaram por alcançar outras áreas como humanoides (BELTER *et al.*, 2016) ou cadeiras de rodas (CHAVEZ-ROMERO *et al.*, 2016).

Na indústria, informações visuais são utilizadas para acompanhar trajetória de manipulador (GAO *et al.*, 2016), controle de solda (HE *et al.*, 2016) e captura de movimentos de humanos para posterior reprogramação de robôs industriais (FERREIRA *et al.*, 2016). As aplicações de serviço abrangem desde controle de focos de incêndio (MCNEIL; LATTIMER, 2016), resgate em minas (SHANG *et al.*, 2016) e inspeção de linhas de alta voltagem (ZHU; WANG; XU, 2016).

Em aplicações médicas a visão oferece apoio em interações emocionais (BEVILL *et al.*, 2016), reabilitação (AIRÒ *et al.*, 2016) e em cirurgias (Moradi Dalvand; NAHAVANDI; HOWE, 2016)(LADDI *et al.*, 2016). Finalmente, em agricultura, sistemas robóticos utilizam a visão para controle de criação animal (VÁZQUEZ-ARELLANO *et al.*, 2016), colheita (FONT *et al.*, 2014) e navegação (BALL *et al.*, 2016). A Figura 9 apresenta o veículo utilizado por Ball *et al.* (2016) para oferecer suporte a um implemento de pulverização de 5 m de largura com capacidade de 200 l. O controle desse veículo é realizada por kit da RoPro Design (ROPRO, 2018).

Figura 9 – Uso de visão em agricultura

Fonte: Ball *et al.* (2016).

As informações obtidas sobre o sentido visão modelado em sistemas robóticos viabilizaram a definição do conceito *RobotVision* (Subseção 4.2.1.6) presente na ontologia *OntPercept*.

2.3 Considerações finais

Quando focamos em ontologias, a pesquisa realizada ressalta sua diversidade em geral e, em particular, no domínio da robótica. O grande catalisador dessa diversidade foi a necessidade de modelar diferentes bases de informação disponíveis na Web. A tecnologia das linguagens RDF/OWL e da [Web Semântica](#) facilitaram essa expansão.

Os trabalhos associados a captura de informações sensoriais apresentam individualmente resultados surpreendentes e de altíssima qualidade. Contudo, a distribuição das pesquisas não é uniforme, sendo notória a diferença entre olfato ou paladar e os outros sentidos. Esta diferença está associada ao fato da visão, audição e tato serem percebidos apenas de acordo com as propriedades físicas do ambiente, assim, os sensores para detectá-los estão em desenvolvimento há muitos anos. Em contraste, sabor e odor são sentidos químicos identificados por quimiorreceptores com mecanismos complexos e, portanto, considerável dificuldade existe em seu desenvolvimento.

Neste contexto, quando consideramos sistemas robóticos, a utilização efetiva desses sensores demanda o equacionamento de questões ainda em aberto, tais como, processamento e fusão de informações, consumo de energia, distribuição dos sensores, tempo de dessaturação dentre outros. A dificuldade e complexidade das pesquisas nesta área representam uma motivação adicional para esta tese. Acreditamos que, a definição da ontologia *OntPercept* possa contribuir para a maturação dos mecanismos associados à ciência cognitiva enquanto as técnicas associadas ao processamento dos sentidos estejam em maturação.

HRI E COGNIÇÃO

No [Capítulo 2](#) foi apresentada a revisão bibliográfica para ontologia robótica e implementação da percepção sensorial em agentes robóticos. Estas áreas são essenciais na pesquisa desenvolvida nesta tese e portanto mereceram uma análise mais detalhada.

Quando consideramos o ambiente para validação e verificação da ontologia proposta, duas áreas adicionais necessitam ser exploradas: HRI e Cognição. Ambas são apresentadas neste capítulo, destacando-se o estado atual da pesquisa em cada uma das áreas.

- **HRI:** O nível de informação necessária na comunicação entre o ser humano e o robô depende da autonomia exercida pelo robô em um dado momento. Por um lado temos o *Automated Robot* que necessita de pouca interação pois possui um comportamento de autômato, sendo em geral utilizado em automação industrial. No outro extremo, com uma complexidade superior, temos o *Fully Autonomous Robot* que é completamente autônomo atuando como robô de serviço¹. Esse é o tema explorado na [Seção 3.1](#).
- **Cognição:** Assumindo que HRI represente um produto de software, podemos realizar sua avaliação utilizando as características de modelo de qualidade em uso presente na norma ISO/IEC 25010 ([ISO/IEC:25010, 2011](#)). Uma dessas características é a “satisfação”, que avalia o grau de satisfação das necessidades do usuário no uso do sistema em um contexto específico. A satisfação é avaliada por suas sub características: utilidade², confiança, prazer e conforto na utilização. Quando consideramos o nível de proximidade exigido pela interação entre humanos e robôs podemos concluir que se trata do mesmo nível presente nas interações entre humanos. Assim, para atender a característica “satisfação” é necessário que o robô interaja com humanos em um nível próximo às interações humanas. Estes aspectos são tratados pela ciência cognitiva que é explorada na [Seção 3.2](#).

¹ Adotada definição de níveis de autonomia presentes na norma [IEEE \(2015\)](#).

² Representa a satisfação do usuário com os resultados de uso.

3.1 *Human-robot interaction*

HRI é uma área relativamente recente de pesquisa que ainda busca definir seu escopo de atuação e maturação. Como forma de ressaltar essa diversidade apresentamos três definições:

- [Goodrich e Schultz \(2008\)](#): “HRI é um campo de estudo dedicado a compreender, projetar e avaliar sistemas robóticos para uso por ou com humanos”.
- [Feil-Seifer e Matarić \(2009\)](#): “HRI é o estudo interdisciplinar da dinâmica da interação entre humanos e robôs”.
- Segundo relatório da *World Technology Evaluation Center* (WTEC) sobre HRI ([VELOSO, 2012](#)), “HRI é uma nova área de pesquisa que visa estender a funcionalidade e usabilidade de robôs fazendo as interações com os seres humanos mais naturais e compatíveis com as expectativas e capacidades humanas”.

A dificuldade de estabelecer uma definição unificada surge da multiplicidade de campos que a área agrega: engenharias (elétrica, mecânica, industrial e design), ciência da computação (interação humano-computador, inteligência artificial, robótica, compreensão de linguagem natural e visão por computador), ciências sociais (psicologia, ciência cognitiva, antropologia e fatores humanos) e humanidades (ética e filosofia) ([GOODRICH; SCHULTZ, 2008](#)). Esta multiplicidade permite também definir HRI como uma disciplina que congrega resultados de outras áreas aplicados conjuntamente na comunicação entre humanos e agentes robóticos.

HRI agrega pesquisadores oriundos das engenharias, computação, ciências sociais e humanidades. Essa herança de formação acaba moldando uma visão particular de uma das áreas principalmente quando a equipe de pesquisa não é multidisciplinar. Como exemplo, pesquisadores com formação original em computação tendem a ressaltar a importância da Interação Humano-Computador (IHC) como principal contribuinte para HRI³. Por outro lado, a experiência alcançada com HRI também é importante para os avanços da IHC. Assim, longe de ser um relacionamento hierárquico, as áreas se interligam e se sobrepõem. Essa perspectiva pode ser extrapolada para as outras áreas de pesquisa envolvidas, como ciências sociais ou humanidades.

A robótica social representa uma importante subárea da HRI com foco no estudo das interações sociais com humanos ou outros robôs. Um aspecto importante nesta subárea consiste em interpretar o ambiente, perceber e identificar o comportamento humano. A computação afetiva e a proxêmica representam linhas de pesquisa presentes na robótica social. A computação afetiva busca reconhecer e expressar emoções através da análise de expressões faciais, voz e gestos. A proxêmica, por sua vez, busca compreender o complexo protocolo adotado por

³ Note que, embora sob alguns aspectos a interação entre robôs e humanos tenha interesse próprio (por exemplo, cognição e *haptics*) a experiência alcançada em décadas de IHC certamente é essencial para HRI.

humanos para estabelecer distâncias físicas em seu convívio social. A robótica social representa a base para duas importantes subáreas presentes na IHR:

Robótica Socialmente Assistiva: representa uma área crescente de pesquisa com potenciais benefícios para os idosos, pessoas com distúrbios sociais e cognitivos, reabilitação, entre outros. A Socially Assistive Robotics (SAR) (FEIL-SEIFER; MATARIĆ, 2009) representa a interseção da robótica assistiva com a robótica social. SAR surgiu da necessidade de resolver problemas voltados para robôs assistivos que envolvam também interação social ao invés de somente aspectos físicos da interação;

Robótica Educacional: a robótica tem se mostrado uma ferramenta poderosa para o aprendizado, não apenas como um tópico de estudo pelos alunos, mas também, como um importante componente no ensino de matemática, computação e engenharia ao motivar trabalhos em equipe envolvendo inclusive participação em competições. Em alguns domínios, robôs podem também serem utilizados para instrução e catalisadores de habilidades sociais.

As próximas duas seções apresentam visões distintas de classificação da HRI considerando: autonomia, atributo da aplicação e métricas de avaliação de qualidade.

3.1.1 Classificação: autonomia

O ponto de convergência nas definições de HRI é a palavra interação. A interação entre humanos e robôs, por sua vez, é dependente da proximidade, frequência e objetivo da comunicação estabelecida. Estes fatores determinam a autonomia do robô, ou seja, a quantidade de tempo no qual a interação com o robô pode ser negligenciada pelo usuário sem comprometer os objetivos de uso. Nesta pesquisa assumimos as definições de autonomia estabelecidas pela norma IEEE 1872-2015 (IEEE, 2015):

Robô automatizado (automated robot): robô executa uma determinada tarefa atuando como um autômato. Não se adapta às mudanças do ambiente, seguindo sempre um sequenciamento de ações pré estabelecidas.

Robô controlado remotamente (remote-controlled robot): robô assume o papel de executar uma determinada tarefa na qual o operador humano atua em uma base contínua, a partir de uma localização externa ao robô, utilizando apenas observação direta. Neste modo, o robô não toma iniciativa e baseia-se somente na entrada contínua (ou quase) do operador humano. Como exemplo, temos os robôs para desarme de bombas.

Robô teleoperado (teleoperated robot): papel assumido pelo robô para executar uma dada tarefa onde um operador humano, usando realimentação sensorial, controla os atuadores ou atribui metas incrementais numa base contínua, a partir de uma localização externa ao robô.

Um robô teleoperado deve completar seu último comando após o operador interromper o envio de comandos, ainda que esse comando seja complexo ou demorado. Como exemplo, temos os robôs enviados para o planeta Marte pela agência espacial americana ([NASA, 2015](#)).

Robô semi autônomo (semi-autonomous robot): executa uma determinada tarefa onde robô e humano planejam e conduzem a tarefa, que exige vários níveis de interação humana.

Robô totalmente autônomo (fully autonomous robot): robô executa uma determinada tarefa sem intervenção humana, enquanto se adapta às condições operacionais e ambientais.

Um ponto interessante é que a atribuição de autonomia pode ser momentânea, como exemplo, num dado instante o robô opera de forma totalmente autônomo, mas ao receber um evento particular (exemplo: falha, novas condições ambientais, final da missão) pode mudar para o modo teleoperado.

3.1.2 Classificação: atributo da aplicação

O desafio da área de HRI é compreender e moldar as interações entre um ou mais humanos e um ou mais robôs. [Goodrich e Schultz \(2008\)](#) consideram que o projeto de HRI envolve cinco níveis de interação:

Nível e comportamento da autonomia: período de tempo no qual o robô pode ser negligenciado sem comprometer a atividade em execução. Este atributo remete a classificação apresentada na [Subseção 3.1.1](#).

Nível de cognição: representa a natureza da troca de informações entre o humano e o robô. Este aspecto é revelado pelas seguintes características:

- *Métricas do protocolo:* tempo da interação, carga cognitiva, volume de compreensão comum entre humano e robôs.
- *Domínios da informação:* meio de comunicação (ver, ouvir e tocar) e formato de comunicação (fala pode ser natural ou limitada; *haptic* pode ser vibração, telepresença e roupas especiais; áudio pode ser alerta ou voz e sons presentes em um espaço 3D).
- *Informação social:* inclui direção do olhar, gestos, divisão de espaço físico, imitação, sons verbalizados e expressão facial.
- *Interfaces gráficas:* incluem realidade virtual e a tradicional interface por sistema de janelas.

Estrutura do time: extrapola o conceito básico de uma pessoa e um robô. Trabalho em equipe tanto de humanos quanto de robôs são comuns, como operações de salvamento e grupo de veículos não tripulados atuando em sincronia. A estrutura do time envolve autonomia, tipo

de tarefa, volume de dados trafegados e formas de comunicação. Outro aspecto interessante é a organização do time, ou seja: “Quem tem autoridade para tomar decisões?”; “Como os conflitos são resolvidos?”; “A hierarquia é estática ou dinâmica?”.

Adaptação, aprendizado e treinamento de pessoas e do robô: o treinamento e o aprendizado dos robôs são alvos de diversas pesquisas especialmente nos contextos de interações de longo prazo. Entretanto, o treinamento dos seres humanos tem recebido menos atenção. Naturalmente, o objetivo da área HRI é gerar sistemas que dispensam treinamento. Entretanto, algumas aplicações demandam um treinamento mínimo do usuário seja através de: cursos, manuais, instruções do pesquisador ou fornecidas pelo próprio robô. Como exemplos de aplicações que demandam treinamento específico temos: militar, policial, espacial e operações de busca e resgate.

Preparo do Ambiente: dependendo da aplicação pode ser necessário o preparo do ambiente antes do início da operação com o robô. A preparação do ambiente consiste em planejar como uma tarefa será realizada quando uma nova tecnologia for introduzida, tais como, a remoção de objetos do espaço antes de usar um robô para limpeza doméstica ou a execução de tarefas de inspeção e mapeamento prévios antes do envio do robô para busca e resgate.

Cada classe de aplicação envolvendo sistemas robóticos possui um valor específico para cada um destes atributos. Vale novamente ressaltar que o valor de um dado atributo pode ser alterado no decorrer da tarefa. Além disso, operações de instalação e manutenção do robô requerem interfaces próprias que não fazem parte da interação requerida durante a fase de operação do sistema. Em [Quadro 5](#), são apresentados os domínios mais comuns de aplicações juntamente com níveis de atributos associados. Esses níveis representam somente uma atribuição empírica para exemplificar conceitos, assim, não devem ser considerados como uma referência absoluta para o tema.

3.1.3 Métricas para HRI

Sistemas para HRI estão continuamente sendo desenvolvidos e utilizados por seres humanos nas mais diversas condições e locais. Estas aplicações demandam criteriosa avaliação de qualidade e respeito à ética ([FEIL-SEIFER; MATARIĆ, 2009](#); [MURPHY; SCHRECKENGHOST, 2013](#)). Como exemplo da relevância do tema, pesquisas envolvendo crianças, deficientes e idosos representam o principal foco de interesse na robótica assistiva ([LEITE; MARTINHO; PAIVA, 2013](#)). A comprovação da qualidade é alcançada pelo uso de métricas que definem o método e a escala de medição para um dado atributo do sistema.

Na [Figura 10](#) é apresentada uma taxonomia para métricas envolvendo HRI. A taxonomia proposta por [Murphy e Schreckenghost \(2013\)](#) foi elaborada a partir de uma pesquisa realizada em 2013, que analisou 29 artigos sobre métricas para HRI. Os autores concluem que na prática,

Quadro 5 – Domínios de aplicações x atributos de HRI

Aplicação	Autonomia	Cognição	Time(qtd)	Aprendizado(nível)	Preparo
Assistiva	autônomo	Alta	H(1) R(1)	H(Baixo) R(Alto)	Médio
Busca e Resgate	controle remoto	Baixa	H(n) R(n)	H(Alto) R(Baixo)	Baixo
Educacional	semi autônomo	Alta	H(n) R(1)	H(Baixo) R(Alto)	Alto
Entretenimento	semi autônomo	Média	H(1) R(1)	H(Médio) R(Médio)	Médio
Espacial	teleoperado	Baixa	H(n) R(1)	H(Alto) R(Baixo)	Baixo
Manufatura	automatizado	Baixa	H(0) R(1)	H(Baixo) R(Baixo)	Alto
Militar	controle remoto	Baixa	H(n) R(n)	H(Alto) R(Baixo)	Baixo
Móvel	autônomo	Baixa	H(n) R(1)	H(Médio) R(Baixo)	Baixo
Residencial	semi autônomo	Alta	H(1) R(1)	H(Médio) R(Médio)	Médio

Fonte: Elaborada pelo autor.

Nota – A autonomia apresentada é a principal podendo mudar durante a execução da tarefa.

Nota – H(x)= x, atributo dos humanos na interação; R(y)= y, atributo dos robôs na interação.

Figura 10 – Taxonomia para métricas envolvendo HRI

Fonte: [Murphy e Schreckenghost \(2013\)](#).

os desenvolvedores de HRI utilizam métricas distintas para cada componente presente no ecossistema robótico: humanos, robôs e sistema. As características de qualidade associadas a cada um destes componentes são apresentadas a seguir.

Métricas HRI para Humano: avalia o conforto do ser humano ao utilizar o sistema. São utilizadas algumas características, tais como, precisão do modelo mental construído para utilizar o sistema, grau de esforço mental e medida da carga de trabalho (STEINFELD *et al.*, 2006).

Métricas HRI para Robô: Ao considerar somente o robô na avaliação, os pesquisadores levantam dados sobre as capacidades do robô durante sua utilização. Tolerância à negligência, ou seja, quanto tempo um robô pode evoluir sem supervisão direta, representa um exemplo deste componente. Outro exemplo, é tempo requerido em operações manuais executadas por humanos entre os ciclos de ação do robô.

Métricas HRI para Sistema: Neste grupo podemos observar uma clara influência da norma *Software product Quality Requirements and Evaluation* (SQuaRE) da *International Organization for Standardization* (ISO) / *International Electrotechnical Commission* (IEC) (ISO/IEC:25010, 2011) especificamente para qualidade de produto de software. Murphy e Schreckenghost (2013) adotaram cinco características:

- *Produtividade:* grau no qual o sistema oferece desempenho considerando a quantidade de recursos utilizados. Como exemplo desta métrica temos a produtividade da equipe.
- *Eficiência:* avalia o tempo e os recursos necessários para completar a tarefa. Métricas de eficiência incluem: esforço (por exemplo, número de movimentos), a razão homem-robô (onde se presume que necessitando de um número menor de humanos, a HRI seja mais eficiente) e tempo para completar a tarefa.
- *Confiabilidade:* grau no qual o sistema pode manter um dado nível de desempenho quando utilizado sobre condições específicas, tais como, falsos alarmes (ELARA *et al.*, 2010), flexibilidade e robustez, número de intervenções e tempo dispendido em operações manuais não programadas.
- *Segurança:* determina níveis aceitáveis de risco de danos às pessoas, negócio, dados, ou ambiente de utilização do sistema. Métricas de segurança incluem: risco para humanos e métrica relacionada à consciência do robô da presença de seres humanos.
- *Coatividade:* representa o grau de atividade conjunta entre robôs e humanos. Possui as seguintes métricas: nível de interação cognitiva, tolerância a negligência, percentagem de pedidos de assistência feitas pelo robô, percentagem de pedidos de assistência realizadas por operador.

Ainda segundo Murphy e Schreckenghost (2013), algumas deficiências são destacadas. A primeira está associada a abordagem de se considerar o agente (robótico ou humano) e não capacidades cognitivas ou autônomas como foco no estabelecimento das métricas. Outra deficiência é que, apesar do grande conjunto de métricas identificadas, muitas não possuem um mecanismo funcional ou genérico para medir uma dada característica.

Finalizando, na medida em que o grau de autonomia do robô aumenta, mais intrincada se torna sua avaliação. Quando humanos e robôs interagem em níveis cognitivos semelhantes, os usuários tendem a atribuir qualidades similares para ambos, humanos e robôs. Neste cenário, a avaliação de um mesmo robô pode ser polarizada por alterações de humor e sentimentos intrínsecos aos usuários humanos. Além disso, questões éticas ainda necessitam de respostas claras. Como exemplo, [Feil-Seifer e Matarić \(2009\)](#) levantaram algumas questões sobre a atuação do robô quando o usuário necessita de atenção de emergência, “qual é a responsabilidade do robô?”, ou ainda, “é aceitável que um ser humano trate seu robô como um escravo?”. Outra questão, envolvendo a hierarquia de comandos também possui implicações importantes: considere um robô cuidador, cujo paciente com demência declara claramente que não deseja ingerir medicamentos. Neste caso o robô deve respeitar, insistir ou entrar em contato com o nível superior na hierarquia da tarefa, ou seja, o cuidador humano?

3.2 Ciência cognitiva

A partir do momento em que robôs passam a interagir diretamente com seres humanos, é desejável que a comunicação seja realizada em um nível cognitivo apropriado para a interação em curso. Para alcançar este objetivo é necessário lançar mão de uma área da ciência ainda pouco familiar para os roboticistas: a psicologia.

Somente a compreensão do funcionamento da natureza humana permite sua modelagem e implementação em sistemas computacionais presentes em robôs. Esta seria uma estratégia simples se existisse consenso entre as diversas linhas das pesquisas presentes na psicologia. Infelizmente, esse consenso não existe, assim sempre que um roboticista citar uma abordagem em psicologia implica que, em algum momento, foi feita uma escolha de um modelo específico de representação da cognição. Para alinhar o raciocínio com o leitor, apresentamos a seguir duas definições do termo psicologia:

- [Train et al. \(2007, p.3\)](#) define psicologia como o estudo científico da natureza humana e a aplicação prática do que foi estudado. A compreensão da psicologia é útil em diversas áreas do ciclo da vida humana como educação, ambiente de trabalho, serviço social e medicina;
- Segundo [Kalat \(2014, p.3\)](#), psicologia é o estudo sistemático do comportamento e da experiência. Psicologia inclui pesquisa sobre sensações e percepção, aprendizado e memória, fome e sede, sono, atenção, desenvolvimento infantil e etc.

Na psicologia, uma área em particular nos interessa, a “psicologia cognitiva” que segundo [Sternberg e Sternberg \(2012, p.3\)](#) contemplam o estudo de como as pessoas percebem, aprendem, memorizam e refletem sobre a informação. Um psicólogo cognitivo pode, por exemplo, estudar

como as pessoas reconhecem diferentes formas, por que elas se lembram de alguns fatos e esquecem de outros ou como elas aprendem um idioma.

Os interesses comuns da psicologia cognitiva, inteligência artificial e linguística acabaram criando uma nova área denominada “Ciência Cognitiva”. O termo apareceu inicialmente em 1975 no livro *Explorations in Cognition* (NORMAN; RUMELHART, 1975, p.409) e rapidamente se disseminou. A ciência cognitiva tem como objetivo a compreensão empírica e teórica da cognição. Com o tempo, a compreensão do que é ser cognitivo tem se diversificado e naturalmente gerado controvérsias (ABRAHAMSEN; BECHTEL, 2012).

A Figura 11 apresenta a diversidade de linhas de pesquisa presentes em psicologia e ressalta que uma delas, psicologia cognitiva, possui forte conexão com a ciência cognitiva. Esta conexão é natural e desejada sendo que, muitas vezes, os temas de pesquisa se entrelaçam. Contudo, é importante ressaltar que são áreas com interesses distintos: na psicologia cognitiva, o foco é direcionado para “como o ser humano aprende, altera e recupera conhecimento”; na ciência cognitiva o foco é direcionado para “modelar e reproduzir os processos que levam ao conhecimento”.

Figura 11 – Psicologia X psicologia cognitiva X ciência cognitiva.

Fonte: Elaborada pelo autor.

3.2.1 Tópicos presentes na ciência cognitiva

A ciência cognitiva objetiva a pesquisa dos processos associados ao conhecimento realizados pela mente e sua reprodução em um ambiente computacional. A percepção do ambiente representa o ponto inicial do aprendizado, a partir dessa percepção modelos cognitivos são construídos e refinados. Como forma de organizar sistematicamente as linhas de pesquisa presentes na ciência cognitiva, utilizamos como base dois *handbooks* distintos: Reisberg (2013) que apresenta um viés direcionado para psicologia e Abrahamsen e Bechtel (2012) com foco

em modelamento e processamento da cognição. Os próximos parágrafos exploram os tópicos presentes nesses *handbooks*.

- **Percepção visual:** a visão representa um importante sentido humano e muitos pesquisadores buscam responder questões pertinentes sobre a forma de obter e organizar informações sobre o mundo visual, reconhecer objetos (HUMMEL, 2013), faces e movimento de olhos.
- **Atenção e conscientização:** a percepção compreende o processo de construir representações significativas para estímulos (visuais, sonoros, táteis, etc) presentes no ambiente. A atenção representa o primeiro passo nesse processo. Uma das metáforas mais comuns para compreender a atenção é considerá-la como um holofote “iluminando”, em um dado momento, somente uma quantidade limitada de informações. Assim, nem todos os estímulos presentes na região de percepção recebem igual quantidade de atenção, pelo contrário, alguns estímulos são selecionados para um acompanhamento intenso tornando-os mais propensos a atingir o limiar da consciência enquanto outros são processados apenas superficialmente, recebendo pouca atenção. Este modelo reconhece o gargalo de atenção presente no sistema cognitivo humano (BODENHAUSEN; HUGENBERGE, 2009). Nesse contexto, pesquisadores procuram compreender como focamos nossa atenção em regiões discretas no espaço e como este processo se realiza de forma automática e muitas vezes inconsciente.
- **Memória:** o estudo da memória está focado em como registramos e recuperamos informações e, por quanto tempo essa informação permanece disponível. Outra perspectiva está associada a memória prospectiva, ou seja, o desafio de lembrar, no futuro, para efetivar as intenções presentes. Ainda associada ao tema, Marsh e Butler (2013) pesquisam um tópico que possui implicações tanto pragmáticas quanto teóricas: o funcionamento da memória em ambientes educacionais.
- **Conhecimento e representação mental:** este tópico da ciência cognitiva está preocupado com a natureza do conhecimento e de sua representação mental. Como exemplo, Markman (2002) explora as diversas formas de representação mental cobrindo desde representações simples (espacial, de características e associativa) até as mais sofisticadas como representações estruturadas.
- **Texto e Linguagem:** psicolinguistas foram os principais contribuintes para a ciência cognitiva quando ela cresceu e amadureceu a partir da década de 70 (ABRAHAMSEN; BECHTEL, 2012). Nesta área, a atenção se volta para como as pessoas analisam, compreendem e produzem sentenças. Um exemplo de abordagem clássica é a utilização de regras sintáticas ($S \rightarrow NP + VP$)⁴ e semânticas (*ator + ação*).
- **O papel da emoção:** por muitos anos, os teóricos assumiram que fatores cognitivos e emocionais fossem separados ou então, estivessem em tensão mútua. Segundo Abrahamsen

⁴ S=Sentence, NP=Noun Phrase, VP=Verb Phrase

e Bechtel (2012), pesquisas recentes têm contestado esta perspectiva abordando temas como: percepção da emoção e a inter-relação entre emoção, estresse e memória.

- **Julgamento, raciocínio e escolha:** “como o ser humano toma decisões?”

Esta é a pergunta chave deste tópico e tem motivado pesquisas nas mais diversas frentes. Uma das primeiras abordagens é que o raciocínio depende da lógica. A lógica compreende duas partes: teoria da prova e teoria do modelo. A teoria da prova oferece regras formais de inferência para a prova, essas regras utilizam símbolos lógicos tais como $\rightarrow$ (implica), mas não define seu significado. A teoria do modelo define esse significado, por exemplo, a verdade de sentenças simples como “2 é maior que 1” em relação a um modelo, tal como o modelo dos números naturais. Como exemplo de teoria das provas temos a lógica proposicional, na qual expressões simbólicas compostas de proposições e conectivos, como $\vee$ (“ou”), $\wedge$ (“e”), e $\neg$ (“não”). Essas expressões são derivadas utilizando regras de inferência como: $(q \vee p) \rightarrow p$, onde p e q são duas proposições quaisquer, $(q \vee p)$ indica que q ou p ou ambos são verdadeiros, e o símbolo $\rightarrow$ indica que se a expressão à esquerda é verdade, então a expressão à direita também é verdade. A abordagem de regras de inferência proposta pela lógica enfrenta dificuldades em algumas situações (JOHNSON-LAIRD; KHEMLANI; P.GOODWIN, 2015). Como exemplo, a lógica convencional é monotônica, isto é, se uma inferência é válida, sua conclusão nunca precisa ser retirada - nem mesmo quando uma nova premissa a contradiz. Esta rigidez não é seguida pelos humanos que, quando confrontados com um fato sólido tendem a retirar qualquer conclusão que o contradiga. A “lógica de probabilidade” ou p – logic contorna as dificuldades da lógica convencional pressupondo que o grau de crença corresponde a probabilidade subjetiva, ou seja, na asserção *if A then B*, o conteúdo da cláusula *if* é acrescentado ao conjunto de crenças e então a probabilidade de B é calculada. Uma abordagem alternativa denominada “*Mental Model Theory*” propõe que indivíduos simulem o mundo através de modelos mentais para realizar previsões, ou seja, pessoas simulam possibilidades viabilizando a escolha de caminhos (JOHNSON-LAIRD; KHEMLANI; P.GOODWIN, 2015).

- **Solução de problemas e criatividade:** a solução de problemas está associada ao processamento cognitivo utilizado para alcançar um objetivo quando a priori não se conhece um método de solução. Em geral, o indivíduo utiliza conhecimento prévio para alcançar a solução de um novo problema. Os principais tipos de conhecimento necessários para a resolução de problemas são: fatos (terminologia e fatos específicos do problema), conceitos (inter-relação entre elementos básicos criando uma estrutura maior que habilitando seu funcionamento), procedimentos (técnicas e métodos, bem como critérios de utilização), estratégias e crenças. Os principais processos cognitivos na resolução de problemas são: representação, planejamento, execução e monitoramento (MAYER, 2002). Abordagens teóricas clássicas para a modelagem da resolução de problemas são: associacionismo (associação de ideias simples na resolução de problema complexo - “aprendendo com o erro”), Gestalt (solução de problemas ocorre pela compreensão estrutural, ou seja, observação

de como partes de um problema se encaixam na busca pela solução) e processamento de informações (raciocínio indutivo e dedutivo) (MAYER, 1992). Abordagens modernas incluem: tomada de decisão, inteligência e criatividade, ensino de habilidades de pensamento, pensamento matemático e científico e resolução de problemas por especialistas. O estudo de criatividade e *insight*⁵ também se encaixa neste tópico.

- **Como somos diferentes:** por muitos anos, a psicologia cognitiva enfatizou princípios que descreviam todos os seres humanos de forma global; considerações sobre diferenças individuais foram deslocadas para uma área exclusiva denominada *psychometrics*. Esta abordagem vem mudando nos últimos anos e questões sobre “como nós somos diferentes” passaram a fazer parte das pesquisas.
- **Desempenho prático e habilidoso:** a psicologia cognitiva tem tradicionalmente enfatizado “processos de entrada” e, na sequência, mentalização sobre as informações capturadas. O tópico “desempenho prático e habilidoso” foca no lado “saída” envolvendo planejamento e implementação da atuação no ambiente associado à qualidade do resultado produzido.

⁵ Ideia brilhante gerada “espontaneamente”.

3.3 Arquiteturas cognitivas

Na [Seção 3.2](#) foi descrito sobre a diversidade de conceitos envolvendo ciência cognitiva. Em uma visão pragmática, esses conceitos podem ser modelados e representados em sistemas computacionais para emular o processamento do conhecimento realizado pela mente humana.

Uma estratégia de implementação consiste em customizar o sistema para uma aplicação específica utilizando padrões clássicos de organização de software como *3-Tier* ou cliente-servidor. Adotando essa linha, [Munoz et al. \(2011\)](#) apresentou o projeto de um *hexapod*, denominado PTinto, para explorar de forma autônoma ambientes externos, em particular o rio Tinto na Espanha. A arquitetura utilizada é *3-Tier*, sendo uma camada para tarefas de planejamento e escalonamento com memória de longo prazo e possibilidades básicas de aprendizagem, outra camada é utilizada para implementar comportamento reativo com apoio de memória de curto prazo e, finalmente, a última camada representa o nível funcional para interface com a plataforma de operação do robô.

Outra estratégia de implementação, muito mais comum, consiste em utilizar arcabouços disponibilizados como componentes de programação que devem ser parametrizados e instanciados para aplicações específicas. Esta estratégia é conhecida como “arquitetura cognitiva” que, segundo [Gudwin et al. \(2018\)](#), “pode ser vista como um sistema de controle de propósito geral inspirado em teorias científicas desenvolvidas para explicar a cognição em animais e humanos”.

[Kotseruba, Gonzalez e Tsotsos \(2018\)](#) estimam que existam cerca de trezentas arquiteturas cognitivas implementadas⁶, sendo que, cerca de um terço se encontram em atividade. Dentre elas, *ACT-R*, *Clarion*, *Soar*, *EPIC* e *LIDA* são os mais citados na literatura.

O mesmo autor utiliza o “tipo de representação do conhecimento” para elaborar uma taxionomia das arquiteturas estabelecendo três grupos: simbólica, emergente (conexionista) e híbrida.

Sistemas simbólicos (*symbolic systems*) representam conceitos pelo uso de símbolos que podem ser manipulados através de um conjunto de instruções predefinidas, denominadas [regras de produção](#). Essas instruções são implementadas como regras *if-then* aplicadas aos símbolos que representam os fatos conhecidos do mundo.

Embora, por padrão, os sistemas simbólicos sejam excelentes em planejamento e raciocínio, eles são menos capazes de tratar a flexibilidade e a robustez necessária para modelar um ambiente em constante mudança, isto é, dinâmico.

A abordagem emergente (*emergent*) resolve as questões de adaptabilidade e aprendizagem com a construção de modelos massivamente paralelos, semelhantes a redes neurais, onde o fluxo de informação é representado como uma propagação de sinais dos nós de entrada. No

⁶ Diretórios na Web também constituem uma importante fonte de registro das arquiteturas cognitivas, como exemplo [BICA \(2012\) <<http://bicasociety.org/mapped/architectures/>>](#).

MAN; LAIRD; ROSENBLOOM, 1998), esta arquitetura possui como objetivo, criar um sistema computacional geral que tenha as mesmas habilidades cognitivas dos seres humanos. Como estratégia para alcançar esse objetivo, Soar integra: raciocínio baseado em conhecimento, execução reativa, raciocínio hierárquico, planejamento, aprendizado por experiência, aprendizado por reforço, memória semântica (armazena e recupera fatos sobre o mundo), memória episódica (armazena e recupera experiências) e um modelo baseado na avaliação de emoção (JOHNSON-LAIRD; KHEMLANI; P.GOODWIN, 2015). Nesta tese, Soar foi selecionada para validação da ontologia desenvolvida devido a sua relevância na área e utilização por outras pesquisas do laboratório LAR.

- ACT-R: a arquitetura cognitiva ACT-R (ACT, 2015) está organizada numa série de módulos, cada um responsável pelo processamento de um tipo diferente de informação associado com regiões específicas do cérebro humano. Por exemplo, o módulo visual está associado com a região occipital. Os módulos permitem: processamento visual e motor, alcançar objetivos e manter conhecimento de longo prazo (exemplo: “França é um país da Europa”). Cada módulo possui um *buffer* que, quando tomados em conjunto, representam o estado atual do sistema. A partir do estado atual é possível realizar buscas no módulo “memória procedural”, para obter produções que efetivamente implementem as atividades realizadas pelo ACT-R. Cada produção possui um custo (em termos de tempo necessário para atingir as metas) e probabilidade de sucesso. Em cada ciclo, ACT-R determina quais produções combinam com o estado do sistema, executando aquela com menor custo. Os resultados alcançados pelo sistema podem ser comparados ao comportamento humano através de métricas obtidas da psicologia cognitiva, a saber: tempo para executar uma atividade, precisão na execução e dados neurológicos para determinar quais áreas do cérebro foram ativadas. Os dados neurológicos são obtidos através da análise de imagem funcional de ressonância magnética (*Functional Magnetic Resonance Imaging* (fMRI)).
- MECA: no contexto nacional, destaca-se o trabalho de Paraense *et al.* (2016), no qual é proposto um *toolkit*, denominado *Cognitive Systems Toolkit* (CST), para a construção de arquiteturas cognitivas genéricas. Uma derivação deste trabalho é apresentada em Gudwin *et al.* (2018) que define a arquitetura cognitiva *Multi-purpose Enhanced Cognitive Architecture* (MECA), construída a partir do *toolkit* CST. Esta arquitetura implementa processamento baseado em regras e exploração de espaço de estados em módulos instanciados da arquitetura Soar, associado a um sistema dinâmico motivacional que utiliza subsunção dinâmica. A abordagem adotada pela arquitetura MECA é aderente a teoria de duplo processamento (*Dual Process Theory*), que particiona o sistema em dois sistemas. O primeiro, baseado em “Arquitetura de Subsunção Dinâmica”, recebe informações da percepção e atua no ambiente. O segundo é responsável pelo processo consciente do raciocínio deliberativo estando associado a um processamento baseado em regras que opera não somente com informação do presente, mas também do passado e previsões de futuro. A arquitetura MECA também define a ontologia *Object World Representation*

Language (OWRL), com o objetivo de representar formalmente objetos, suas agregações e composições, viabilizando o seu armazenamento na memória de trabalho.

Na área da robótica, o desejo de incorporar a cognição em robôs catalisou o surgimento de “arquiteturas cognitivas robóticas” (MOULIN-FRIER *et al.*, 2017), muitas baseadas em arquiteturas cognitivas clássicas como ACT-R e Soar.

Novikova, Watts e Inamura (2015) apresentaram uma plataforma denominada *SIGVerse* que possibilita a modelagem de um avatar humano interagindo com um agente robô em um ambiente 3D. A plataforma apresentada na Figura 13 consiste de um ambiente cliente-servidor, onde o lado servidor simula a inteligência do agente robótico e o lado cliente realiza a interação com o usuário. O componente *Service providers* habilita o controle do avatar humano pelo uso de interfaces, tais como *Wii*, *Kinect* e *Oculus Rift*. O *Wii* é usado para controlar o movimento de caminhar do humano, o *Kinect* é usado para controlar o tronco habilitando o humano a pegar objetos e executar gestos. O *Oculus Rift* incrementa o efeito da interação com o ambiente 3D. O aspecto cognitivo é alcançado pelo reconhecimento de duas emoções na interação: surpresa e alegria. Este é um trabalho inicial, no qual o aspecto cognitivo é minimizado pela captura de somente duas emoções, sendo o foco do trabalho direcionado para viabilizar a presença de dois agentes (humano e robô) no mesmo ambiente simulado.

Figura 13 – Sistema SIGVerse

Fonte: Novikova, Watts e Inamura (2015).

O trabalho de Davani, Ackers e Teyakome (2012) apresentou a utilização de redes neurais para controlar robôs autônomos que se movem em um ambiente desconhecido desviando de obstáculos. O ajuste dos parâmetros da rede neural é realizado com o apoio de algoritmos genéticos. Essa abordagem permitiu que, após treinamento com 500 gerações, o sistema evitasse 88% das colisões. Antes de instanciar o resultado em um robô real é possível realizar simulações em 3D com o uso da tecnologia de WebGL (KHRONOS, 2016), visualizando o resultado em um

browser. Vale ressaltar que neste trabalho o aspecto cognitivo é limitado somente ao treinamento da rede neural.

Alguns trabalhos buscam simular cognição em seres humanos ao invés dos robôs. [Furukawa \(2005\)](#) explora a simulação de operadores humanos atribuindo dinamicamente tarefas para um conjunto de 4 ou 8 robôs. Esse tipo de alocação é conhecida como “*Adaptable Automation*”, pois permite que as alocações sejam alteradas de acordo com as condições da linha de produção durante a execução. A carga de trabalho consumida pelo operador humano é calculada a partir de simulações Monte Carlo baseada em recursos cognitivos (visual, auditiva, cognitiva, motora e da fala). Na mesma linha de sistemas de manufatura, encontra-se o trabalho de [Faber et al. \(2013\)](#), que realiza o planejamento de tarefas de montagem considerando o conhecimento de operadores humanos absorvido pela análise das estratégias utilizadas durante a montagem de componentes mecânicos. Posteriormente, esse conhecimento é utilizado na construção de uma base com regras de produção utilizada no planejamento de manufatura.

Quando consideramos a aplicação de ciência cognitiva em robótica algumas questões ainda requerem uma análise mais aprofundada, como por exemplo ([BAXTER; LEMAIGNAN; TRAFTON, 2016](#)):

- como o ser humano deve ser considerado na preparação/seleção da ação atribuída para o agente robótico?
- se e como, a fisiologia e a psicologia humanas devam diretamente serem informadas (ou restringirem) as ações e a cognição presentes no agente robótico?
- a arquitetura cognitiva para robótica social deve ser baseada diretamente em modelos de comportamento humano?

3.4 Considerações finais

Neste capítulo foram abordadas duas importantes áreas presentes na robótica social: HRI e ciência cognitiva.

A percepção sensorial do ambiente representa o primeiro passo para viabilizar aspectos cognitivos como memória, aprendizado e tomada de decisão pelo agente robótico. As arquiteturas cognitivas representam um catalisador no desenvolvimento da robótica social suprimindo um ambiente ágil para o processamento de conhecimento.

Com a mesma importância, a HRI necessária para um “robô totalmente autônomo” contribui diretamente na complexa interação com seres humanos, envolvendo a representação e processamento de emoções e gestos, tanto por parte dos humanos quanto pelo robô.

DESENVOLVIMENTO

Conforme já mencionado, para alcançar o objetivo proposto nesta tese, foi elaborada uma ontologia para modelar a percepção do ambiente capturado pelo agente robótico. Esta percepção é traduzida por cinco sentidos similares aos presentes nos seres humanos. As questões associadas são:

- **Qual é o tipo da informação desejada?** O espectro de informação cognitiva é amplo e varia desde sensações até memória, emoções e criatividade. Nossa hipótese inicial assume que a ontologia, *OntPercept*, proposta nesta tese, seja utilizada na padronização da interface entre os sistemas cognitivo e robótico. Neste cenário, limitaremos o nosso interesse aos sentidos, ou seja: visão, audição, tato, paladar e olfato.
- **Como representar essa informação?** A representação dos sentidos não deve usar o *stream* de dados obtidos diretamente dos sensores. Pelo contrário, um nível de abstração superior é considerado na representação. Assim, tomando como exemplo o sentido audição, as informações seriam: palavras, sirene, pássaro, água fluindo, música, etc. imagina-ala
- **Meta informação?** Esta questão envolve o tratamento a ser dado às informações que podem ser obtidas pelo próprio sistema robótico sobre os dados associados aos sentidos. Vamos denominar essas informações como “meta informação”. Um aspecto envolve como agregar o “foco de atenção”¹ utilizado para obter a informação. Outro aspecto envolve o tratamento de emoções. Em geral a emoção é obtida a partir de expressões faciais reconhecidas pelo processamento de imagem. Outras abordagens também podem ser utilizadas, como exemplo, características emocionais que estão presentes no batimento cardíaco e na voz.

As próximas seções apresentam a metodologia utilizada ([Seção 4.1](#)), os conceitos e relacionamentos da ontologia ([Seção 4.2](#)) e as interfaces de aplicação desenvolvidas para sua

¹ “foco de atenção” foi descrito no item “Atenção e Conscientização” presente na [Subseção 3.2.1](#).

utilização concreta (Seção 4.3).

4.1 Metodologia

O desenvolvimento de uma ontologia pode ser acelerado pelo uso de metodologias e ferramentas de modelagem. Como exemplo, temos:

- para a geração do modelo de uma ontologia temos as seguintes metodologias : *Methontology*; (FERNÁNDEZ; GÓMEZ-PÉREZ; JURISTO, 1997) e “*Ontology Development 101*” (NOY; MCGUINNESS, 2001);
- para validar relacionamentos: OntoClean (GUARINO, 2009) ;
- para editar e manter uma ontologia: Protégé (PROTÉGÉ, 2018; MUSEN, 2015).

Nesta tese, foi escolhida a ferramenta Protégé para representar a ontologia *OntPercept*. Esta escolha foi realizada devido aos seguintes fatores: ser software livre; possuir baixa curva de aprendizado e larga disseminação na comunidade, sendo inclusive recomendada pela W3C (W3C, 2018). Diretamente associada ao Protégé, também foi escolhida a metodologia “*Ontology Development 101*” que propõe sete passos para a construção de uma ontologia:

1. *Determinar o domínio e o escopo da ontologia*: o início do desenvolvimento ocorre pela definição do seu domínio e escopo. Nesta tese, o domínio envolve agente robótico com aplicação em robótica social e o escopo está associado a percepção do ambiente.
2. *Considerar a reutilização de ontologias existentes*: o reuso oferece as seguintes vantagens: redução de custo, maior confiabilidade, uso eficaz de especialistas e conformidade com padrões. Na classificação apresentada por Guarino (1998)² pode-se associar o reuso à utilização de ontologias de nível superior (*top ontology*), que oferecem recursos para as ontologias de níveis mais básicos. A *OntPercept* reusa as ontologias definidas na norma 1872-2015 IEEE Standard Ontologies for Robotics and Automation (IEEE, 2015). A norma IEEE 1872-2015, por sua vez, utiliza a ontologia SUMO (PEASE, 2016) como *top ontology*.
3. *Enumerar conceitos importantes na ontologia*: a identificação dos conceitos presentes na ontologia está diretamente associada à definição de seu domínio. A *OntPercept* trata diretamente da percepção capturada pelo agente robótico. Neste contexto, os conceitos mais importantes estão associados ao modelo de percepção adotado e propriedades dos cinco sentidos.
4. *Definir os conceitos e sua hierarquia*: após a identificação dos conceitos relevantes, estes elementos são organizados em uma hierarquia revelando relacionamentos. Esta hierarquia

² A classificação proposta por Guarino foi explorada na Seção 2.1.

estabelece uma estrutura ontológica para a compreensão humana, bem como para a integração e reutilização de outras estruturas ontológicas. Existem três abordagens no desenvolvimento de uma hierarquia de classes: *top-down*, *bottom-up*, e combinação de ambas. No caso da *OntPercept*, a estratégia foi uma combinação de *top-down* e *bottom-up* ao utilizar conceitos de alto nível da IEEE 1872-2015 e do SUMO, para definir as características dos sentidos presentes no agente robótico e, ao mesmo tempo, identificar os relacionamentos associados à percepção do ambiente.

5. *Definir as propriedades dos conceitos*: a determinação da estrutura interna e atributos dos conceitos associados ao sentidos nasce naturalmente da revisão bibliográfica realizada no [Capítulo 2](#), envolvendo o uso de sentidos por agentes robóticos. A partir desta visão geral, a [Subseção 4.2.1](#) apresenta as propriedades selecionadas para cada conceito.
6. *Definir as facetas do conceito*: as propriedades podem ter diferentes facetas descrevendo seu tipo, valores permitidos (domínio) ou cardinalidade. Estas facetas representam axiomas que apoiam a definição da ontologia.
7. *Criar instâncias*: ao receberem valores concretos as propriedades determinam uma instância única do conceito na hierarquia definida. No caso da *OntPercept*, as instâncias são geradas automaticamente pelo agente robótico na medida que percebe o ambiente através dos sensores.

Após a elaboração do modelo é possível identificar anomalias com o uso da ferramenta *Pellet reasoner* ([STARDOG-UNION, 2016](#)), presente como plugin no editor *Protégé*.

4.2 O projeto da *OntPercept*

Uma importante característica do projeto da *OntPercept* é a aderência a padrões de organizações internacionais com o intuito de minimizar o esforço e prover uma maior aceitação dos resultados alcançados. Na [Figura 14](#), é apresentado o relacionamento com ontologias de nível superior, bem como, os principais conceitos da nova ontologia. O bloco superior desta figura detalha as *top ontologies* usadas como base, a saber: SUMO ([PEASE, 2016](#)) e IEEE 1872/2015 ([IEEE, 2015](#)). No bloco inferior, são apresentados os conceitos básicos da *OntPercept*.

SUMO é uma ontologia formal descrita por uma lógica de primeira ordem denominada SUO-KIF ([PEASE, 2011](#)). Sua base envolve cerca de 1000 conceitos e 4000 axiomas para cobrir noções de senso comum da realidade como: agentes, quantidades físicas, atributos biológicos, medidas de tempo, estados emocionais, eventos e processos. As ontologias presentes na ([IEEE, 2015](#)) têm por objetivo modelar sistemas robóticos e de automação, sendo também descrita pela linguagem SUO-KIF.

Figura 14 – *OntPercept* e ontologias superiores

Fonte: Elaborada pelo autor.

Ainda na [Figura 14](#), temos no pacote *OntPercept* seis conceitos básicos. O conceito *RobotPerceptionEvent* representa a superclasse usada como base para definir todos os sentidos presentes no ambiente. Todos os outros conceitos derivam desta superclasse e envolvem captura de informações dos cinco sentidos associados ao agente robótico. As propriedades e facetas de cada um destes elementos são detalhadas a seguir.

4.2.1 Definição dos conceitos

A definição dos conceitos representa um dos passos iniciais no desenvolvimento de uma nova ontologia segundo a metodologia *Ontology Development 101*. Na definição das propriedades associadas aos conceitos da *OntPercept*, focou-se na interpretação das informações reportadas pelo sensor em lugar de reportar diretamente os dados coletados. Por exemplo, em vez de reportar a direção/magnitude das quantidades físicas associadas ao sentido de Tato, são consideradas pressão, rugosidade, umidade e dureza.

Para representar medidas é necessário adotar uma padronização definindo unidade de cada grandeza. A ontologia *OntPercept* disponibiliza duas formas de representar os sinais capturados. A primeira utiliza valores parametrizados na faixa entre 0 e 1. A segunda disponibiliza exatamente os valores obtidos considerando normas de padronização de medidas internacionais, ou seja, Sistema Internacional de Unidades (*Système International d'unités* (SI)).

4.2.1.1 Conceito *RobotPerceptionEvent*

O conceito *RobotPerceptionEvent* representa a super classe da qual derivam os outros conceitos associados aos sentidos. No [Quadro 6](#), são apresentado suas propriedades e valores associados.

Quadro 6 – Propriedades do conceito *RobotPerceptionEvent*

Propriedade	Descrição	Valores
<i>generateBy</i>	Objeto gerador do evento.	Índice para objeto associado.
<i>hasCaptureTime</i>	O instante de captura do evento.	Instante de captura no calendário Gregoriano. Formato: [-] CCYY-MM-DDThh:mm:ss[Z (+ -)hh:mm]
<i>isSenseOf</i>	Identifica o agente robótico responsável pela captura.	Id. do agente robótico responsável pela captura.

4.2.1.2 Conceito *RobotTaste*

Segundo [Kobayashi e Ikezaki \(2013\)](#), uma das dificuldades presentes em sua pesquisa estava associada à definição de uma unidade apropriada para as informações geradas pelos sensores de sabor. Diversas abordagens são utilizadas para apresentar a concentração ou relacionar a quantidade de um [analito](#) numa amostra: concentração comum, molaridade, percentual, título, fração molar, molalidade e partes por milhão, são alguns exemplos. Essas representações são utilizadas nos processos químicos e naturalmente poderiam ser nossa unidade de informação para as medidas associadas ao paladar. Contudo, antes de aprofundar a questão sobre a unidade de informação é interessante apresentar algumas das características associadas ao processamento de estímulos químicos em seres humanos, em particular: a lei de Weber-Fechner ([HEIDELBERGER, 2001](#)) e a interação entre substâncias. A lei de Weber-Fechner afirma que:

1. Considerando seres humanos, a relação entre a intensidade inicial (I) para estímulos e o [limiar de discriminação](#) é uma constante k (a fração de Weber), ou seja: $\Delta I = k.I$. Considerando a sensibilidade humana, o menor aumento detectável para o sentido gustativo de salgado, doce e [umami](#) é de cerca de 20%.
2. A resposta R a qualquer estímulo é proporcional ao logaritmo da intensidade I do estímulo, ou seja: $R = \log(I)$.

Outra característica importante está associada às interações entre substâncias, reduzindo ou aumentando a intensidade do sabor percebido pelo ser humano. Em um exemplo bem conhecido, o amargor do café é reduzido pela adição de uma porção de açúcar. Este fato ocorre devido a uma interação específica que faz com que uma substância doce possa mascarar o amargor suprimindo um dado sabor.

Tanto a lei de *Weber-Fechner* quanto as interações entre substâncias ressaltam o comportamento subjetivo associado aos sentidos presentes nos seres humanos. Um dispositivo que

apresente exatamente a molaridade de um **analito** em uma amostra é exata sob o ponto de vista químico, mas insuficiente sob o ponto de vista da sensação do paladar.

Nesse contexto, considerando nosso objetivo de representar o paladar, seria de pouca valia obter exatamente a quantidade de açúcar e de amargor e ignorar sua interação. Assim, será adotada uma normalização adimensional para cada sabor, mantendo valores parametrizados de 0 (menor concentração) até 1 (maior concentração). No **Quadro 7**, encontra-se uma lista das propriedades associadas ao paladar (*Taste*).

Quadro 7 – Propriedades associadas ao conceito *RobotTaste*

Propriedade	Descrição	Valores
<i>sweetness</i>	Concentração do sabor doce.	Entre 0 e 1.
<i>sourness</i>	Concentração do sabor ácido.	Entre 0 e 1.
<i>saltiness</i>	Concentração do sabor salgado.	Entre 0 e 1.
<i>bitterness</i>	Concentração do sabor amargo.	Entre 0 e 1.
<i>umami</i>	Concentração do sabor umami.	Entre 0 e 1.

Fonte: Elaborada pelo autor.

4.2.1.3 Conceito RobotSmell

As propriedades associadas ao olfato são apresentadas no [Quadro 8](#).

Quadro 8 – Propriedades associadas ao conceito *RobotSmell*

Propriedade	Descrição	Valores
<i>isPositionedAt</i>	referência para indivíduo que mantém a posição do objeto gerador do odor.	Coordenadas cartesianas (x,y,z).
<i>fragrantSmell</i>	perfumado. Exemplos: perfume, floral, rosa, violeta, etc.	Entre 0 e 1.
<i>woodySmell</i>	amadeirado. Exemplos: cedro, ervas, grama cortada, terra, mofo, etc.	Entre 0 e 1.
<i>fruitySmell</i>	frutado (não cítrico). Exemplos: abacaxi, cereja, banana, morango, etc.	Entre 0 e 1.
<i>chemicalSmell</i>	químico. Exemplos: medicinal, anestésico, desinfetante, ácido, leite azedo, etc.	Entre 0 e 1.
<i>decayedSmell</i>	fétido. Exemplos: putrefato, sujeira, rançoso, fezes, etc.	Entre 0 e 1.
<i>mintySmell</i>	mentolado. Exemplos: Fresco, aromático, anis, alcaçuz, etc.	Entre 0 e 1.
<i>é sweetSmell</i>	doce. Exemplos: baunilha, chocolate, amêndoa, caramelo, etc.	Entre 0 e 1.
<i>popcornSmell</i>	pipoca. Exemplos: amanteigado, pasta de amendoim, oleoso, noz, amêndoa, etc.	Entre 0 e 1.
<i>pungentSmell</i>	acre. Exemplos: alho, cebola, queimado, fumaça, etc.	Entre 0 e 1.
<i>lemonSmell</i>	limão. Exemplos: cítrico, laranja, frutado (cítrico), etc.	Entre 0 e 1.

Fonte: Elaborada pelo autor.

A identificação do odor é produzida a partir de dez tipos de odores básicos, denotados por: $\{W_1, W_2, \dots, W_{10}\}$, seguindo a classificação criada por [Castro, Ramanathan e Chennubhotla \(2013\)](#). Cada um desses tipos de odores foram representados como *Data property assertions* associadas ao conceito. A proposta é realizar uma composição desses odores parametrizados na faixa de $[0, 1]$. Assim, temos:

$$\vec{odor} = (o_1, o_2, \dots, o_{10})$$

Nesta representação cada $o_i, i = 1, \dots, 10$ representa uma ponderação sobre os odoríferos presentes no [Quadro 8](#), ou seja, o_1 corresponde à ponderação sobre *fragrantSmell*, o_2 corresponde à ponderação sobre *woodySmell* e assim sucessivamente até o_{10} . A proposta é realizar uma composição similar ao espaço RGB utilizado para classificar cores. Nesse contexto um “branco olfativo” corresponde a um vetor com 10 posições zeradas: $\vec{odor} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$.

Através do olfato, o agente robótico tem conhecimento de que partículas odoríferas estão sendo produzidas em algum ponto do ambiente, contudo, não possui sua posição exata, somente uma direção para seguir através da propriedade *isPositionedAt*. Naturalmente, a decisão

de acompanhar a nuvem odorífera é exclusiva do sistema cognitivo. A partir do momento que o objeto responsável pela geração do odor é avistado, o agente robótico passa a enviar o identificador do objeto³ em vez da direção da nuvem de odor.

4.2.1.4 Conceito *RobotHear*

A estratégia adotada para representar as informações sonoras consiste em definir uma enumeração dos tipos de sons possíveis no ambiente. Além dessa informação, também é modelado a intensidade do som e, quando necessário, sua descrição. Por exemplo, para o caso da fala, apresenta a sentença sendo pronunciada; para o caso de música, apresenta o nome da música sendo executada; para pássaro, apresenta seu tipo; etc. No [Quadro 9](#), são apresentados detalhes dessas propriedades.

Quadro 9 – Propriedades associadas ao conceito *RobotHear*

Propriedade	Descrição	Valores
<i>isPositionedAt</i>	Referência para indivíduo que mantém a posição do objeto gerador do som.	Coordenadas cartesianas (x,y,z).
<i>volume</i>	Intensidade do som capturado.	Entre 0 e 1.
<i>detail</i>	Descrição ou informação adicional associada ao som.	Valores do tipo <i>String</i> .
<i>hasSoundType</i>	Tipo do som emitido. Por exemplo, campainha, motor, música, etc.	<ul style="list-style-type: none"> ● <i>unknownSound</i> ● <i>barkingSound</i> ● <i>bellSound</i> ● <i>birdsSound</i> ● <i>liquidFlowingSound</i> ● <i>motorSound</i> ● <i>musicSound</i> ● <i>tvSound</i> ● <i>humanVoiceSound</i> ● <i>robotVoiceSound</i>

Fonte: Elaborada pelo autor.

Através da audição, o agente robótico tem conhecimento de que algum som está sendo emitido em determinado ponto do ambiente, mas nem sempre é capaz de estabelecer exatamente o objeto gerador deste estímulo. Contudo, ele é capaz de identificar a posição provável da origem do som, permitindo que a arquitetura cognitiva envie comandos para mover o agente robótico. A partir do momento que o elemento responsável pelo som é avistado, com o apoio da visão, o agente robótico passa a enviar o identificador do objeto em vez de sua posição.

³ Propriedade *generateBy* do conceito *RobotPerceptionEvent*, super classe de *RobotHear*.

4.2.1.5 Conceito *RobotTouch*

O modelo elaborado para o conceito *RobotTouch* assume que sensores de tato estão dispostos em cada uma das “mãos” do agente robótico. Os sensores captam informações táteis do objeto desde que exista um contato entre a “mão” do agente robótico e o objeto. Estas informações dizem respeito à temperatura, pressão, rugosidade, umidade e rigidez do objeto sendo apresentadas no [Quadro 10](#). Essas características são modeladas como *Data property assertions* associadas ao conceito.

Quadro 10 – Propriedades associadas ao conceito *RobotTouch*

Propriedade	Descrição	Valores
<i>isPositionedAt</i>	Referência para indivíduo que mantém a posição do objeto identificado pelo tato.	Coordenadas cartesianas (x,y,z).
<i>temperature</i>	Temperatura em graus Célsius.	Decimais (<i>double</i>) positivos e negativos (°C).
<i>pressure</i>	Força que o objeto exerce sobre o sensor.	Entre 0 e 1.
<i>roughness</i>	Nível de aspereza do objeto.	Entre 0 e 1.
<i>moisture</i>	Umidade presente no objeto.	Entre 0 e 1.
<i>hardness</i>	Resistência do objeto quando uma força é aplicada em sua superfície.	Entre 0 e 1.

Fonte: Elaborada pelo autor.

O atributo *isPositionedAt* representa a posição do contato dos sensores presentes na “mão” do agente robótico com o objeto. Quando o objeto é visível, pela sentido da visão, o atributo *generateBy* da super classe *RobotPerceptionEvent* fornece o identificador do objeto.

4.2.1.6 Conceito *RobotVision*

O conceito *RobotVision* estabelece as propriedades necessárias para o reconhecimento de objetos presentes no ambiente. Apesar de sua potencial capacidade de gerar indivíduos através do conceito *Object*, este é o conceito mais simples, pois, simplesmente utiliza as propriedades herdadas do conceito *RobotPerceptionEvent*, a saber: *generateBy*, *hasCaptureTime* e *isSenseOf*.

4.2.1.7 Conceito *Object*

O conceito *Object* modela todos elementos físicos presentes no ambiente e está diretamente relacionado com os sentidos, os quais são subclasses do conceito *RobotPerceptionEvent*. Conforme apresentado na [Capítulo 1](#), os conceitos da ontologia *OntPercept* disponibilizam representações simbólicas com alto nível de abstração. Nesse contexto, assumimos como hipótese que informações de *Mesh 3D* associadas aos objetos serão supridas diretamente pela Web Semântica à partir da propriedade *associateURI*.

No [Quadro 11](#), são apresentadas as propriedades associadas aos objetos capturando sua essência, posição e aparência.

Quadro 11 – Propriedades associadas ao conceito *Object*

Propriedade	Descrição	Valores
<i>objectID</i>	Armazena identificador único do elemento.	Número inteiro.
<i>tag</i>	Tipo do objeto. Ex. parede, porta, gaveta, chão, torneira, interruptor, etc.	Valores do tipo <i>String</i> .
<i>name</i>	Nome do objeto associado.	Valores do tipo <i>string</i> .
<i>cartesianPosition</i>	Referência para indivíduo que mantém a posição do objeto no ambiente.	Coordenadas cartesianas (x,y,z).
<i>isMadeOf</i>	Tipo de material predominante na composição do objeto.	<ul style="list-style-type: none"> • <i>UnknownMaterial</i> • <i>WoodMaterial</i> • <i>MetalMaterial</i> • <i>PlasticMaterial</i> • <i>RubberMaterial</i> • <i>GlassMaterial</i> • <i>RockMaterial</i> • <i>OrganicMaterial</i> • <i>PaperMaterial</i> • <i>ClothMaterial</i>
<i>hasColor</i>	Referência para indivíduo que mantém a cor predominante do objeto utilizando o sistema de cores aditivas, RGB.	Valores (RGB) entre 0 e 1.
<i>hasSmellType</i>	Referência para indivíduo que mantém informações sobre o odor associado.	Composição parametrizada de dez odores básicos.
<i>hasInternalState</i>	Estado atual do objeto. Geralmente associado aos estados físicos de um objeto inanimado como portas e interruptores.	<ul style="list-style-type: none"> • <i>NoState</i> • <i>OnState</i> • <i>OffState</i> • <i>OpenedState</i> • <i>ClosedState</i>
<i>hasEmotionalState</i>	Armazena emoções humanas. Objetos não possuem este atributo.	<ul style="list-style-type: none"> • <i>NoEmotion</i> • <i>AngerEmotion</i> • <i>DisgustEmotion</i> • <i>FearEmotion</i> • <i>HappinessEmotion</i> • <i>SadnessEmotion</i> • <i>SurpriseEmotion</i>
<i>associateURI</i>	Armazena um localizador ou nome do objeto associando-o à Web Semântica .	<i>Link</i> para endereço Web.

Fonte: Elaborada pelo autor.

4.2.2 Construindo a ontologia

Uma vez definidos os conceitos básicos da *OntPercept*, é possível estabelecer os relacionamentos com as ontologias de nível superior IEEE 1872-2015 e SUMO. Um ponto importante nesse processo está associado à seleção da linguagem utilizada para representar a ontologia. Conforme apresentado na [Figura 2](#), na medida que a linguagem oferece maior expressividade, aumenta sua formalidade e por conseguinte a complexidade de seu processamento⁴.

⁴ Maiores detalhes sobre este tópico podem ser encontrados na [Seção 2.1](#)

O grupo que desenvolveu a IEEE 1872-2015 apresentou um experimento prático utilizando as ontologias CORA e POS, para modelar o envio de um objeto com o apoio de robôs (JORGE *et al.*, 2015). Os insumos presentes nesse experimento são: um humanoide NAO da SoftBank (atuando como manipulador); um robô Pioneer 3DX (atuando como transporte); um servidor de ontologia e dois agentes humanos. Os autores modelaram as ontologias da IEEE 1872-2015 na linguagem OWL. A utilização da OWL simplificou a implementação e a validação devido ao uso de ferramentas consolidadas para Web Semântica. A partir de contato com o grupo responsável pelo experimento foi possível obter o arquivo com a descrição eletrônica das ontologias utilizadas neste experimento, minimizando o esforço de construção da *ontPercept*. Esta troca de informações entre grupos distintos representa um dos aspectos positivos na utilização de hierarquia de ontologias.

Quadro 12 – Variações de terminologia utilizadas pela comunidade

LPO	OWL	DL	Protégé
<i>unary predicate</i>	<i>class</i>	<i>concept</i>	<i>class</i>
<i>constant</i>	<i>individual</i>	<i>individual</i>	<i>instance</i>
<i>binary predicate</i>	<i>property</i>	<i>role</i>	<i>property</i>

Fonte: Elaborada pelo autor.

Nota – LPO representa Linguagem de Primeira Ordem

Conforme apresentado na Seção 4.1, a ferramenta Protégé (PROTÉGÉ, 2018) foi utilizada para editar e manter a ontologia *OntPercept*. Dessa forma, a partir deste ponto, será utilizada a terminologia utilizada pelo Protégé para referenciar os elementos associados a ontologia. No Quadro 12, é apresentada a terminologia utilizada por cada tecnologia referenciada nesta pesquisa, a saber, Lógica de Primeira Ordem (LPO), OWL, *Description Logic* (DL) e Protégé).

A utilização da terminologia “classe” (presente no Protégé) para representar um “objeto que agrega propriedades” deve ser interpretada com cautela, principalmente pelo leitor acostumado com o uso do termo em orientação a objetos. As terminologias “predicado unário”, “classe” e “conceito” são melhores associados com a definição de conjuntos presentes na matemática, ou seja, novos conjuntos podem ser gerados pela união, intersecção ou diferença de elementos.

O Protégé utiliza a linguagem OWL-DL que representa uma sublinguagem da OWL-Full com limitações na utilização de construtores para ganhar eficiência computacional.

Na Figura 15, é apresentada a tela inicial da ferramenta após o carregamento da ontologia *OntPercept*. Nela são apresentadas informações gerais que são detalhadas a seguir.

A definição da *Internationalized Resource Identifier* (IRI) encontra-se na janela *Ontology header*, que representa o identificador de recursos associado à ontologia, neste caso: <http://example.org/>. Esta IRI está relacionada ao domínio */net/OntPercept* registrado como endereço permanente da *OntPercept* no site PURL (PURL-TEAM, 2018).

Figura 15 – Tela principal da ferramenta Protégé representando a *OntPercept*

Fonte: Elaborada pelo autor.

A janela *DL metrics* revela a expressividade DL (BAADER; HORROCKS; SATTLER, 2009) alcançada pela ontologia. A expressividade representa a habilidade da linguagem em exprimir: propriedades inversas, negação, restrições universais, restrições de cardinalidade, dentre outras. No caso da *OntPercept* a expressividade está associada ao tipo *SRIQ(D)* que indica, *S*: ontologia é modelada por uma linguagem atributiva⁵; *I*: utiliza propriedades inversas; *Q*: utiliza restrições de cardinalidade; *(D)*: utiliza propriedades com *datatypes* ou *data values*. Na verdade, essa indicação não é selecionada pelo usuário do Protégé, mas sim, automaticamente calculada pela ferramenta na medida que o desenvolvimento da ontologia evolui.

A métrica da ontologia é apresentada na janela *Ontology metrics* na qual, por exemplo pode-se verificar que foram definidos 723 axiomas e 82 classes. Estes valores representam o valor global e agregam as *top ontologies* importadas pela ferramenta.

Finalmente, tem-se a janela *Imported ontologies*, que apresenta a IRI das ontologias “basicPos” e “CORA” definidas pela norma IEEE 1872-2015. Embora não visível na Figura 15, as ontologias importadas indiretamente também compõem o modelo final. Nesse grupo tem-se, por exemplo, a ontologia SUMO, *top ontology* utilizada pelas ontologias da norma IEEE 1872-2015.

Uma vez definidos os conceitos e as ontologias importadas, é possível iniciar a definição dos relacionamentos e axiomas associados a ontologia *OntPercept*. Na Figura 16, são apresentados os relacionamentos estabelecidos entre a classe *RobotPerceptionEvent* e as ontologias

⁵ A linguagem atributiva permite: negação atômica, intersecção, restrição universal e quantificador existencial limitado.

tipo de dados” *occursAt*, que relaciona a percepção de sentido com o instante de sua ocorrência. O código com a descrição completa da ontologia *OntPercept* é disponibilizado no Seção A.3, sendo também possível o acesso eletrônico no repositório [GitHub](https://github.com/helioaz/ontPercept)⁶.

Código-fonte 1 – Exemplos de elementos em linguagem Manchester

```

1:
2: # // CLASS
3: Class: RobotHear
4: Annotations:
5:   rdfs:comment "The subclass of RobotPerception in which the sensing is
   done by an auditory sensor."@en
6: EquivalentTo:
7:   RobotPerceptionEvent
8:   and (hasSoundType exactly 1 HearingAttribute)
9: SubClassOf:
10:   RobotPerceptionEvent,
11:   isPositionedAt some CartesianPositionPoint
12:
13: # // OBJECT PROPERTY
14: ObjectProperty: isMadeOf
15: Annotations:
16:   rdfs:comment "type of material that composes the object"@en
17: SubPropertyOf:
18:   owl:topObjectProperty
19: Characteristics:
20:   Functional
21: Domain:
22:   Object
23: Range:
24:   Material
25:
26: # // DATA PROPERTY
27: DataProperty: occursAt
28: Annotations:
29:   rdfs:comment "Moment of occurrence perception"@en
30: Characteristics:
31:   Functional
32: Domain:
33:   TimePoint
34: Range:
35:   xsd:dateTime
36: SubPropertyOf:
37:   owl:topDataProperty

```

Na Figura 17, é apresentado um exemplo envolvendo a captura de informação sensorial associada ao sentido de olfato. O elemento *O00000000005* representa uma instância da classe *Object*. A instância *O00000000005* possui relacionamentos que definem sua posição (*isPositionedAt*), sua cor (*hasColor*), sua composição (*isMadeOf*) e seu odor (*hasSmellType*). O objeto *S1961080841* representa uma instância da classe *RobotSmell* que, por sua vez, mantém a identificação do odor como composição de dez odoríferos.

⁶ <https://github.com/helioaz/ontPercept>

Figura 18 – Detalhes das propriedades de uma instância da classe *RobotSmell*

Fonte: Elaborada pelo autor.

com apoio da ontologia *OntPercept*, utilizando um protocolo de comunicação pré-definido como XML ou REST/HTTP, por exemplo. A arquitetura cognitiva processa a percepção do ambiente, toma decisões e envia para o agente robótico através de comunicação via socket.

Figura 19 – Visão geral dos nós de processamento associados ao *OntPercept*

Fonte: Elaborada pelo autor.

A implementação do modelo apresentado na Figura 19 é facilitada pelo uso de compo-

nentes disponibilizados na Web para acesso aos recursos de ontologias. Assim, para implementar o nó *OntPercept Server Node* é utilizada a *triple store* Fuseki (FUSEKI-TEAM, 2017), sendo a comunicação com o Fuseki realizada através de SPARQL (WEB-ONTOLOGY-WG, 2017; DUCHARME, 2013).

SPARQL é uma linguagem de consulta para RDF, recomendada pela W3C (W3C, 2018), que rapidamente se tornou padrão na comunidade. Ela opera de forma similar as linguagens de acesso a banco de dados onde questões são formuladas e submetidas a um servidor que produz e retorna as respostas.

Fuseki implementa o acesso às triplas RDF através de um ponto de acesso SPARQL com o uso do protocolo REST. Ele faz parte do arcabouço Jena (JENA-TEAM, 2018) que agrega diversas ferramentas de software livre para acesso à Web Semântica criadas com a linguagem Java.

Com o objetivo de reduzir o esforço do usuário final da ontologia *OntPercept*, foram criadas duas APIs de acesso ao nó *OntPercept Server Node*, que criam uma interface simples de uso das tecnologias citadas nos parágrafos anteriores.

A primeira API, implementada na linguagem Java, supre o acesso da arquitetura cognitiva com o ponto de acesso SPARQL. A segunda, implementada na linguagem C#, viabiliza a comunicação entre o simulador robótico e a *triple store* Fuseki.

A necessidade de APIs para linguagens distintas nasce da linguagem e sistemas operacionais utilizados pelos componentes selecionados para implementar os nós *Cognition Node* e *Robotic Agent Node*, respectivamente, Java no Linux e C# no Windows. Naturalmente, existem estratégias para diretamente utilizar bibliotecas de outras linguagens nos arcabouços do Java e C#, tal como, a interface JNI para o Java. Contudo, essas estratégias implicam na ativação de máquinas virtuais adicionais, seja para o Java (JVM) ou para o C# (CLR), que acabam por reduzir o desempenho do sistema ou aumentam a probabilidade de erros na codificação/execução. Assim, as próximas seções detalham as duas APIs desenvolvidas.

Finalmente vale observar que o nó *Robotic Agent Node* pode ser instanciado como um robô ou um simulador. Nesta tese, a verificação e validação do modelo proposto foi realizada com o desenvolvimento de um simulador desenvolvido especificamente para essa finalidade. Nesse contexto, o nó *Robotic Agent Node* é instanciado como um simulador e passa a ser identificado como *Simulator Node* nas próximas figuras.

4.3.1 API *ontPerceptCSharp*

A “*OntPerceptCSharp*” foi elaborada em linguagem C# para a plataforma Windows. Seu objetivo é fornecer uma interface simples e segura para acesso a *triple store*, que mantém a ontologia *ontPercept*.

A versão atual implementa operações de atualização, possibilitando a inserção de eventos de percepção do ambiente capturados pelos sensores presentes no robô. O processo de inserção dos eventos ocorre em duas etapas: na primeira, a construção do objeto em C# associado ao evento é realizada e, na segunda, o envio do evento para a *triple store*.

Na Figura 20, são explicitados os componentes que integram os nós de processamento. Neste caso, encontra-se o *Simulation Node*, que utiliza dois componentes. O primeiro representa o *Robot House Simulator (RHS)*⁸ que implementa um simulador robótico e o segundo envolve a *API OntPerceptCSharp*, responsável pela interface entre o simulador e o Fuseki.

Figura 20 – Visão geral das APIs associadas ao *OntPercept*

Fonte: Elaborada pelo autor.

O diagrama de classes em UML, presente na Figura 21, representa as classes, no contexto de Orientação a Objetos, criadas para modelar a API. As classes apresentadas em azul estão diretamente associadas aos conceitos criados para modelar os sentidos na *OntPercept*. Um diagrama mais detalhado da API, com atributos e operações, é apresentado na Seção A.1. Nesse diagrama, pode-se ver a classe *SparqlAccess*, que implementa a comunicação com *triple store* via linguagem SPARQL. A implementação dessa classe utiliza a biblioteca *dotNetRDF* (AHMED, 2016), que oferece recursos para trabalhar com SPARQL em ambientes .NET da Microsoft.

O código presente no Código-fonte 2 detalha a inserção do objeto *Salmon_Pack* na *triple store* Fuseki. A linha 23 registra a chamada ao método *insert()*, o qual efetivamente cria e envia o comando SPARQL com apoio da biblioteca *dotNetRDF*.

Código-fonte 2 – Exemplo da inserção de objeto na *triple store*

```

1:
2: // Start access to Sparql End Point: just one time at main method is enough
3: SparqlEndPoint instanceSparql = SparqlEndPoint.getInstance();
4: instanceSparql.init (ONT_SENSE_URL);
  
```

⁸ A descrição do RHS é realizada na Seção 5.1.

Figura 21 – Diagrama de classes para API de interface com a *OntPercept*

Fonte: Elaborada pelo autor.

```

5:
6: CartesianPos cPos = new CartesianPos(0.35, 0.36, 0.37); // creates a
    CartesianPos object
7: RGBValue rgb = new RGBValue(0.45, 0.46, 0.47); // creates a RGBValue
8: Thing thing = new Thing(
9:     123456789,                // identifier
10:    "",                        // name
11:    "Salmon_Pack",            // tag,
12:    rgb,                      // color RGBValue
13:    cPos,                    // CartesianPos
14:    PhysicalState.noneState,  // Physical State
15:    Material.organicMaterial, // Material type
16:    "https://www.wikidata.org/wiki/Q7405484"); // URI associated
17:
18: // creates a RobotVision instance seen at moment
19: RobotVision rVision = new RobotVision(DateTime.Now, thing);
20:
21: try
22: {
23:     rVision.insert(); // dotNetRDF inserts data in the triple store
24: }
25:
26:
27: catch (Exception e)
28: {
29:     Console.WriteLine(e); // Call a custom error logging procedure.
30:     throw;                // Re-throw the error.
31: }

```

4.3.2 API *ontPerceptJava*

A “*ontPerceptJava*” foi desenvolvida em linguagem Java para a plataforma Linux utilizando o arcabouço Jena ([JENA-TEAM, 2018](#)), como ferramenta de suporte. O objetivo da API é fornecer uma interface simples e segura com base na linguagem SPARQL para acesso à *triple store*, mantendo as informações descritas pela ontologia *OntPercept*.

Na [Figura 20](#), tem-se esta API presente no nó *Cognition Node* fornecendo suporte para o componente que implementa a arquitetura cognitiva. A versão atual da API disponibiliza operações de consulta, possibilitando recuperar informações de eventos de percepção do ambiente capturados pelos sensores presentes no robô.

Após a efetivação da consulta, os registros recuperados são imediatamente removidos da base de dados, não havendo possibilidade de recuperar o mesmo registro novamente. Os dados recuperados são organizados localmente em uma estrutura do tipo *Map*, possuindo como chave de acesso o identificador do evento de percepção. Essa estrutura permanece disponível para uso da aplicação até que nova consulta seja realizada.

A estrutura das classes utilizadas para modelar esta API segue o diagrama de classes presente na [Figura 21](#), a mesma estrutura utilizada para a “*OntPerceptCSharp*”. A diferença principal está associada ao uso do arcabouço *Jena* para acesso a *triple store*.

4.4 Considerações finais

O desenvolvimento da ontologia *OntPercept*, apresentado neste capítulo, envolveu diversos passos e tecnologias.

Inicialmente, foi detalhada a definição dos conceitos para modelar os sentidos necessários para percepção do ambiente pelo agente robótico. Essa definição foi embasada pelos conhecimentos estabelecidos durante a pesquisa bibliográfica realizada no [Capítulo 2](#).

No passo seguinte, a ontologia foi elaborada utilizando como ferramenta o editor Protégé ([PROTÉGÉ, 2018](#)). A metodologia utilizada para esta atividade foi a “*Ontology Development 101*” ([NOY; MCGUINNESS, 2001](#)), que estabelece 7 passos para a criação de novas ontologias.

Uma vez definida a ontologia, foi selecionada a *triple store* Fuseki ([FUSEKI-TEAM, 2017](#)) para manter as triplas RDF acessadas com o apoio da linguagem de consulta SPARQL.

O volume de software gerado nesta etapa alcança 4699 linhas de código, sendo 1759 associadas a “*OntPerceptCSharp*” e 2940 linhas associadas a “*OntPerceptJava*”. Todo esse conjunto de software, bem como tutoriais, projetos e modelos foram disponibilizados no [GitHub](#)⁹. A licença atribuída ao projeto é a GNU-GPL v3.0 ([Free Software Foundation, 2018](#)), a qual habilita o usuário a utilizar livremente o software além de realizar e compartilhar alterações de

⁹ <<https://github.com/helioaz/ontPerception>>

acordo com suas necessidades.

VERIFICAÇÃO & VALIDAÇÃO

Uma vez concluído o desenvolvimento da ontologia é necessária sua [verificação](#) e [validação](#) para garantir que os resultados almejados foram alcançados e, eventualmente, realizar correções pontuais nos elementos definidos.

A verificação¹ é realizada pelo raciocinador Pellet ([STARDOG-UNION, 2016](#)) para analisar consistência, processar hierarquia e detalhar inferências realizadas. Esse raciocinador é um *plugin* do editor Protégé, estando continuamente ativo durante o desenvolvimento da ontologia.

A validação² é realizada pelo exercício da ontologia em um ambiente controlado aderente à arquitetura CMDE. Mais especificamente, dado um cenário de uso, é necessário certificar que a “ontPercept” oferece recursos para representar a troca de informações entre os sistemas cognitivo e robótico. O ambiente controlado deve prover recursos para viabilizar experimentos associados à robótica social envolvendo, pelo menos, um agente robótico e um humano atuando em [ambientes não estruturados](#). O agente robótico deve interagir com o ser humano, observando e refletindo reações cognitivas apropriadas à comunicação estabelecida. Três abordagens podem ser utilizadas para implementar esse ambiente:

- Validação com *drivers e stubs*: associada aos testes de integração, esta estratégia define o diálogo entre os agentes envolvidos como atos de uma peça teatral. Cada ato gera textualmente as triplas [RDF](#) necessárias para refletir a percepção do ambiente de forma aderentes à ontologia *ontPercept*. A arquitetura cognitiva recebe essa informação, processa e atua no “ambiente” pelo envio de comandos com o uso de sockets. Um *driver* implementa a comunicação via sockets e gera respostas aos comandos. Esta abordagem permite acompanhar a atualização das informações sensoriais na [triple store](#) Fuseki, além de validar as [APIs](#) de acesso desenvolvidas.

¹ Construimos o produto corretamente?

² Construimos o produto certo?

- Validação Simulada: utiliza um simulador, aderente à ontologia *OntPercept*, capaz de modelar a interação de um agente robótico e um ser humano presentes em um ambiente simulado. Naturalmente, além do simulador, também é necessária a presença da arquitetura cognitiva para receber as informações sensoriais e estabelecer o comportamento do agente robótico.
- Validação Emulada: construir um ambiente controlado envolvendo um agente robótico e um ser humano em um cenário real. Os nós de processamento presentes no agente robótico implementam a captura de informações sensoriais, seu processamento e análise inicial para tomada de decisões realizada pelo sistema cognitivo com o apoio da ontologia *OntPercept*.

Essas abordagens não são excludentes. Na verdade, representam o caminho natural na evolução da validação de uma pesquisa no sentido de sua aplicação efetiva. A validação com *drivers* e *stubs* foi efetivada com o apoio de interface disponibilizada pelo *triple store* Fuseki para envio de comandos SPARQL, por meio de um *browser*. O acesso ao Fuseki é realizado através da página web (Figura 22), na qual informações da percepção do ambiente são enviadas para *triple store* Fuseki utilizando a linguagem SPARQL.

Apesar de ser uma etapa importante na integração de sistemas, *drivers* e *stubs* não validam a dinâmica presente na interação humano-robô. Nas próximas seções, são exploradas a utilização do simulador RHS e do sistema cognitivo para efetivamente exercitar a arquitetura CMDE e, por conseguinte, a ontologia *OntPercept*.

5.1 Simulador *Robot House Simulator* (RHS)

Os simuladores robóticos clássicos não validam aspectos cognitivos presentes no sistema robótico. Longe de ser uma deficiência, esse requisito não é atendido simplesmente pelo fato dos simuladores tradicionais terem por objetivo serem fidedignos a determinado modelo de robô, considerando o comportamento físico de cada sensor e atuador. Nesse contexto, foram estabelecidos os requisitos de um novo simulador, denominado *Robot House Simulator* (RHS), voltado para robótica social (BELO, 2018b; BELO JOSÉ PEDRO RIBEIRO; AZEVEDO, 2017; BELO; ROMERO; AZEVEDO, 2017). Esse simulador foi desenvolvido no contexto de dissertação de mestrado sob orientação da mesma orientadora desta tese com minha coorientação. O objetivo final é viabilizar a validação da arquitetura CMDE e da ontologia *OntPercept*, minimizando os aspectos particulares associados à movimentação física do agente robótico para priorizar os aspectos cognitivos inerentes à robótica social. O RHS possui os seguintes requisitos:

- interface aderente à ontologia *OntPercept*: a utilização de uma ontologia padroniza a percepção do ambiente simplificando a comunicação com o sistema cognitivo;
- operações de alto nível de cognição: uma sintaxe de ações e respostas de alto nível é utilizada para acompanhar as atividades do robô. Exemplos de ações são: falar, pegar,

Figura 22 – Página de interface da *triple store* Fuseki

Fonte: Elaborada pelo autor.

mover, ativar, etc;

- encarnar um personagem: o usuário pode assumir o papel do personagem humano presente no ambiente alterando seu comportamento. O sistema cognitivo deve contornar situações inusitadas geradas por essa intervenção. Essa estratégia evita que o sistema “vicie” em um dado cenário de teste.

Nesta seção será apresentado o estágio atual do simulador RHS, parte constituinte da CMDE, desenvolvido em paralelo a esta tese (BELO, 2018b). Suas principais funcionalidades são: integração com a arquitetura cognitiva, suporte a novos comandos, utilização dos cinco sentidos básicos dos seres humanos (visão, audição, tato, paladar e olfato) e representação de emoções. Na Figura 23, é apresentada uma visão geral do simulador RHS com seus principais módulos e interfaces com sistemas externos. O sistema *Cognitive System* é responsável por tomar decisões e comandar o humanoide virtual. Os dados sensoriais gerados pelo humanoide

Figura 23 – Módulos do simulador RHS

Fonte: Adaptada de [Belo \(2018b, p. 54\)](#).

são padronizados utilizando a ontologia *OntPercept* e armazenados em uma base de dados identificada como *Sense Database*. O módulo *Communication Interface* é responsável por realizar a comunicação com a arquitetura cognitiva e atualizar a *triple store*.

O módulo *Simulator Core* é responsável por gerenciar as principais funcionalidades do simulador, tais como, processar os comandos recebidos da arquitetura e atuar sobre o humanoide de forma apropriada (*Humanoid Robot Control*). Por fim, o módulo *Human-Computer Interface* permite que o usuário comande o avatar humano (*Human Avatar Control*), via teclado, mouse e interface gráfica.

5.1.1 Características do RHS

O simulador possui as seguintes características:

- comandos em alto nível de abstração: a abordagem do simulador é que o desenvolvedor de sistemas cognitivos não necessite configurar sensores e atuadores diretamente para a execução de tarefas. Para isto, o simulador implementa uma ação complexa como uma sequência de comandos mais simples para o agente robótico;
- Informações sensoriais em alto nível: a utilização da ontologia *OntPercept* tem como característica modelar informações sensoriais dos cinco sentidos em alto nível, não exigindo

que a arquitetura cognitiva refine este dados para seu processamento;

- utilização de *game engines* para o desenvolvimento: o simulador tem como base a *game engine* Unity (Unity Technologies, 2018). As *game engines* oferecem diversas bibliotecas de renderização, de comportamento físico, modelamento de ambientes e agentes, etc. Estes recursos aceleram o desenvolvimento e evolução do simulador;
- Log de Dados: o simulador oferece ao usuário um *log* com informações relacionadas à simulação. Estes dados dizem respeito a comandos recebidos, erros ocorridos, estado atual de execução de comandos, etc;
- aderente ao conceito de código aberto: o simulador é distribuído livremente sob a licença GNU GPL v3.0 (Free Software Foundation, 2018) no repositório GitHub (BELO, 2018a).
- Aderente ao ROS: no repositório GitHub também está disponibilizado um *plugin* que viabiliza o envio de comandos ao simulador via mensagens ROS. (ROS-TEAM, 2015).

5.1.2 Comandos de atuação sobre o humanoide

Na [Quadro 13](#), é mostrado o conjunto de comandos disponíveis para que a arquitetura cognitiva possa atuar no humanoide presente no simulador. Note que, existem comandos que determinam qual mão o robô deve usar, sendo assim, a arquitetura cognitiva é capaz de manipular objetos com mãos diferentes.

Atualmente, os comandos mapeiam tarefas que permitem o robô abrir/ligar ou fechar/desligar objetos do ambiente (*ActivateLeft*, *ActivateRight*, *DeactivateLeft*, *DeactivateRight*), pegar e soltar objetos (*TakeLeft*, *TakeRight*, *LeaveLeft*, *LeaveRight*), manipular o foco da visão do robô (*LookAt*, *LookFor*, *HeadReset*), locomovê-lo (*Move*, *Rotate*, *Turn*), saborear ou cheirar (*Taste*, *Smell*), comunicar com o agente humano (*Speech*) ou cancelar a lista de comandos ainda não executados (*CancelCommands*). foram definidos os requisitos do

Naturalmente, cada comando necessita da definição de alguns parâmetros para sua execução. O comando *Move*, por exemplo, permite a locomoção do robô até um objeto (Id) ou posição. No comando *Speech* é passado um texto, informando a frase a ser dita pelo robô.

Durante sua execução, um comando passa por vários estados, podendo assumir os seguintes valores: *Running*, *Success* ou *Fail*. O primeiro indica um comando em execução ou aguardando na fila para ser executado. O segundo indica se o comando teve êxito. Já o terceiro, *Fail*, indica que não foi possível executar o comando, seja porque o objeto não estava ao alcance ou a “mão” solicitada para a tarefa estava ocupada. Ao final da execução, a arquitetura recebe uma indicação se o comando foi executado com sucesso ou falha. Note que o motivo da falha não é informado à arquitetura. Desta forma, ela deve utilizar os dados provenientes dos canais sensoriais para inferir a causa do problema. Caso ocorra uma falha sistêmica, o usuário poderá acessar o *log* de dados para compreender a causa da falha.

Quadro 13 – Comandos disponibilizados pelo RHS

Comando	Parâmetros	Descrição
<i>ActivateLeft</i>	ID	Liga (abre) determinado elemento do ambiente com a mão esquerda.
<i>ActivateRight</i>	ID	Liga (abre) determinado elemento do ambiente com a mão direita.
<i>CancelCommands</i>	Sem parâmetros.	Cancela todos os comandos enviados com conclusão pendente.
<i>DeactivateLeft</i>	ID	Desliga (fecha) determinado elemento do ambiente com a mão esquerda.
<i>DeactivateRight</i>	ID	Desliga (fecha) determinado elemento do ambiente com a mão direita.
<i>HeadReset</i>	Sem parâmetros	Reinicia o foco da visão do robô.
<i>LeaveLeft</i>	ID ou Posição	Deixa objeto da mão esquerda em determinada posição.
<i>LeaveRight</i>	ID ou Posição	Deixa objeto da mão direita em determinada posição.
<i>LookAt</i>	ID ou Posição	Direciona o foco da visão do robô para determinada posição ou objeto.
<i>LookFor</i>	Texto	Busca objeto específico no ambiente através da visão.
<i>Move</i>	ID ou Posição	Movimenta o robô até uma posição ou objeto.
<i>Rotate</i>	Ângulo (em graus)	Rotaciona o robô dado um ângulo.
<i>SmellLeft</i>	Sem Parâmetros	Sente o odor do objeto da mão esquerda.
<i>SmellRight</i>	Sem Parâmetros	Sente o odor do objeto da mão direita.
<i>Speech</i>	Texto	Envia uma mensagem que o robô deve transmitir através da fala.
<i>TakeLeft</i>	ID	Pega objeto com a mão esquerda.
<i>TakeRight</i>	ID	Pega objeto com a mão direita.
<i>TasteLeft</i>	Sem Parâmetros	Prova o sabor do objeto contido na mão esquerda.
<i>TasteRight</i>	Sem Parâmetros	Prova o sabor do objeto contido na mão direita.
<i>Turn</i>	ID ou Posição	Rotaciona o robô para que ele encare determinado objeto ou posição.

Fonte: [Belo \(2018b, p. 51\)](#).

5.1.3 Elementos do ambiente de simulação

Atualmente, o ambiente de simulação é composto por uma sala de estar e uma cozinha. A sala é composta de sofá, mesa, estantes e alguns objetos, como livros, caixa, etc. A cozinha é composta de geladeira, fogão, pia com torneira, mesa, panelas, copos, caneca, pacote de bolachas, pacote de carne e de salmão, etc. ([Figura 24](#)). Alguns objetos são passíveis de interação como por exemplo: transportados para outras posições; portas, gavetas e torneiras, podem ser abertas ou fechadas e interruptores podem ser acionados com o intuito de acender ou apagar luzes. Os objetos podem ainda possuir características particulares, tais como, uma torneira aberta emite o

Figura 24 – Visão geral do ambiente disponibilizado pelo RHS

Fonte: Elaborada pelo autor.

som de água fluindo ou alimentos emitem partículas de odor e têm propriedades de sabor. Estas características enriquecem a simulação, fornecendo ao sistema cognitivo informações adicionais do ambiente que permitem ao agente robótico interações similares as de um ser humano.

Além dos objetos, o RHS também disponibiliza dois agentes: um humano e um humanoide apresentados na [Figura 25](#). O humanoide (denominado Kyle) e o avatar humano (denominada Mariana) são dotados de um “corpo”, com capacidade de interação com o ambiente. Com o auxílio de funções trigonométricas e de técnicas de *Inverse Kinematics* (IK), os agentes são capazes de movimentar braços, agachar e inclinar o corpo, facilitando a interação com objetos. Além disto, com o uso da classe *NavMesh* da ferramenta Unity, é possível a movimentação dos agentes pelo ambiente.

Apesar do robô e do avatar humano serem bastante semelhantes estruturalmente, eles apresentam algumas particularidades. Primeiramente, o robô é controlado pela arquitetura cognitiva, já o avatar humano é controlado pelo usuário. Outra diferença é que o robô é dotado de cinco sentidos e o avatar humano, de emoções. Para esta última característica, são utilizadas as seis emoções básicas definidas em (EKMAN; FRIESEN, 1971), *Happiness*, *Sadness*, *Anger*, *Surprise*, *Disgust* e *Fear*. Contudo, caso o humano não apresente alguma emoção é utilizada a classe *NoEmotion*.

O RHS possui três modos de interação: robô, “avatar humano” e o *god mode*. O modo de interação robô permite que o usuário acompanhe o humanoide Kyle com uma câmera e

Figura 25 – Avatar humano e agente robótico

Fonte: [Belo \(2018b, p. 59\)](#).

visualize, em janelas apresentadas na tela principal, as informações capturadas de percepção e os comandos recebidos da arquitetura cognitiva. O modo de interação “avatar humano” possibilita ao usuário comandar o avatar Mariana, com o auxílio de uma câmara própria, para realizar ações de: movimento, pegar e liberar objetos, construir sentenças verbais e expressar emoções.

O modo de interação *god mode* permite que o usuário interaja com o simulador, com o uso de uma câmera independente dos agentes, que viabiliza o acompanhamento da simulação de diversos ângulos. Além disto, neste modo o usuário é capaz de alterar estados, reposicionar e visualizar informações dos objetos.

5.1.4 Percepção do ambiente

Como se trata de um ambiente simulado, é trivial a obtenção de informações sem utilizar custosas técnicas de processamento e extração de características físicas dos objetos. Os objetos no RHS possuem propriedades que são capturadas pelos sensores e modelados de forma aderente à ontologia *OntPercept*. Nesta seção, são apresentadas particularidades da percepção do ambiente realizadas na implementação do RHS³.

Através da visão, o robô é capaz de abstrair um conjunto de características do objeto, tais como o identificador, o tipo do objeto (por exemplo, *human*), o nome (por exemplo, Mariana), a posição no ambiente, o material, a cor, o estado (aberto, fechado, aceso, apagado), a emoção caso objeto seja um humano e o URI. Este último armazena uma referência para que a arquitetura tenha acesso às informações adicionais do objeto na [Web Semântica](#).

Dada a audição, o robô tem acesso a propriedades de posição, descrição do som, volume e seu tipo. O tipo de som determina, por exemplo, o som de água fluindo pela torneira. A

³ características detalhadas dos sentidos foram detalhadas na [Seção 4.2](#)

propriedade “descrição” diz respeito à informação que o som está transmitindo; no caso de um avatar humano este campo fornece a sentença verbalizada. Caso o objeto gerador do som seja visível pelo robô, é enviado à arquitetura cognitiva um identificador para que a mesma associe o som ao objeto.

Os sensores de tato do robô estão dispostos em suas mãos e deve haver contato com algum objeto para que a captura de informações possa ocorrer. Além da posição, estas informações dizem respeito a *temperature*, *pressure*, *roughness*, *moisture* e *hardness*. Com exceção da *temperature*, dada em Célsius, todas as propriedades recebem valores parametrizados entre 0 e 1.

O olfato do robô é estimulado quando uma partícula, emitida por um objeto, entra em contato com este sensor. O olfato do robô é capaz de detectar a posição do objeto emissor e o tipo de odor. Foi adotada a classificação proposta em (CASTRO; RAMANATHAN; CHEN-NUBHOTLA, 2013), onde o odor pode ser do tipo *Fragrant*, *Woody*, *Fruity*, *Chemical*, *Decayed*, *Minty*, *Sweet*, *Popcorn*, *Pungent* ou *Lemon*. Duas formas de identificação da origem do odor são fornecidas. Na primeira somente uma posição aproximada da fonte é fornecida. Na segunda, o robô deve pegar o objeto gerador e leva-lo até a altura do sensor olfativo, presente na face, com o uso do comando *Smell*.

Finalmente, o paladar permite reconhecer o sabor dos objetos. Este estímulo somente ocorre quando o comando *Taste* é executado pelo humanoide. Desta forma, tem-se informações relacionadas aos sabores básicos apresentados em (HUANG *et al.*, 2006), sendo eles a *sweetness*, *sourness*, *sattness*, *bitterness* e *umami*. Estes valores também são parametrizados entre 0 e 1.

5.2 Modelando arquitetura cognitiva

O elemento responsável pelo comportamento do agente robótico é o sistema cognitivo, que nesta validação é instanciado através da arquitetura cognitiva Soar, apresentada na Seção 3.3.

A arquitetura Soar disponibiliza duas ferramentas que auxiliam a elaboração de comportamentos pela arquitetura cognitiva (LAIRD *et al.*, 2017). A primeira, denominada “Soar-JavaDebugger”, permite o acesso a variáveis locais, o carregamento das produções, além do controle sobre sua execução. A segunda ferramenta, denominada “VisualSoar”, disponibiliza uma interface para edição das produções e do “mapa de dados” (*DataMap*), que estabelece uma hierarquia de variáveis para a interface utilizada durante a execução das produções. A Figura 26 apresenta a tela principal do editor “VisualSoar”, sendo que, na esquerda, uma *tree view* pode ser visualizada com as produções e, na direita, uma janela de edição da produção selecionada é mostrada.

O “mapa de dados” representa a interface do arcabouço Soar com o ambiente associado ao problema sendo modelado. A partir da definição desse mapa é possível a construção das produções que, em última instância, representam o comportamento desejado. Para a arquitetura

Figura 26 – Tela da ferramenta “VisualSoar”

CMDE, essa interface modela os comandos que são disponibilizados pelo simulador RHS e a percepção do ambiente é modelada pela ontologia *OntPercept*. Na Figura 27, é apresentado o “mapa de dados” elaborado, onde a interface de comandos é expandida através do *output-link* e a percepção do ambiente mapeada a partir do nó *input-link*. Os sentidos presentes no *input-link* são detalhados na Figura 28, apresentando seus atributos e tipos de dados.

Um importante elemento na utilização da Soar está associado à gerência da interface com o ambiente. A interface *Soar Markup Language* (SML) (SOAR-GROUP, 2014) representa a abordagem utilizada pela Soar para permitir que agentes se comuniquem com o ambiente através do envio e recepção de pacotes XML.

No contexto da CMDE, foi desenvolvido o programa “CogArc” que instancia a arquitetura cognitiva utilizando o Soar e sua interface SML. O diagrama de classes em UML, presente na Figura 29, representa as classes criadas para modelar essa implementação⁴, sendo o modelo completo disponibilizado no repositório GitHub⁵. A classe principal “CogArc” possui relacionamento com: a API “ontPerceptJava” para acesso a *triple store* Fuseki, com a classe “CmdSocket” para envio de comandos e recepção de respostas do simulador RHS, a classe “Soar”, que representa o arcabouço Soar e, finalmente com “SimpleConsole” que estabelece uma interface básica com

⁴ Um diagrama mais detalhado, com atributos e operações, é apresentado na Seção A.2

⁵ <<https://github.com/helioaz/ontPercept>>

Figura 27 – Mapa de dados do Soar

Fonte: Elaborada pelo autor.

Figura 28 – Detalhe dos atributos presentes na percepção do ambiente

Fonte: Elaborada pelo autor.

o usuário. O programa “CogArc” totaliza 1685 linhas de código, sendo baseado no programa “SimpleAsyncEnv.java” fornecido como exemplo pela equipe da Soar ([SOAR-GROUP, 2014](#)).

Figura 29 – Diagrama de classe do programa “CogArc”

Fonte: Elaborada pelo autor.

5.3 Descrição do experimento

O desenvolvimento do simulador RHS e do programa “CogArc” permitiram a elaboração de experimentos para validar a arquitetura CMDE. Os passos para definir um experimento envolvem: definição do comportamento desejado para o agente robótico, definição das regras de produção Soar que implementam o comportamento e, finalmente, exercitar os resultados no simulador RHS.

O experimento realizado consiste na solicitação: “Pegar pacote bolacha”, ao humanoide por parte do agente humano. Ambos agentes se encontram na sala de estar e o pacote de biscoito se encontra sobre a pia da cozinha. O experimento também tem como objetivo demonstrar o efeito da emoção humana no comportamento do humanoide, sob controle da arquitetura cognitiva. Desta forma, o agente humano é exercitado com dois estados de emoção: neutra (*noEmotion*) e com raiva (*Anger*). Conforme apresentado na Seção 5.1, o controle do agente humano é realizado pelo usuário de sistema, com o uso de teclado, mouse e de uma janela de interface que viabiliza a inserção de comandos e definição das emoções.

A execução do arcabouço Soar ocorre por meio de mudanças de estado disparadas pela execução das regras de produções (LAIRD, 2017). Na Quadro 14, são apresentadas as regras elaboradas para modelar o comportamento de “buscar um pacote de biscoito”. Cada regra demanda, em geral, a construção de três produções: *Propose*, *Apply* e *Remove*. O corpo das produções de proposição de operadores (*Propose*) é apresentado na Seção A.4⁶.

Quadro 14 – Regras de produção Soar

Regra	Descrição
<i>initialize-butler</i>	estabelece o estado inicial e aguarda comandos.
<i>cmdReception</i>	analisa comando recebido.
<i>getKitchenCracker</i>	movimenta androide até a cozinha.
<i>lookForCracker</i>	procura pacote de biscoito no ambiente.
<i>moveToCracker</i>	movimenta androide até pacote de biscoito.
<i>takeCracker</i>	pega pacote de biscoito com a mão direita.
<i>headReset</i>	reposiciona a cabeça do androide.
<i>moveLivingRoom</i>	movimenta androide até a sala de estar.
<i>lookForHuman</i>	procura agente humano no ambiente.
<i>moveToHuman</i>	movimenta androide até a posição do humano.

Fonte: Elaborada pelo autor.

O exercício das regras de produção Soar ocorre através de ações ativadas pelo usuário do simulador RHS. O experimento se inicia quando o avatar humano, com emoção neutra, verbaliza “get cracker” via interface de comando do usuário. Esta informação é imediatamente enviada a Soar, que analisa a sentença e envia comandos apropriados para atender o pedido. Os comandos

⁶ Devido a limitação de espaço, somente produções *Propose* são apresentados neste documento. O conjunto completo pode ser acessado em <<https://github.com/helioaz/ontPerception>> .

recebidos pelo simulador RHS são: *go to Kitchen*; *look for Cracker*; *move to Cracker*; *grab Cracker*; *move to Living Room*; *look for “Mariana”*; *move to Mariana* e *speak “Mariana, take the cracker”*. Finalmente, o usuário novamente controla o avatar humano para pegar o pacote de bolachas da mão do humanoide. Conforme apresentado na [Seção 5.1](#), o controle do avatar humano é realizado pelo módulo *Human-Computer Interface* (Figura 23) utilizando teclado e mouse. Na Figura 30, é apresentado o diagrama de sequência dessa interação do simulador RHS

Figura 30 – Diagrama de sequência para a execução do pedido: “get cracker”

Fonte: Elaborada pelo autor.

(representado pelos atores “Robô” e “Humano”) com a arquitetura cognitiva (ator “Arquitetura”). Este diagrama tem como objetivo elucidar a relação do simulador com dois componentes da CMDE: simulador RHS e arquitetura cognitiva. Os focos de controle, realçados em vermelho, presentes na linha da vida do ator “Robô” representam o processamento associado a percepção do ambiente. Por simplicidade, somente um sentido é representado por vez, contudo, na realidade

todas as alterações de ambiente ocorridas em um intervalo de tempo são enviadas para a [triple store](#) simultaneamente.

Ainda neste experimento, no final da sequência de ações (devido a emoção neutra do avatar humano), o humanoide solicita que o pacote de bolachas seja recebido de forma neutra. Na segunda parte do experimento, ocorre o mesmo pedido “*get cracker*”, porém a emoção expressa pelo avatar humano é de raiva (*anger*). Nesse contexto, o humanoide declara o final da ação com uma frase mais amigável: “*Adorable Princess Mariana, please accept this cookie package with love...*”. Este experimento, com as duas configurações de emoção, está disponibilizado na plataforma de compartilhamento de vídeo YouTube ([CMDE-GROUP, 2014](#)).

Na [Figura 31](#), é apresentada a tela do RHS durante a execução do experimento. Nessa tela temos a visão geral do ambiente sendo sobreposta por quatro janelas. Na janela no canto superior direito é apresentada a visão do ambiente sob a perspectiva do humanoide. Na janela do canto inferior direito é apresentada abas contendo as informações de percepção do ambiente, sendo apresentado o sentido de visão. Na janela superior central é permitido pausar, ativar e controlar a velocidade de execução do simulador. Finalmente, no canto inferior direito, um conjunto de três abas é disponibilizado, uma para cada modo de operações de comandos: *ROBOT*, *HUMAN* e *GOD VIEW*.

Figura 31 – Tela do simulador durante a execução do experimento

Fonte: Elaborada pelo autor.

A configuração física desse experimento envolve dois sistemas de processamento independentes conectados por um roteador privado com velocidade de 100Mbps. Essa configuração é aderente à arquitetura CMDE presente na [Figura 1](#) e reproduzida na [Figura 32](#) para conveniência

do leitor. O nó “Sistema Cognitivo” implementa o arcabouço Soar e a *triple store* Fuseki em um ambiente Linux (Ubuntu 16.4 LTS). O nó Simulador Robótico é instanciado em um ambiente Windows (Windows 10).

Figura 32 – Cognitive Model Development Environment - CMDE

Fonte: Elaborada pelo autor.

Consoante ao apresentado na [Seção 5.2](#), o arcabouço Soar é gerenciado pelo programa “CogArc” ativado através de um console conforme apresentado no [Quadro 15](#). Nessa quadro, a ativação do “CogArc” com dois parâmetros são mostrados: o IP 192.168.0.199 e a porta 6321 utilizados para acessar o *socket* de envio de comandos ao RHS. Após a ativação, o programa continuamente apresenta mensagens de informações associadas ao progresso na execução das regras presentes no Soar.

5.4 Considerações finais

Neste capítulo, foi apresentada a validação da ontologia *OntPercept*, parte integrante da arquitetura CMDE, com a realização de um experimento simulado envolvendo um humanoide e um ser humano.

Para concretizar o experimento dois componentes foram utilizados. O primeiro, envolveu o desenvolvimento do Simulador RHS para modelar um ambiente composto por: sala de estar, cozinha, um agente robótico e um humano.

O segundo componente envolveu a utilização da arquitetura cognitiva Soar para exercitar o comportamento desejado para o agente robótico. A tarefa mapeada na Soar foi “Pegar pacote bolacha”, que compreende a recepção e execução de comando pelo humanoide a partir de interação verbal com o ser humano. Nesse experimento, a finalização da tarefa leva em consideração o estado emocional do humano.

Quadro 15 – Ativação do programa “CogArc” via terminal Linux

```
hazevedo@Inspiron5457: /Implementacao/Soar$ ./execSoarUSP.sh CogArc 192.168.0.199 6321
Number of args = 2
Using host= 192.168.0.199 port = 6321
Starting Socket. Warning: do not forget to unblock the windows firewall
Warning: If you are planning use the debugger then do not activate using START or RUN commands.
Instead, start SoarDebugger, in the sequence, press Remote and Step buttons.
> run
Soar interface: Starting Soar with OntPercept.
Soar interface: Response received: e64fec04-da12-429d-8c5f-1d5c26c17b4a<|>1<|>
Soar interface: Input Vision: Sofa O0000000002
Soar interface: Input Vision: Wall O0000000022
Soar interface: Input Vision: Human O0000000016
Soar interface: Input Vision: Book_Shelf O0000000019
Soar interface: Input Vision: Door O0000000017
Soar interface: Input Vision: Ceiling O0000000014
Soar interface: Input Vision: Floor O0000000020
Soar interface: Input Vision: Lamp O0000000011
...
```

Fonte: Elaborada pelo autor.

Neste momento é importante ressaltar que a finalidade deste modelamento foi exercitar os componentes associados a definição da ontologia *OntPercept*. Assim, somente a tarefa “Pegar pacote bolacha” foi modelada na Soar. Essa tarefa não demandou a utilização de funcionalidades adicionais da Soar como: memória semântica e episódica ou aprendizagem por reforço e episódica. Naturalmente, a estrutura de regras criadas permite agregar outros comportamentos como: fechar torneira ao ouvir o som de água fluindo ou guardar o pacote de salmão na geladeira ao identificar seu odor no ambiente.

A realização do experimento revelou a potencialidade desta tese ao criar um ambiente isolado para testar rapidamente o comportamento do agente robótico, atuando em interação direta com seres humanos em [ambientes não estruturados](#), ou seja, robótica social.

Naturalmente, o passo seguinte para uma validação completa da arquitetura CMDE envolve a utilização de robô e humano reais, atuando em ambiente controlado. Essa classe de validação está fora do contexto desta tese, sendo considerada uma linha de pesquisa futura, conforme detalhado na [Seção 6.1](#).

CONCLUSÃO

Atualmente, arquiteturas cognitivas aplicadas na área de robótica interagem diretamente nos sensores e atuadores presentes no robô. Conforme detalhado no [Capítulo 1](#), essa abordagem possui algumas desvantagens que acabam por postergar a obtenção dos resultados. Como forma de contribuir nesse processo, foram propostas nesta tese, uma arquitetura, CMDE, um simulador para robótica social, RHS e, finalmente, uma ontologia, *OntPercept*. Todos estes elementos são utilizados em conjunto para projetar, desenvolver e validar sistemas voltados à robótica social.

A arquitetura *Cognitive Model Development Environment* (CMDE) divide a atuação do agente robótico em dois nós de processamento: o cognitivo e o robótico. O nó cognitivo é voltado ao processamento de comportamentos e o nó robótico realiza operações de “baixo nível” necessárias para o controle de sensores e atuadores presentes no agente robótico. A conexão entre estes dois nós é realizada com o apoio da ontologia *OntPercept* especificamente definida para esse propósito. Essa separação simplifica o equacionamento do problema, além de, facilitar a substituição dos nós de processamento, seja para testar uma nova arquitetura cognitiva ou para migrar de um sistema simulado para um robô real. A CMDE também simplifica a utilização de arquiteturas cognitivas em robótica social ao permitir que o projetista do sistema cognitivo reduza sua preocupação com detalhes de processamento presentes nos sistemas robóticos e direcione sua atenção para modelar a captura, processamento e armazenamento de conhecimento.

Para validar a arquitetura ocorreu minha coorientação no desenvolvimento do *Robot House Simulator* (RHS), que disponibiliza um ambiente residencial onde temos dois agentes: um humano e um androide. A opção de desenvolver um simulador, ao invés de utilizar um de "prateleira", foi adotada devido à característica única desta pesquisa que envolve a utilização de todos os sentidos para percepção do ambiente. O domínio do software associado ao simulador RHS facilita, por exemplo, a criação de novos sensores ou algoritmos para coleta de informações sensoriais.

A ontologia proposta nesta tese, *OntPercept*, enfatiza a necessidade de formalização

da informação sensorial que transita entre os nós de processamento de forma semelhante ao proposto por Jamone et al. (JAMONE *et al.*, 2016): a proposta não é criar uma representação detalhada do ambiente, mas sim, compreender qual é a informação mínima que deve ser extraída para permitir que o robô realize suas tarefas com êxito.

Três classes de informação sensorial disponibilizadas pela *OntPercept* merecem destaque por seu caráter inédito. Os conceitos associados ao olfato, paladar e emoções viabilizam sua utilização pela arquitetura cognitiva, mesmo antes que sua captura e processamento tenham sido consolidadas como um produto comercialmente acessível para uso em agentes robóticos. Em particular, realçamos a abordagem utilizada para modelar o sentido de olfato que utiliza uma representação vetorial para ponderar os dez odores básicos definidos por Castro, Ramanathan e Chennubhotla (2013).

A utilização desta ontologia cria uma taxionomia nas informações que simplifica a definição de comportamentos na arquitetura cognitiva. Além disso, apesar de não ter sido explorada nesta tese, o uso de raciocinador acoplado a *triple store* viabiliza a geração dinâmica de novos conceitos a partir de inferências realizadas sobre conceitos previamente estabelecidos.

Uma característica interessante associado às ontologias é facilitar o acesso às informações presentes na Web. No experimento apresentado no Capítulo 5, cada objeto possui a propriedade *associateURI*, que associa sua entrada à *DBpedia* (DBPEDIA-TEAM, 2018). A partir desta entrada é possível, com o uso da tecnologia da Web Semântica, acessar informações como composição, fabricante, preço, origem, dentre outros. Essas informações podem ser utilizadas pela arquitetura cognitiva para ampliar seu conhecimento sobre o ambiente.

No mundo real, o ambiente não é fornecido com uma configuração prévia de entradas nas bases da Web. Uma estratégia é suprir essa informação com código de barras ou utilizar APIs de análise de imagens como o *Google Cloud Vision* (GOOGLE-GROUP, 2018) para reconhecer o objeto. Vale ressaltar que a tecnologia da Web Semântica não representa uma solução completa e aceita de forma absoluta pelos pesquisadores da área. Como exemplo, o filósofo Smith (2009, vídeo 6), da *University at Buffalo*, apresenta uma lista de problemas associados à utilização da Web Semântica (Apêndice A). Apesar das críticas, trata-se de uma tecnologia em franca expansão sob o cuidado de entidades como a W3C (W3C, 2018), com indiscutível relevância na evolução da Web.

O desenvolvimento da ontologia *OntPercept* surgiu naturalmente da necessidade de um protocolo para comunicação das informações sensoriais capturadas pelo sistema robótico. O uso da IEEE 1872-2015 e da SUMO como ontologias de nível superior acelerou o processo ao fazer uso de conhecimento e habilidades previamente alcançadas na área. Esta abordagem também atua como catalisadora na interação com outros grupos de pesquisa envolvidos na elaboração e utilização de ontologias em aplicações de robótica e automação industrial.

Os resultados alcançados por esta pesquisa oferecem uma maior flexibilidade ao pro-

cesso de elaboração e validação de sistemas para robótica social com a redução do esforço do pesquisador e, viabilização de desenvolvimento em laboratórios menores com recursos reduzidos. O conjunto completo pode facilmente ser utilizado em cursos de HRI, com foco em robótica social, envolvendo o uso de simuladores cognitivos nos experimentos práticos (BERRY, 2015).

Para simplificar a utilização do sistema, foram criadas duas APIs de interface para a *triple store* Fuseki que mantém a informação sensorial. Além das APIs, também foi elaborado o programa “CogArc” que implementa o acesso à arquitetura cognitiva Soar. Todo esse conjunto de software totaliza 6.384 linhas de código nas linguagens Java e C#.

A divulgação dos resultados desta pesquisa ocorre tanto através de artigos científicos, quanto através da disponibilização dos itens de configuração, associados à CMDE e ao simulador RHS, no repositório [GitHub](#)¹. Ao compartilhar os resultados através da licença GNU GPL v3.0 (Free Software Foundation, 2018), espera-se motivar a comunidade a contribuir na evolução da CMDE com a criação de *Issues* no [GitHub](#) e, paralelamente, aumentar a visibilidade do grupo.

6.1 Trabalhos futuros

A arquitetura CMDE representa uma pesquisa em constante evolução e, como tal, catalisadora de propostas futuras, seja para, aprimorar os resultados atuais ou para iniciar novas linhas de atuação, como por exemplo:

- **Aprimorar *OntPercept*:** Um aspecto importante de uma ontologia é viabilizar a evolução constante dos conceitos e relacionamentos com a inclusão de novos elementos na hierarquia sem alterar o modelo base (ZESHAN *et al.*, 2014). Assim, os conceitos atuais devem ser interpretados como resultados alcançados em uma primeira etapa do desenvolvimento que devem ser refinados na medida que a CMDE seja aplicada na prática.
- **Arquitetura Robótica Híbrida:** a arquitetura CMDE adota o paradigma hierárquico/deliberativo, onde o sistema robótico opera em um modelo *top-down* obtendo informações do ambiente (*Sense*) para as fases de planejamento (*Plan*) e atuação (*Act*). Uma interessante linha de pesquisa ocorre ao se utilizar uma abordagem menos rígida, como o paradigma híbrido, que utiliza uma fase de planejamento mesclada com características do paradigma reativo. Essa estratégia viabiliza reações mais rápidas do agente robótico, além de simplificar o uso de *affordances* (JAMONE *et al.*, 2016) para mapear comportamentos já estabelecidos.
- **Ontologia *OntAct* e *OntKnow*** a ontologia proposta é unidirecional, ou seja, explora os sentidos, mas não abrange as ações a serem executadas pelo agente robótico. A elaboração de uma ontologia (*OntAct*) para representar as ações, moldadas às capacidades disponíveis nos robôs, representa um passo importante na consolidação do CMDE. Na mesma linha de

¹ <<https://github.com/helioaz/ontPerception>>

raciocínio, temos a definição de uma ontologia para modelar o conhecimento pré existente (*OntKnow*), que forneceria informações de topologia, operação de eletrodomésticos ou manipulação de animais domésticos.

- **Interfaces para outras arquiteturas:** a validação da ontologia foi realizada utilizando a arquitetura cognitiva Soar (LAIRD *et al.*, 2017). A elaboração de interfaces para outras arquiteturas cognitivas, como MECA (GUDWIN *et al.*, 2018), ACT-R (ACT, 2015) ou Clarion (KOTSERUBA; GONZALEZ; TSOTSOS, 2018), representa um passo importante no estabelecimento de *benchmarks* para comparação do comportamento de arquiteturas cognitivas distintas.
- **Novos sentidos:** acréscimo de outros sensores inspirados na natureza, como exemplo temos, nos mamíferos, o órgão de equilíbrio que transforma a deflexão de milhares de fibras ciliadas, presentes no sistema vestibular do ouvido interno, em sinais elétricos para o cérebro. Outros exemplos são: a visão por infravermelho das cobras ou a leitura do campo magnético da Terra por aves, baleias e tartarugas para orientação durante migração. Naturalmente, temos ainda sensores não encontrados na natureza como os de radiação ou de propriocepção (nível de energia, identificação de falhas).
- **Aprimorar Robot House Simulator (RHS):** como exemplo de melhorias temos: inclusão de agentes humanos especializados (crianças e idosos); utilização de interfaces com realidade aumentada; definição de novos ambientes, prédios e ruas.
- **Validação em ambiente real:** a validação da arquitetura CMDE foi realizada com um simulador especificamente projetado para essa finalidade. Naturalmente, o passo seguinte envolve a utilização de robô e humano reais, atuando em ambiente especificamente elaborado para essa interação. Mesmo com as limitações dos sistema robóticos atuais (não possuem paladar ou odor e raramente possuem tato) o uso de um ecossistema real representa um passo importante na maturação desta pesquisa.

6.2 Divulgação dos resultados

Esta tese gerou artigos que foram submetidos e aceitos em conferências importantes da área, apresentados a seguir:

- AZEVEDO, H.; BELO, J. P. R.; ROMERO, R. A. F. . Reducing the gap between cognitive and robotic systems. In: 26th IEEE International Symposium on Robot and Human Interactive Communication - RO-MAN2017, 2017, Lisboa.
- AZEVEDO, H.; BELO, J. P. R.; ROMERO, R. A. F. . Cognitive and Robotic Systems: Speeding up Integration and Results. In: 14rd Latin American Robotics Symposium - LARS-2017, 2017, Curitiba.

- BELO, J. P. R.; AZEVEDO, H. ; ROMERO, R. A. F. . RHS Simulator for Robotic Cognitive Systems. In: 14rd Latin American Robotics Symposium - LARS-2017, 2017, Curitiba.
- ROMERO, R. A. F.; RAMOS, J. J. G.; AZEVEDO, H. . A Proposal of a Graduate Class for the Human Robot Interaction Area. In: 14rd Latin American Robotics Symposium - LARS-2017, 2017, Curitiba.
- BELO, J. P. R.; ROMERO, R. A. F.; AZEVEDO, H. . Simulador para sistemas cognitivos voltado para robótica social. In: XIII Simpósio Brasileiro de Automação Inteligente (SBAI) 2017, 2017, Porto Alegre.
- AZEVEDO, H.; ROMERO, R. A. F. . Interface for Communication Between Robotic and Cognitive Systems Through the Use of a Cognitive Ontology. In: COGNITIVE 2016, The Eighth International Conference on Advanced Cognitive Technologies and Applications, 2016, Roma.

Também foram publicados/submetidos os seguintes artigos em importantes periódicos relacionados com a área de Robótica Social:

- AZEVEDO, H.; BELO, J. P. R. ; ROMERO, R. A. F. . Cognition, Affordance and Robotics: using ontology to formalize the sensory information. In: Integrated Computer-Aided Engineering. ISSN online: 1875-8835; EM REVISÃO.
- TROVATO, GABRIELE; RAMOS, JOSUE G.; AZEVEDO, HELIO; MORONI, ARTEMIS; MAGOSS, SILVIA; SIMMONS, REID; ISHII, HIROYUKI; TAKANISHI, ATSUO. A receptionist robot for Brazilian people: study on interaction involving illiterates. In: Paladyn, Journal of Behavioral Robotics, v. 8, p. 1-17, 2017.

REFERÊNCIAS

ABRAHAMSEN, A.; BECHTEL, W. History and core themes. In: Keith Frankish; RAMSEY, W. M. (Ed.). **The Cambridge Handbook of Cognitive Science**. 1. ed. Cambridge: Cambridge University Press, 2012. cap. 1, p. 9–27. ISBN 9780521691901. Citado nas páginas 63, 64 e 65.

ACT. **ACT-R Home**. 2015. Disponível em: <<http://act-r.psy.cmu.edu/>>. Acesso em: 24 jan. 2016. Citado nas páginas 30, 69 e 116.

AHMED, K. **UserGuide Getting Started**. 2016. Disponível em: <<https://github.com/dotnetrdf/dotnetrdf/wiki/UserGuide-Getting-Started>>. Acesso em: 27 mai 2018. Citado na página 90.

AIRÒ, G. F.; PIANU, D.; CEMPINI, M.; CORTESE, M.; RUSSO, L.; INDACO, M.; NERINO, R.; CHIMIENTI, A.; ODDO, C.; VITIELLO, N. Vision-Based Pose Estimation for Robot-Mediated Hand Telerehabilitation. **Sensors**, v. 16, n. 2, p. 208, feb 2016. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/16/2/208>>. Citado na página 53.

AKOLKAR, H.; VALEIRAS, D. R.; BENOSMAN, R.; BARTOLOZZI, C. Visual-auditory saliency detection using event-driven visual sensors. In: **2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)**. IEEE, 2015. p. 1–6. ISBN 978-1-4673-7888-8. Disponível em: <<http://ieeexplore.ieee.org/document/7300674/>>. Citado na página 47.

ALAMEDA-PINEDA, X.; HORAUD, R. Vision-guided robot hearing. **The International Journal of Robotics Research**, v. 34, n. 4-5, p. 437–456, apr 2015. ISSN 0278-3649. Disponível em: <<http://journals.sagepub.com/doi/10.1177/0278364914548050>>. Citado na página 47.

ANDO, N.; KANZAKI, R. A simple behaviour provides accuracy and flexibility in odour plume tracking - the robotic control of sensory-motor coupling in silkmoths. **Journal of Experimental Biology**, v. 218, n. 23, p. 3845–3854, dec 2015. ISSN 0022-0949. Citado nas páginas 43, 44 e 45.

ANGELE, J.; KIFER, M.; LAUSEN, G. Ontologies in F-Logic. In: STAAB, S.; STUDER, R. (Ed.). **Handbook on Ontologies**. 2. ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. cap. 1, p. 45–70. ISBN 978-3-540-70999-2. Citado na página 32.

ANTONIOU, G.; GROTH, P.; HARMELEN, F. van; HOEKSTRA, R. **A Semantic Web Primer**. 3. ed. Cambridge, Massachusetts: The MIT Press, 2012. 288 p. ISBN 978-0-262-01828-9. Citado nas páginas 33, 138, 139 e 140.

ANTONIOU, G.; HARMELEN, F. van. Web Ontology Language: OWL. In: STAAB, S.; STUDER, R. (Ed.). **Handbook on Ontologies**. 2. ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. cap. 1, p. 91–110. ISBN 978-3-540-70999-2. Citado na página 33.

ARDIANSYAH, R. A. Design of An Electronic Narrator on Assistant Robot for Blind People. **MATEC Web of Conferences**, v. 42, p. 03013, feb 2016. ISSN 2261-236X. Disponível em: <<http://www.matec-conferences.org/10.1051/matecconf/20164203013>>. Citado na página 46.

- ARENA, P.; PATANÉ, L.; STORNANTI, V.; TERMINI, P. S.; ZÄPF, B.; STRAUSS, R. Modeling the insect mushroom bodies: Application to a delayed match-to-sample task. **Neural Networks**, v. 41, p. 202–211, 2013. Citado na página 44.
- ASADNIA, M.; KOTTAPALLI, A. G. P.; KARAVITAKI, K. D.; WARKIANI, M. E.; MIAO, J.; COREY, D. P.; TRIANTAFYLLOU, M. From Biological Cilia to Artificial Flow Sensors: Biomimetic Soft Polymer Nanosensors with High Sensing Performance. **Scientific Reports**, v. 6, n. 1, p. 32955, dec 2016. ISSN 2045-2322. Disponível em: <<http://www.nature.com/articles/srep32955>>. Citado na página 47.
- AUSTIN, M. **Google built an entire fake city to test the AI of its driverless cars**. 2017. Disponível em: <<https://www.digitaltrends.com/cars/google-fake-city/>>. Citado na página 26.
- BAADER, F.; HORROCKS, I.; SATTLER, U. Description Logics. In: STAAB, S.; STUDER, R. (Ed.). **Handbook on Ontologies**. 2. ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. cap. 1, p. 21–44. ISBN 10.1007/978-3-540-92673-3. Citado nas páginas 32 e 84.
- BAI, Y.; SNYDER, J. B.; PESHKIN, M.; MACIVER, M. A. Finding and identifying simple objects underwater with active electrosense. **The International Journal of Robotics Research**, v. 34, n. 10, p. 1255–1277, sep 2015. ISSN 0278-3649. Disponível em: <<http://journals.sagepub.com/doi/10.1177/0278364915569813>>. Citado na página 51.
- BALL, D.; UPCROFT, B.; WYETH, G.; CORKE, P.; ENGLISH, A.; ROSS, P.; PATTEN, T.; FITCH, R.; SUKKARIEH, S.; BATE, A. Vision-based Obstacle Detection and Navigation for an Agricultural Robot. **Journal of Field Robotics**, v. 33, n. 8, p. 1107–1130, dec 2016. ISSN 15564959. Disponível em: <<http://doi.wiley.com/10.1002/rob.21644>>. Citado na página 53.
- BARRETT, T. **FingerTPS system**. 2018. Disponível em: <<http://www.barrett.com/products.htm>>. Acesso em: 27 apr. 2018. Citado na página 50.
- BATEMAN, J. **John Bateman's ontology portal...** 2016. Disponível em: <<http://www.fb10.uni-bremen.de/anglistik/langpro/webpace/jb/info-pages/ontology/ontology-root.htm>>. Acesso em: 11 dez. 2016. Citado nas páginas 33 e 39.
- BAXTER, P.; LEMAIGNAN, S.; TRAFTON, J. G. Cognitive Architectures for social human-robot interaction. In: **2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)**. IEEE, 2016. p. 579–580. ISBN 978-1-4673-8370-7. Disponível em: <<http://ieeexplore.ieee.org/document/7451865/>>. Citado na página 71.
- BAYRO-CORROCHANO, E.; VAZQUEZ-SANTACRUZ, E.; MOYA-SANCHEZ, E.; CASTILLO-MUNIS, E. Geometric Bioinspired Networks for Recognition of 2-D and 3-D Low-Level Structures and Transformations. **IEEE Transactions on Neural Networks and Learning Systems**, v. 27, n. 10, p. 2020–2034, 2016. Citado na página 51.
- BELO, J. P. R. **RHS - Robot House Simulator**. 2018. Disponível em: <<https://github.com/JPedroRBelo/RHS>>. Acesso em: 1 jun 2018. Citado na página 99.
- BELO, J. P. R. **Um simulador para robótica social aplicado a ambientes internos**. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação - ICMC, USP - São Carlos, São Carlos, mar 2018. Citado nas páginas 29, 96, 97, 98, 100 e 102.

BELO, J. P. R.; ROMERO, R. A. F.; AZEVEDO, H. Simulador para sistemas cognitivos voltado para robótica social. In: **XIII Simpósio Brasileiro de Automação Inteligente (SBAI) 2017**. Porto Alegre, Brazil: Sociedade Brasileira de Automática (SBA), 2017. p. 51–56. ISSN 2175 8905. Citado na página 96.

BELO JOSÉ PEDRO RIBEIRO, R. A. F.; AZEVEDO, H. RHS simulator for robotic cognitive systems. In: **2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)**. Curitiba, Brazil: IEEE, 2017. p. 1–6. ISBN 978-1-5386-0956-9. ISBN: 978-1-5090-3656-1. Disponível em: <http://ieeexplore.ieee.org/document/8215306/>. Citado na página 96.

BELTER, D.; ŁABECKI, P.; FANKHAUSER, P.; SIEGWART, R. RGB–D terrain perception and dense mapping for legged robots. **International Journal of Applied Mathematics and Computer Science**, v. 26, n. 1, jan 2016. ISSN 2083-8492. Disponível em: <https://www.degruyter.com/view/j/amcs.2016.26.issue-1/amcs-2016-0006/amcs-2016-0006.xml>. Citado na página 53.

BERNUS, P.; SHAW, M. J.; BŁAŻEWICZ, J.; SCHMIDT, G. **Handbook on Ontologies**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. 654 p. ISSN 10744770. ISBN 978-3-540-70999-2. Citado na página 32.

BERRY, C. A. Teaching a First Course in Human-Robot Interaction. In: **2015 ASEE Annual Conference and Exposition**. Seattle, Washington: ASEE Conferences, 2015. <https://peer.asee.org/24799>. Citado na página 115.

BEVILL, R.; AZZI, P.; SPADAFORA, M.; PARK, C. H.; KIM, H. J.; LEE, J.; RAIHAN, K.; JEON, M.; HOWARD, A. M. Multisensory robotic therapy to promote natural emotional interaction for children with ASD. In: **2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)**. IEEE, 2016. p. 571–571. ISBN 978-1-4673-8370-7. Disponível em: <http://ieeexplore.ieee.org/document/7451861/>. Citado na página 53.

BICA, S. **MAPPED Repository: Comparative Database of Biologically Inspired Cognitive Architectures**. 2012. Disponível em: <http://bicasociety.org/mapped/architectures/>. Acesso em: 12 may 2018. Citado na página 67.

BILLARD, A.; BONFIGLIO, A.; CANNATA, G.; COSSEDDU, P.; DAHL, T.; DAUTENHAHN, K.; MASTROGIOVANNI, F.; METTA, G.; NATALE, L.; ROBINS, B.; SEMINARA, L.; VALLE, M. The ROBOSKIN Project: Challenges and Results. In: . [s.n.], 2013. p. 351–358. Disponível em: http://www.springerlink.com/index/10.1007/978-3-7091-1379-0_43. Citado nas páginas 49 e 50.

BIOPORTAL. **Browse the library of ontologies**. 2016. Disponível em: <https://biportal.bioontology.org/ontologies>. Acesso em: 10 dez. 2016. Citado nas páginas 33 e 38.

BODENHAUSEN, G. V.; HUGENBERGE, K. Attention, perception, and social cognition. In: STRACK, F.; FÖRSTER, J. (Ed.). **Social cognition: The basis of human interaction**. New York, NY: Psychology Press, 2009. cap. 1, p. 1–22. ISBN 978-1-84169-451-1. Citado na página 64.

BRUNETTO, N.; SALT, S.; FIORAIO, N.; CAVALLARI, T.; STEFANO, L. D. Fusion of Inertial and Visual Measurements for RGB-D SLAM on Mobile Devices. In: **2015 IEEE**

International Conference on Computer Vision Workshop (ICCVW). IEEE, 2015. p. 148–156. ISBN 978-1-4673-9711-7. Disponível em: <<http://ieeexplore.ieee.org/document/7406378/>>. Citado na página 52.

CASTRO, J. B.; RAMANATHAN, A.; CHENNUBHOTLA, C. S. Categorical Dimensions of Human Odor Descriptor Space Revealed by Non-Negative Matrix Factorization. **PLoS ONE**, Public Library of Science, San Francisco, USA, v. 8, n. 9, p. e73289, sep 2013. ISSN 19326203. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3776812/>>. Citado nas páginas 46, 79, 103 e 114.

CECIERJ. **Paladar**. 2017. Disponível em: <<http://cederj.edu.br/fundacao/>>. Acesso em: 01 may 2017. Citado na página 40.

CHAVEZ-ROMERO, R.; CARDENAS, A.; MAYA, M.; SANCHEZ, A.; PIOVESAN, D. Camera Space Particle Filter for the Robust and Precise Indoor Localization of a Wheelchair. **Journal of Sensors**, v. 2016, p. 1–11, 2016. ISSN 1687-725X. Disponível em: <<http://www.hindawi.com/journals/js/2016/8729895/>>. Citado na página 53.

CHRYSOSTOMOU, D.; BARATTINI, P.; KILDAL, J.; WANG, Y.; FO, J.; DAUTENHAHN, K.; FERLAND, F.; TAPUS, A.; VIRK, G. S. ReHRI'17 - Towards Reproducible HRI Experiments. In: **Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17**. New York, New York, USA: ACM Press, 2017. p. 421–422. ISBN 9781450348850. Disponível em: <<http://dl.acm.org/citation.cfm?doid=3029798.3029800>>. Citado na página 26.

CMDE-GROUP. **Integrating Cognitive Architecture and OntSense Ontology**. 2014. Disponível em: <youtu.be/pY7E5NXzfw4>. Acesso em: 02 jun. 2018. Citado na página 109.

COGPOWIKI. **Main Page**. 2016. Disponível em: <<http://www.wiki.cogpo.org/>>. Acesso em: 10 dez. 2016. Citado na página 38.

COLLINS, J. **How Big Is The Web? How Many Websites Are There?** 2016. Disponível em: <<https://www.lifewire.com/how-big-is-the-web-4065573>>. Acesso em: 01 jun. 2018. Citado na página 28.

CUPPONE, A. V.; SQUERI, V.; SEMPRINI, M.; MASIA, L.; KONCZAK, J. Robot-Assisted Proprioceptive Training with Added Vibro-Tactile Feedback Enhances Somatosensory and Motor Performance. **PLOS ONE**, v. 11, n. 10, p. e0164511, oct 2016. ISSN 1932-6203. Disponível em: <<http://dx.plos.org/10.1371/journal.pone.0164511>>. Citado na página 49.

CYCOPR. **Which Cyc is Right for Me?** 2016. Disponível em: <<http://opencyc.org/>>. Acesso em: 10 dez. 2016. Citado nas páginas 35 e 38.

DAML. **DAML Ontology Library**. 2016. Disponível em: <<http://www.daml.org/ontologies/>>. Acesso em: 11 dez. 2016. Citado na página 39.

DARPA. **The DARPA Grand Challenge: Ten Years Later**. 2014. Disponível em: <<http://www.darpa.mil/newsevents/releases/2014/03/13.aspx>>. Acesso em: 24 jan. 2016. Citado na página 26.

_____. **What is the DARPA Robotics Challenge (DRC)?** 2015. Disponível em: <<http://theroboticschallenge.org/overview>>. Acesso em: 24 jan. 2016. Citado na página 26.

DAVANI, D.; ACKERS, F.; TEYAKOME, A. Simulation of a Cognitive Robot Using WebGL. In: **Proceedings of the 2012 Simulation Multi-Conference(SCS)**. San Diego, CA, USA: Society for Modeling & Simulation International (SCS), 2012. ISBN 9781622761005. ISSN 07359276. Citado na página 70.

DBPEDIA-TEAM. **Learn About DBpedia**. 2018. Disponível em: <<http://wiki.dbpedia.org/about>>. Acesso em: 26 mai 2018. Citado nas páginas 85 e 114.

DIAS, M. A.; OSORIO, F. S. Reconfigurable Hardware Architecture for Vision-Based Driving Systems. In: **2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)**. IEEE, 2015. p. 97–102. ISBN 978-1-4673-7129-2. Disponível em: <<http://ieeexplore.ieee.org/document/7402148/>>. Citado na página 52.

DRAVNIEKS, A. **Atlas of odor character profiles**. Philadelphia, PA: ASTM, 1985. 354 p. ISBN 0803104561. Citado na página 46.

DUCHARME, B. **Learning SPARQL**. 2. ed. Sebastopol, CA, USA: O'Reilly, 2013. 386 p. ISBN 978-1449371432. Citado na página 89.

EKMAN, P.; FRIESEN, W. V. Constants across cultures in the face and emotion. **Journal of Personality and Social Psychology**, American Psychological Association, v. 17, n. 2, p. 124–129, 1971. ISSN 1939-1315. Disponível em: <<http://doi.apa.org/getdoi.cfm?doi=10.1037/h0030377>>. Citado na página 101.

ELARA, M.; CALDERON, C.; ZHOU, C. Z. C.; WIJESOMA, W. False alarm metrics: Evaluating safety in human robot interactions. **Robotics Automation and Mechatronics (RAM), 2010 IEEE Conference on**, p. 230–236, 2010. Citado na página 61.

ESCALERA, A. de la; IZQUIERDO, E.; MARTÍN, D.; GARCÍA, F.; ARMINGOL, J. M. Stereo Visual Odometry for Urban Vehicles Using Ground Features. In: . [s.n.], 2016. p. 385–397. Disponível em: <http://link.springer.com/10.1007/978-3-319-27146-0_30>. Citado na página 52.

FABER, M.; KUZ, S.; MAYER, M. P.; SCHLICK, C. M. Design and Implementation of a Cognitive Simulation Model for Robotic Assembly Cells. In: Don Harris (Ed.). **Lecture Notes in Computer Science**. [S.l.]: Springer Berlin Heidelberg, 2013. cap. 24, p. 205–214. Citado na página 71.

FEIL-SEIFER, D. J.; MATARIĆ, M. J. Human-robot interaction. In: MEYERS, R. A. (Ed.). **Encyclopedia of Complexity and Systems Science**. New York, USA: Springer New York, 2009. p. 4643–4659. ISBN 978-0-387-30440-3. Disponível em: <<http://robotics.usc.edu/publications/585/>>. Citado nas páginas 56, 57, 59 e 62.

FERNÁNDEZ, M.; GÓMEZ-PÉREZ, A.; JURISTO, N. Methontology: from ontological art towards ontological engineering. In: **AAAI Spring Symposium**. [S.l.]: AAAI Press, 1997. SS-97-06, p. 33–40. Citado nas páginas 32 e 74.

FERREIRA, M.; COSTA, P.; ROCHA, L.; MOREIRA, A. P. Stereo-based real-time 6-DoF work tool tracking for robot programming by demonstration. **The International Journal of Advanced Manufacturing Technology**, v. 85, n. 1-4, p. 57–69, jul 2016. ISSN 0268-3768. Disponível em: <<http://link.springer.com/10.1007/s00170-014-6026-x>>. Citado na página 53.

FIORINI, S. R.; CARBONERA, J. L.; GONÇALVES, P.; JORGE, V. A.; REY, V. F.; HAI-DEGGER, T.; ABEL, M.; REDFIELD, S. A.; BALAKIRSKY, S.; RAGAVAN, V.; LI, H.; SCHLENOFF, C.; PRESTES, E. Extensions to the core ontology for robotics and automation. **Robotics and Computer-Integrated Manufacturing**, v. 33, p. 3–11, jun 2015. ISSN 07365845. Citado na página 35.

FONG, T.; NOURBAKHSH, I.; DAUTENHAHN, K. A survey of socially interactive robots. **Robotics and Autonomous Systems**, v. 42, n. 3-4, p. 143–166, 2003. Citado na página 25.

FONT, D.; PALLEJÀ, T.; TRESANCHEZ, M.; RUNCAN, D.; MORENO, J.; MARTÍNEZ, D.; TEIXIDÓ, M.; PALACÍN, J. A Proposal for Automatic Fruit Harvesting by Combining a Low Cost Stereovision Camera and a Robotic Arm. **Sensors**, v. 14, n. 7, p. 11557–11579, jun 2014. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/14/7/11557>>. Citado na página 53.

FWLER, M. **UML Distilled: A Brief Guide to the Standard Object Modeling Language**. [S.l.]: Pearson Education, Inc, 2003. Citado na página 140.

Free Software Foundation. **The GNU General Public License v3.0 - GNU Project - Free Software Foundation**. 2018. Disponível em: <<https://www.gnu.org/licenses/gpl-3.0.en.html>>. Acesso em: 16 jan. 2018. Citado nas páginas 29, 92, 99 e 115.

FURUKAWA, H. Human Cognitive Simulation for Evaluation of Human-Robot Interface - A trade-off between flexibility in robot control and mental workload. In: **Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics**. [S.l.]: SciTePress - Science and Technology Publications, 2005. p. 277–282. ISBN 972-8865-29-5. Citado na página 71.

FUSEKI-TEAM. **Apache Jena Fuseki**. 2017. Disponível em: <<https://jena.apache.org/documentation/fuseki2/>>. Acesso em: 05 jun 2017. Citado nas páginas 89 e 92.

GAO, C.; CONG, D.; LIU, X.; YANG, Z.; TAO, H. Hybrid position/force control of 6-dof hydraulic parallel manipulator using force and vision. **Industrial Robot: An International Journal**, v. 43, n. 3, p. 274–283, may 2016. ISSN 0143-991X. Disponível em: <<http://www.emeraldinsight.com/doi/10.1108/IR-10-2015-0192>>. Citado na página 53.

GIL, P.; MATEO, C. M.; DELGADO, A.; TORRES, F. Visual/tactile sensing to monitor grasps with robot-hand for planar elastic objects. In: **Proceedings of ISR 2016: 47st International Symposium on Robotics**. Munich, Germany: VDE Verlag GmbH, 2016. p. 439–445. Citado na página 49.

GOODRICH, M. A.; SCHULTZ, A. C. Human–robot interaction: A survey. **Foundations and Trends in Human–Computer Interaction**, v. 1, n. 3, p. 203–275, 2008. ISSN 1551-3955. Citado nas páginas 56 e 58.

GOOGLE-GROUP. **Documentação da Google Cloud Vision API**. 2018. Disponível em: <<https://cloud.google.com/vision/docs/>>. Acesso em: 05 jun. 2018. Citado na página 114.

GRIGONYTE, G. **Building and Evaluating Domain Ontologies: NLP Contributions**. Berlin: Logos Verlag, 2010. 214 p. ISBN 978-3832526573. Citado na página 33.

GU, Q.-Y.; ISHII, I. Review of some advances and applications in real-time high-speed vision: Our views and experiences. **International Journal of Automation and Computing**, v. 13, n. 4, p. 305–318, 2016. Citado na página 52.

GUAN, H.; HE, W.; ZHANG, J. What to do first: The initial behavior in a multi-sensory household object recognition and categorization system. In: **2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)**. IEEE, 2014. p. 1–6. ISBN 978-1-4799-6732-2. Disponível em: <<http://ieeexplore.ieee.org/document/6997736/>>. Citado na página 46.

GUARINO, C. W. N. An overview of ontoclean. **Handbook on Ontologies, International Handbooks on Information Systems**, p. 201–220, 2009. Citado nas páginas 32 e 74.

GUARINO, N. Formal Ontology in Information Systems. In: **International Conference on Formal Ontology in Information Systems (FOIS'98)**. Trento, Italy: Formal Ontology and Information Systems, 1998. p. 3–15. Citado nas páginas 34 e 74.

GUDWIN, R.; PARAENSE, A.; PAULA, S. de; FRÓES, E.; GIBAUT, W.; CASTRO, E.; FIGUEIREDO, V.; RAIZER, K. An Overview of the Multipurpose Enhanced Cognitive Architecture (MECA). **Procedia Computer Science**, Elsevier, v. 123, p. 155–160, 2018. ISSN 18770509. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1877050918300267>>. Citado nas páginas 67, 69 e 116.

HA, E.; KIM, T.; AHN, D.; JEONG, S.; YOON, I.; HAN, S. A stable control of legged robot based on ultrasonic sensor. In: **2015 15th International Conference on Control, Automation and Systems (ICCAS)**. IEEE, 2015. p. 1256–1258. ISBN 978-8-9932-1508-3. Disponível em: <<http://ieeexplore.ieee.org/document/7364826/>>. Citado na página 47.

HAIDEGGER, T.; BARRETO, M.; GONCALVES, P. J. S.; HABIB, M. K.; RAGAVAN, S. V.; LI, H.; VACCARELLA, A.; PERRONE, R.; PRESTES, E. Robot ontologies for sensor- and Image-guided surgery. In: **2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE)**. [S.l.]: IEEE, 2013. p. 19–24. ISBN 978-1-4673-2939-2. Citado na página 34.

HAIDEGGER, T.; BARRETO, M.; GONÇALVES, P.; HABIBE, M. K.; RAGAVAN, S. K. V.; LI, H.; VACCARELLA, A.; PERRONE, R.; PRESTES, E. Applied ontologies and standards for service robots. **Robotics and Autonomous Systems**, Elsevier B.V., v. 61, p. 1215–1223, 2013. Citado na página 35.

HE, Y.; CHEN, Y.; XU, Y.; HUANG, Y.; CHEN, S. Autonomous Detection of Weld Seam Profiles via a Model of Saliency-Based Visual Attention for Robotic Arc Welding. **Journal of Intelligent & Robotic Systems**, v. 81, n. 3-4, p. 395–406, mar 2016. ISSN 0921-0296. Disponível em: <<http://link.springer.com/10.1007/s10846-015-0226-y>>. Citado na página 53.

HEIDELBERGER, M. Gustav Theodor Fechner. In: **Statisticians of the Centuries**. New York, NY: Springer New York, 2001. p. 142–147. ISBN 978-1-4613-0179-0. Citado na página 77.

HERTEL, A.; BROEKSTRA, J.; STUCKENSCHMIDT, H. RDF Storage and Retrieval Systems. In: STAAB, S.; STUDER, R. (Ed.). **Handbook on Ontologies**. 2. ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. cap. 1, p. 489–508. ISBN 978-3-540-70999-2. Citado na página 32.

HOFFMANN, M.; ŠTĚPÁNOVÁ, K.; REINSTEIN, M. The effect of motor action and different sensory modalities on terrain classification in a quadruped robot running with multiple gaits. **Robotics and Autonomous Systems**, v. 62, n. 12, p. 1790–1798, dec 2014. ISSN 09218890. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S092188901400133X>>. Citado na página 49.

- HOLMAN, D.; GIROUARD, A.; BENKO, H.; VERTEGAAL, R. The Design of Organic User Interfaces: Shape, Sketching and Hypercontext. **Interacting with Computers**, v. 25, n. 2, p. 133–142, mar 2013. ISSN 0953-5438. Disponível em: <<https://academic.oup.com/iwc/article-lookup/doi/10.1093/iwc/iws018>>. Citado na página 48.
- HUANG, A. L.; CHEN, X.; HOON, M. A.; CHANDRASHEKAR, J.; GUO, W.; TRÄNKNER, D.; RYBA, N. J. P.; ZUKER, C. S. The cells and logic for mammalian sour taste detection. **Nature**, Nature Publishing Group, v. 442, n. 7105, p. 934–938, aug 2006. ISSN 0028-0836. Disponível em: <<http://www.nature.com/articles/nature05084>>. Citado na página 103.
- HUMMEL, J. E. Object recognition. In: **The Oxford Handbook of Cognitive Psychology**. [S.l.]: Oxford University Press, 2013. p. 32–46. ISBN 9780195376746. Citado na página 64.
- HUSSEIN, A.; MARIN-PLAZA, P.; MARTIN, D.; ESCALERA, A. de la; ARMINGOL, J. M. Autonomous off-road navigation using stereo-vision and laser-rangefinder fusion for outdoor obstacles detection. In: **2016 IEEE Intelligent Vehicles Symposium (IV)**. IEEE, 2016. p. 104–109. ISBN 978-1-5090-1821-5. Disponível em: <<http://ieeexplore.ieee.org/document/7535372/>>. Citado na página 53.
- Hyo Seung Han; Junwoo Park; Tien Dat Nguyen; Uikyum Kim; Canh Toan Nguyen; Hoa Phung; Hyouk Ryeol Choi. A highly sensitive dual mode tactile and proximity sensor using Carbon Microcoils for robotic applications. In: **2016 IEEE International Conference on Robotics and Automation (ICRA)**. IEEE, 2016. p. 97–102. ISBN 978-1-4673-8026-3. Disponível em: <<http://ieeexplore.ieee.org/document/7487121/>>. Citado na página 49.
- IEEE. "IEEE 1872 2015, Standard Ontologies for Robotics and Automation". New York, NY, 2015. Citado nas páginas 28, 34, 35, 55, 57, 74 e 75.
- IEEE-RAS/SC-ORA. **1872-2015 - IEEE Standard Ontologies for Robotics and Automation**. 2016. Disponível em: <<https://standards.ieee.org/findstds/standard/1872-2015.html>>. Acesso em: 10 dez. 2016. Citado na página 38.
- IFR. **World Robotics Survey: Service robots are conquering the world**. Alemanha, Frankfurt, 2015. Disponível em: <http://www.worldrobotics.org/index.php?id=home&news_id=285>. Acesso em: 24 jan. 2016. Citado na página 25.
- INSENT. **Taste Sensing System TS-5000Z**. 2017. Disponível em: <http://www.insent.co.jp/en/products/TS5000Z_web_E_0512.pdf>. Acesso em: 21 jan. 2017. Citado nas páginas 40 e 41.
- ISHIDA, H.; WADA, Y.; MATSUKURA, H. Chemical Sensing in Robotic Applications: A Review. **IEEE Sensors Journal**, IEEE-Inst Electrical Electronics Engineers Inc, 445 HOES LANE, PISCATAWAY, NJ 08855-4141 USA, v. 12, n. 11, p. 3163–3173, nov 2012. ISSN 1530-437X. Disponível em: <<http://ieeexplore.ieee.org/document/6247459/>>. Citado nas páginas 42, 43 e 45.
- ISO/IEC:25010. **Software engineering-Software product Quality Requirements and Evaluation (SQuaRE)Quality model**. 1. ed. Geneva, Switzerland: ISO International Organization for Standardization, 2011. Citado nas páginas 55 e 61.
- JAMALI, N.; MAGGIALI, M.; GIOVANNINI, F.; METTA, G.; NATALE, L. A new design of a fingertip for the iCub hand. In: **2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. IEEE, 2015. p. 2705–2710. ISBN 978-1-4799-9994-1. Disponível em: <<http://ieeexplore.ieee.org/document/7353747/>>. Citado na página 49.

JAMONE, L.; UGUR, E.; CANGELOSI, A.; FADIGA, L.; BERNARDINO, A.; PIATER, J.; SANTOS-VICTOR, J. Affordances in psychology, neuroscience and robotics: a survey. **IEEE Transactions on Cognitive and Developmental Systems**, p. 1–1, 2016. ISSN 2379-8920. Citado nas páginas 114 e 115.

JENA-TEAM. **Apache Jena**. 2018. Disponível em: <<http://jena.apache.org/>>. Acesso em: 28 mai 2018. Citado nas páginas 36, 37, 89 e 92.

JI, Z.; QIU, R.; NOYVIRT, A.; SOROKA, A.; PACKIANATHER, M.; SETCHI, R.; LI, D.; XU, S. Towards automated task planning for service robots using semantic knowledge representation. In: **IEEE 10th International Conference on Industrial Informatics**. [S.l.]: IEEE, 2012. p. 1194–1201. ISBN 978-1-4673-0311-8. Citado na página 35.

JOHNSON-LAIRD, P. N.; KHEMLANI, S. S.; PGOODWIN, G. Logic, probability, and human reasoning. **Trends in Cognitive Sciences**, Elsevier Ltd, v. 19, n. 4, p. 201–214, apr 2015. ISSN 13646613. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1364661315000303>>. Citado nas páginas 65 e 69.

JORGE, V. A.; REY, V. F.; MAFFEI, R.; FIORINI, S. R.; CARBONERA, J. L.; BRANCHI, F.; MEIRELES, J. P.; FRANCO, G. S.; FARINA, F.; SILVA, T. S. da; KOLBERG, M.; ABEL, M.; PRESTES, E. Exploring the IEEE ontology for robotics and automation for heterogeneous agent interaction. **Robotics and Computer-Integrated Manufacturing**, v. 33, p. 12–20, jun 2015. ISSN 07365845. Citado na página 83.

KALAT, J. W. **Introduction to Psychology**. 10. ed. [S.l.]: Wadsworth Publishing, 2014. 640 p. ISBN 978-0495810766. Citado na página 62.

KAPPASSOV, Z.; CORRALES, J.-A.; PERDEREAU, V. Tactile sensing in dexterous robot hands — Review. **Robotics and Autonomous Systems**, v. 74, p. 195–220, dec 2015. ISSN 09218890. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0921889015001621>>. Citado na página 48.

KHILAR, R.; CHITRAKALA, S.; SELVAMPARVATHY, S. 3D image reconstruction: Techniques, applications and challenges. In: **2013 International Conference on Optical Imaging Sensor and Security (ICOSS)**. IEEE, 2013. p. 1–6. ISBN 978-1-4799-0936-0. Disponível em: <<http://ieeexplore.ieee.org/document/6678395/>>. Citado na página 52.

KHRONOS, G. **WebGL Specification**. 2016. Disponível em: <<https://www.khronos.org/registry/webgl/specs/1.0/>>. Acesso em: 24 jan. 2016. Citado na página 70.

KNOWROB. **Ontologies**. 2016. Disponível em: <<http://www.knowrob.org/ontologies>>. Acesso em: 29 dez. 2016. Citado nas páginas 35 e 39.

KOBAYASHI, Y.; IKEZAKI, H. Advanced Taste Sensors Based on Artificial Lipid Membrane. In: TOKO, K. (Ed.). **Biochemical Sensors, Mimicking Gustatory and Olfactory Senses**. [S.l.]: Pan Stanford, 2013. cap. 1, p. 5–44. Citado nas páginas 40, 41 e 77.

KOTSERUBA, I.; GONZALEZ, O. J. A.; TSOTSOS, J. K. A review of 40 years of cognitive architecture research: Focus on perception, attention, learning and applications. **Computing Research Repository - CoRR**, abs/1610.08602v3, 2018. Disponível em: <<https://arxiv.org/abs/1610.08602v3>>. Citado nas páginas 67, 68 e 116.

LACHAT, E.; MACHER, H.; LANDES, T.; GRUSSENMEYER, P. Assessment and Calibration of a RGB-D Camera (Kinect v2 Sensor) Towards a Potential Use for Close-Range 3D Modeling. **Remote Sensing**, v. 7, n. 10, p. 13070–13097, oct 2015. ISSN 2072-4292. Disponível em: <http://www.mdpi.com/2072-4292/7/10/13070>. Citado na página 51.

LADDI, A.; BHARDWAJ, V.; MAHAPATRA, P.; PANKAJ, D.; KUMAR, A. Real-time stereo generation for surgical vision during minimal invasive robotic surgery. In: . [s.n.], 2016. p. 020005. Disponível em: <http://aip.scitation.org/doi/abs/10.1063/1.4942687>. Citado na página 53.

LAIRD, J. E. The Soar Cognitive Architecture. **AISB Quarterly**, v. 171, n. 134, p. 224–235, nov 2012. Citado na página 30.

_____. **The Soar 9 Tutorial**. Ann Arbor, Michigan, EUA, 2017. Updated for Soar 9.6.0. Disponível em: <https://soar.eecs.umich.edu/downloads/Documentation/SoarTutorial/>. Citado na página 107.

LAIRD, J. E.; CONGDON, C. B.; ASSANIE, M.; DERBINSKY, N.; XU, J. **The Soar User's Manual**. Ann Arbor, Michigan, EUA, 2017. 1–303 p. Version 9.6.0. Disponível em: <https://soar.eecs.umich.edu/downloads/SoarManual.pdf>. Citado nas páginas 30, 103 e 116.

LANZANI, G.; ANTOGNAZZA, M. R.; De Vittorio, M.; PETRONI, S.; RIZZI, F. Biosensing Detection. In: **Bioinspired Approaches for Human-Centric Technologies**. [S.l.]: Springer International Publishing, 2014. p. 77–109. Citado nas páginas 41 e 48.

LEHMAN, J. F.; LAIRD, J. E.; ROSENBLOOM, P. S. A gentle introduction to Soar, an architecture for human cognition. In: OSHERSON, D. N.; SCARBOROUGH, D.; STERNBERG, S. (Ed.). **Invitation to cognitive science**. 2. ed. Cambridge, Massachusetts: The MIT Press, 1998. v. 4, cap. 6, p. 211–248. Citado na página 69.

LEITE, I.; MARTINHO, C.; PAIVA, A. Social Robots for Long-Term Interaction: A Survey. **International Journal of Social Robotics**, v. 5, n. 2, p. 291–308, 2013. ISSN 1875-4791. Citado na página 59.

LEMAIGNAN, S.; ROS, R.; MÖSENLECHNER, L.; ALAMI, R.; BEETZ, M. ORO, a knowledge management platform for cognitive architectures in robotics. In: **2010 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.]: IEEE, 2010. p. 3548–3553. ISBN 978-1-4244-6674-0. Citado na página 36.

LIU, S.; TAN, P.; YUAN, L.; SUN, J.; ZENG, B. MeshFlow: Minimum Latency Online Video Stabilization. In: LEIBE, B.; MATAS, J.; SEBE, N.; WELLING, M. (Ed.). Cham: Springer International Publishing, 2016, (Lecture Notes in Computer Science, v. 9910). p. 800–815. ISBN 978-3-319-46465-7. Disponível em: http://link.springer.com/10.1007/978-3-319-46466-4_48. Citado na página 52.

LOA-GROUP. **Descriptive Ontology for Linguistic and Cognitive Engineering**. 2016. Disponível em: <http://www.loa.istc.cnr.it/old/DOLCE.html>. Acesso em: 10 dez. 2016. Citado na página 38.

LÓPEZ-SERRANO, L. L.; VOULOUTSI, V.; Escudero Chimeno, A.; MATHEWS, Z.; VERSCHURE, P. F. M. J. Insect-Like Odor Classification and Localization on an Autonomous Robot.

In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [S.l.: s.n.], 2012. v. 7375 LNAI, p. 371–372. Citado na página 44.

LÓPEZ, L.; MATHEWS, Z.; VERSCHURE, P.; VOULOUTSI, V.; CHIMENO, A. E.; MARCOS, E.; BADIA, S. B. I.; ZIYATDINOV, A.; LLUNA, A. P. I. Moth-like chemo-source localization and classification on an indoor autonomous robot. In: PRAMATAROVA, L. (Ed.). **On Biomimetics**. Rijeka: InTech, 2011. cap. 21. Citado na página 42.

MAGASSOUBA, A.; BERTIN, N.; CHAUMETTE, F. Sound-based control with two microphones. In: **2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. IEEE, 2015. p. 5568–5573. ISBN 978-1-4799-9994-1. Disponível em: <<http://ieeexplore.ieee.org/document/7354166/>>. Citado na página 47.

MARJOVI, A.; MARQUES, L. Optimal spatial formation of swarm robotic gas sensors in odor plume finding. **Autonomous Robots**, Institute of Systems and Robotics, University of Coimbra, 3030-290 Coimbra, Portugal, v. 35, n. 2-3, p. 93–109, oct 2013. ISSN 0929-5593. Citado nas páginas 42 e 44.

MARKMAN, A. B. Knowledge Representation. In: **Stevens' Handbook of Experimental Psychology**. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2002. Citado na página 64.

MARSH, E. J.; BUTLER, A. C. Memory in Educational Settings. In: Daniel Reisberg (Ed.). **The Oxford Handbook of Cognitive Psychology**. [S.l.]: Oxford University Press, 2013. p. 299–320. ISBN 9780195376746. Citado na página 64.

MARTINSON, E.; BROCK, D. Auditory Perspective Taking. **IEEE Transactions on Cybernetics**, v. 43, n. 3, p. 957–969, jun 2013. ISSN 2168-2267. Disponível em: <<http://ieeexplore.ieee.org/document/6334486/>>. Citado na página 47.

MARZBAN, M.; PACKIRISAMY, M.; DARGAHI, J. Fiber-mirror integrated compliant mechanical system for measuring force and displacement simultaneously. In: **2015 Photonics North**. IEEE, 2015. p. 1–8. ISBN 978-1-4673-6805-6. Disponível em: <<http://ieeexplore.ieee.org/document/7292534/>>. Citado na página 48.

MAYER, R. E. **Thinking, Problem Solving, Cognition**. 2. ed. New York: W.H.Freeman, 1992. ISBN 9780716722151. Citado na página 66.

_____. A taxonomy for computer-based assessment of problem solving. **Computers in Human Behavior**, v. 18, n. 6, p. 623–632, nov 2002. ISSN 07475632. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0747563202000201>>. Citado na página 65.

MCNEIL, J. G.; LATTIMER, B. Y. Autonomous Fire Suppression System for Use in High and Low Visibility Environments by Visual Servoing. **Fire Technology**, v. 52, n. 5, p. 1343–1368, sep 2016. ISSN 0015-2684. Disponível em: <<http://link.springer.com/10.1007/s10694-016-0564-8>>. Citado na página 53.

Moradi Dalvand, M.; NAHAVANDI, S.; HOWE, R. D. Fast vision-based catheter 3D reconstruction. **Physics in Medicine and Biology**, v. 61, n. 14, p. 5128–5148, jul 2016. ISSN 0031-9155. Disponível em: <<http://stacks.iop.org/0031-9155/61/i=14/a=5128?key=crossref.36755a7cc9bf36ec3d59298fe713bc58>>. Citado na página 53.

MOULIN-FRIER, C.; FISCHER, T.; PETIT, M.; POINTEAU, G.; PUIGBO, J.-Y.; PATTACINI, U.; LOW, S. C.; CAMILLERI, D.; NGUYEN, P.; HOFFMANN, M.; CHANG, H. J.; ZAMBELLI, M.; MEALIER, A.-L.; DAMIANOU, A.; METTA, G.; PRESCOTT, T. J.; DEMIRIS, Y.; DOMINEY, P. F.; VERSCHURE, P. F. M. J. **DAC-h3: A Proactive Robot Cognitive Architecture to Acquire and Express Knowledge About the World and the Self**. 2017. Disponível em: <<http://arxiv.org/abs/1706.03661>>. Citado na página 70.

MUEGGLER, E.; HUBER, B.; SCARAMUZZA, D. Event-based, 6-DOF pose tracking for high-speed maneuvers. In: **2014 IEEE/RSJ International Conference on Intelligent Robots and Systems**. IEEE, 2014. p. 2761–2768. ISBN 978-1-4799-6934-0. Disponível em: <<http://ieeexplore.ieee.org/document/6942940/>>. Citado na página 51.

MUNOZ, P.; R-MORENO, M. D.; GALLEGU, P.; CASTANO, B. A Cognitive Architecture and Simulation Environment for the Pntino Robot. **2011 IEEE Fourth International Conference on Space Mission Challenges for Information Technology**, n. i, p. 129–136, 2011. Citado na página 67.

MURPHY, R. R.; SCHRECKENGHOST, D. Survey of metrics for human-robot interaction. **ACM/IEEE International Conference on Human-Robot Interaction**, p. 197–198, 2013. ISSN 21672148. Citado nas páginas 59, 60 e 61.

MUSEN, M. A. The protégé project: a look back and a look forward. **AI Matters**, v. 1, n. 4, p. 4–12, jun 2015. ISSN 23723483. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2757001.2757003>>. Citado na página 74.

NAGLE, H.; GUTIERREZ-OSUNA, R.; SCHIFFMAN, S. The how and why of electronic noses. **IEEE Spectrum**, v. 35, n. 9, p. 22–31, 1998. ISSN 00189235. Disponível em: <<http://ieeexplore.ieee.org/document/715180/>>. Citado na página 138.

NASA. **Mars: Exploration Rovers**. 2015. Disponível em: <<http://mars.nasa.gov/mer/home/>>. Acesso em: 24 jan. 2016. Citado na página 58.

NEUSTEIN, A. **Speech and Automata in Health Care**. Fort Lee, NJ.: De Gruyter, 2014. ISBN 9781614515159. Citado na página 46.

NORMAN, D. A.; RUMELHART, D. E. **Explorations in Cognition**. San Francisco: W. H. Freeman, 1975. 430 p. ISBN 9780716707363. Citado na página 63.

NOVIKOVA, J.; WATTS, L.; INAMURA, T. Modeling Human-Robot Collaboration in a Simulated Environment. In: **Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts - HRI'15 Extended Abstracts**. New York, New York, USA: ACM Press, 2015. p. 181–182. ISBN 9781450333184. Citado na página 70.

NOY, N. F.; MCGUINNESS, D. L. **Ontology Development 101: A Guide to Creating Your First Ontology**. [S.l.], 2001. Disponível em: <https://protege.stanford.edu/publications/ontology_development/ontology101.pdf>. Citado nas páginas 74 e 92.

OBO. **The OBO Foundry**. 2016. Disponível em: <<http://www.obofoundry.org/>>. Acesso em: 11 dez. 2016. Citado na página 39.

OLIVER, B.; JEAHEUNG, P.; MARC, T. Mobility and Manipulation. In: SICILIANO, B.; KHATIB, O. (Ed.). **Springer Handbook of Robotics**. 2. ed. Berlin, Heidelberg: Springer International Publishing, 2016. cap. 40, p. 1007–1036. ISBN 978-3-319-32550-7. Disponível em: <http://link.springer.com/10.1007/978-3-319-32552-1>. Citado na página 137.

PARAENSE, A. L.; RAIZER, K.; PAULA, S. M. de; ROHMER, E.; GUDWIN, R. R. The cognitive systems toolkit and the CST reference cognitive architecture. **Biologically Inspired Cognitive Architectures**, v. 17, p. 32–48, jul 2016. ISSN 2212683X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2212683X1630038X><http://linkinghub.elsevier.com/retrieve/pii/S2212683X1630038X>. Citado na página 69.

PARK, D.; ERICKSON, Z.; BHATTACHARJEE, T.; KEMP, C. C. Multimodal execution monitoring for anomaly detection during robot manipulation. In: **2016 IEEE International Conference on Robotics and Automation (ICRA)**. IEEE, 2016. p. 407–414. ISBN 978-1-4673-8026-3. Disponível em: <http://ieeexplore.ieee.org/document/7487160/>. Citado na página 47.

PEASE, A. **Ontology: A Pratical Guide**. Angwin, CA: Articulate Software Press, 2011. 303 p. ISBN 978-1-889455-10-5. Citado nas páginas 33 e 75.

_____. **Suggested Upper Merged Ontology (SUMO)**. 2016. Disponível em: <http://www.adampease.org/OP/>. Acesso em: 10 dez. 2016. Citado nas páginas 38, 74 e 75.

POLDRACK, R. **Cognitive Atlas**. 2016. Disponível em: <http://www.cognitiveatlas.org/>. Acesso em: 24 jan. 2016. Citado na página 33.

PPS-INC. **FingerTPS system**. 2018. Disponível em: <https://pressureprofile.com/finger-tps>. Acesso em: 27 apr. 2018. Citado na página 50.

PRESTES, E.; CARBONERA, J. L.; FIORINI, S. R.; JORGE, V. A. M.; ABEL, M.; MADHAVAN, R.; LOCORO, A.; GONCALVES, P.; BARRETO, M. E.; HABIB, M.; CHIBANI, A.; GERARD, S.; AMIRAT, Y.; C.SCHLENOFF. Towards a core ontology for robotics and automation. **Robotics and Autonomous Systems**, Elsevier B.V., v. 161, p. 1193–1204, 2013. Citado nas páginas 28 e 35.

PROTéGé. **Welcome to the Protege Ontology Library**. 2016. Disponível em: http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library. Acesso em: 11 dez. 2016. Citado nas páginas 37 e 39.

_____. **Welcome to the Protégé wiki!** 2018. Disponível em: https://protegewiki.stanford.edu/wiki/Main_Page. Acesso em: 11 mai. 2018. Citado nas páginas 74, 83 e 92.

PURL-TEAM. **PURL help**. 2018. Disponível em: <https://archive.org/services/purl/help>. Acesso em: 25 mai 2018. Citado na página 83.

RASKIN, V.; TAYLOR, J. M. Comprehensive semantics in robotic intelligence and communication: Necessity and feasibility. In: **6th International Conference on Automation, Robotics and Applications**. [S.l.: s.n.], 2015. p. 519–524. Citado na página 37.

RAVULAKOLLU, K. K.; ERWIN, H.; BURN, K. Improving Robot-Human Communication by Integrating Visual Attention and Auditory Localization Using a Biologically Inspired Model of Superior Colliculus. **Advanced Materials Research**, v. 403-408, p. 4711–4717, nov 2011. ISSN 1662-8985. Disponível em: <http://www.scientific.net/AMR.403-408.4711>. Citado na página 47.

- RDF-TEAM. **Resource Description Framework (RDF)**. 2016. Disponível em: <<https://www.w3.org/RDF/>>. Acesso em: 27 dez. 2016. Citado nas páginas 33 e 36.
- REISBERG, D. (Ed.). **The Oxford Handbook of Cognitive Psychology**. [S.l.]: Oxford University Press, 2013. ISBN 9780195376746. Citado na página 63.
- RIAZUELO, L.; TENORTH, M.; MARCO, D. D.; SALAS, M.; GALVEZ-LOPEZ, D.; MOSENLECHNER, L.; KUNZE, L.; BEETZ, M.; TARDOS, J. D.; MONTANO, L.; MONTIEL, J. M. M. RoboEarth Semantic Mapping: A Cloud Enabled Knowledge-Based Approach. **IEEE Transactions on Automation Science and Engineering**, 2015. Citado na página 35.
- ROBOEARTH-TEAM. **About RoboEarth**. 2015. Disponível em: <<http://robearth.ethz.ch/>>. Acesso em: 29 dez. 2016. Citado nas páginas 35 e 36.
- ROBOHOW-TEAM. **Project overview**. 2016. Disponível em: <<http://robohow.eu/project>>. Acesso em: 29 dez. 2016. Citado na página 36.
- ROBOTICS, V. O. **A Roadmap for U.S. Robotics: From Internet to Robotics**. 2013. Disponível em: <<https://robotics-vo.us/sites/default/files/2013%20Robotics%20Roadmap-rs.pdf>>. Acesso em: 24 jan. 2016. Citado na página 26.
- ROPPO, D. **Robotics Projects**. 2018. Disponível em: <<http://www.ropdesign.com/projects/>>. Acesso em: 01 may 2018. Citado na página 53.
- ROS-TEAM. **About ROS**. 2015. Disponível em: <<http://www.ros.org/about-ros/>>. Acesso em: 24 jan. 2016. Citado nas páginas 37, 39 e 99.
- ROSATI, G.; OSCARI, F.; SPAGNOL, S.; AVANZINI, F.; MASIERO, S. Effect of task-related continuous auditory feedback during learning of tracking motion exercises. **Journal of Neuro Engineering and Rehabilitation**, v. 9, n. 1, p. 79, 2012. ISSN 1743-0003. Disponível em: <<http://jneuroengrehab.biomedcentral.com/articles/10.1186/1743-0003-9-79>>. Citado na página 46.
- ROSTA-TEAM. **RoSta Main Page**. 2009. Disponível em: <http://wiki.robot-standards.org/index.php/Main_Page>. Acesso em: 26 dez. 2016. Citado na página 36.
- SADUN, A. S.; JALANI, J.; SUKOR, J. A. Force Sensing Resistor (FSR): a brief overview and the low-cost sensor for active compliance control. In: JIANG, X.; CHEN, G.; CAPI, G.; ISHLL, C. (Ed.). **Proceedings of First International Workshop on Pattern Recognition**. [S.l.: s.n.], 2016. p. 1001112. Citado na página 49.
- SANTHOSH, S. Innovative methodology and designing of intelligent mobile robots computer vision with genetic algorithm SLE mechanism and advanced sensors in managing disaster. In: **2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014]**. IEEE, 2014. p. 1692–1697. ISBN 978-1-4799-2397-7. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7054895>>. Citado na página 47.
- SCHLENOFF, C.; PRESTES, E.; MADHAVAN, R.; GONCALVES, P.; LI, H.; BALAKIRSKY, S.; KRAMER, T.; MIGUELANEZ, E. An IEEE standard Ontology for Robotics and Automation. In: **2012 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.]: IEEE, 2012. p. 1337–1342. ISBN 978-1-4673-1736-8. Citado na página 35.

SHANG, W.; CAO, X.; MA, H.; ZANG, H.; WEI, P. Kinect-Based Vision System of Mine Rescue Robot for Low Illuminous Environment. **Journal of Sensors**, v. 2016, p. 1–9, 2016. ISSN 1687-725X. Disponível em: <<http://www.hindawi.com/journals/js/2016/8252015/>>. Citado na página 53.

SMITH, B. **An Introduction to Ontology: From Aristotle to the Universal Core**. 2009. Disponível em: <http://ontology.buffalo.edu/smith/IntroOntology_Course.html>. Acesso em: 05 jun. 2018. Citado nas páginas 114 e 167.

SOAR-GROUP. **SML Quick Start Guide**. 2014. Disponível em: <<https://soar.eecs.umich.edu/articles/articles/soar-markup-language-sml/78-sml-quick-start-guide>>. Acesso em: 02 jun. 2018. Citado nas páginas 104 e 106.

SOCK, J.; KIM, J.; MIN, J.; KWAK, K. Probabilistic traversability map generation using 3D-LIDAR and camera. In: **2016 IEEE International Conference on Robotics and Automation (ICRA)**. IEEE, 2016. p. 5631–5637. ISBN 978-1-4673-8026-3. Disponível em: <<http://ieeexplore.ieee.org/document/7487782/>>. Citado na página 52.

SPIERS, A. J.; LIAROKAPIS, M. V.; CALLI, B.; DOLLAR, A. M. Single-Grasp Object Classification and Feature Extraction with Simple Robot Hands and Tactile Sensors. **IEEE Transactions on Haptics**, v. 9, n. 2, p. 207–220, apr 2016. ISSN 1939-1412. Disponível em: <<http://ieeexplore.ieee.org/document/7390277/>>. Citado na página 49.

STARDOG-UNION. **FAQ - Pellet**. 2016. Disponível em: <<https://github.com/stardog-union/pellet/wiki/FAQ>>. Acesso em: 27 dez. 2016. Citado nas páginas 37, 75 e 95.

STEINFELD, A.; FONG, T.; KABER, D.; LEWIS, M.; SCHOLTZ, J.; SCHULTZ, A.; GO-ODRICH, M. Common metrics for human-robot interaction. In: **Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction - HRI '06**. New York, New York, USA: ACM Press, 2006. p. 33. ISBN 1595932941. Citado na página 61.

STERNBERG, R. J.; STERNBERG, K. **Cognitive Psychology**. 6. ed. [S.l.]: Wadsworth Publishing, 2012. 643 p. ISBN 978-1-111-34476-4. Citado na página 62.

SULLIVAN, J. C.; MITCHINSON, B.; PEARSON, M. J.; EVANS, M.; LEPORA, N. F.; FOX, C. W.; MELHUISE, C.; PRESCOTT, T. J. Tactile Discrimination Using Active Whisker Sensors. **IEEE Sensors Journal**, v. 12, n. 2, p. 350–362, feb 2012. ISSN 1530-437X. Disponível em: <<http://ieeexplore.ieee.org/document/5756632/>>. Citado na página 48.

SYNTOUCH. **The BioTac Toccare**. 2018. Disponível em: <www.syntouchinc.com/en/products/>. Acesso em: 27 apr. 2018. Citado na página 50.

TAATGEN, N.; ANDERSON, J. R. The Past, Present, and Future of Cognitive Architectures. **Topics in Cognitive Science**, v. 2, n. 4, p. 693–704, 2010. ISSN 17568757. Disponível em: <<http://doi.wiley.com/10.1111/j.1756-8765.2009.01063.x>>. Citado na página 68.

TEKSCAN. **Grip System**. 2018. Disponível em: <www.tekscan.com/products-solutions/systems/grip-system>. Acesso em: 27 apr. 2018. Citado na página 50.

TRAIN, B.; AHMED, R.; BANDAWE, C.; COCKCROFT, K.; CRAFTFORD, A.; GREENOP, K.; STACEY, M.; TOMLINSON, M.; TOMMY, J. **Fresh Perspectives: Introduction to Psychology**. 1. ed. [S.l.]: David Langhan and Belinda Train, 2007. 497 p. ISBN 978 86891 289 6. Citado na página 62.

ULLAH, A.; VIRK, G. S. Mobile robot navigation via olfaction. In: **Adaptive Mobile Robotics - Proceedings of the 15th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, CLAWAR 2012**. Applied Sciences Campus, South Westphalia Univ, Soest, Germany: [s.n.], 2012. p. 97–104. Citado na página 44.

Unity Technologies. **Unity**. 2018. Disponível em: <<https://unity3d.com/pt>>. Acesso em: 2 jan. 2018. Citado na página 99.

VÁZQUEZ-ARELLANO, M.; GRIEPENTROG, H.; REISER, D.; PARAFOROS, D. 3-D Imaging Systems for Agricultural Applications—A Review. **Sensors**, v. 16, n. 5, p. 618, apr 2016. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/16/5/618>>. Citado na página 53.

VELOSO, M. **Introduction**. [S.l.]: WTEC, 2012. 1-11 p. Citado na página 56.

VIDAL, E. G.; ZARRICUETA, E. F.; CHEEIN, F. A. Human-inspired sound environment recognition system for assistive vehicles. **Journal of Neural Engineering**, v. 12, n. 1, p. 016012, feb 2015. ISSN 1741-2560. Disponível em: <<http://stacks.iop.org/1741-2552/12/i=1/a=016012?key=crossref.d50f9e55a7a6eabc44bf2f579f08c7ba>>. Citado na página 47.

VILLARREAL, B. L.; OLAGUE, G.; GORDILLO, J. Synthesis of odor tracking algorithms with genetic programming. **Neurocomputing**, Center for Robotics and Intelligent Systems, Tecnológico de Monterrey, Monterrey, N.L., Mexico, v. 175, p. 1019–1032, jan 2016. ISSN 09252312. Citado nas páginas 42 e 43.

W3C. **Ontology editors**. 2018. Disponível em: <https://www.w3.org/wiki/Ontology_editors>. Acesso em: 21 may. 2018. Citado nas páginas 74, 89, 114 e 140.

W3C-SENSORES. **Review of Sensor and Observations Ontologies**. 2016. Disponível em: <https://www.w3.org/2005/Incubator/ssn/wiki/Review_of_Sensor_and_Observations_Ontologies>. Acesso em: 11 dez. 2016. Citado na página 39.

WANG, S.; YANG, P.; SUN, H. Design and implementation of auditory system for mobile robot based on Kinect sensor. In: **2016 12th World Congress on Intelligent Control and Automation (WCICA)**. IEEE, 2016. p. 3094–3098. ISBN 978-1-4673-8414-8. Disponível em: <<http://ieeexplore.ieee.org/document/7578518/>>. Citado na página 46.

WANG, T.; FARAJOLLAHI, M.; CHOI, Y. S.; LIN, I.-T.; MARSHALL, J. E.; THOMPSON, N. M.; KAR-NARAYAN, S.; MADDEN, J. D. W.; SMOUKOV, S. K. Electroactive polymers for sensing. **Interface Focus**, v. 6, n. 4, p. 20160026, aug 2016. ISSN 2042-8898. Citado na página 49.

WEB-ONTOLOGY-WG. **OWL Web Ontology Language Guide**. 2016. Disponível em: <<https://www.w3.org/TR/owl-guide/>>. Acesso em: 29 dez. 2016. Citado na página 37.

_____. **SPARQL Query Language for RDF**. 2017. Disponível em: <<https://www.w3.org/TR/rdf-sparql-query/>>. Acesso em: 05 jun 2017. Citado na página 89.

WIKIDATA-TEAM. **Welcome to Wikidata**. 2018. Disponível em: <https://www.wikidata.org/wiki/Wikidata:Main_Page>. Acesso em: 26 mai 2018. Citado na página 85.

WINIARSKI, T.; BANACHOWICZ, K.; SEREDYŃSKI, D. Multi-sensory Feedback Control in Door Approaching and Opening. In: **Intelligent Systems'2014. Advances in Intelligent Systems and Computing**. Springer, Cham, 2015. p. 57–70. Disponível em: <http://link.springer.com/10.1007/978-3-319-11310-4_6>. Citado nas páginas 49 e 50.

YOUSSEFI, S.; DENEI, S.; MASTROGIOVANNI, F.; CANNATA, G. A real-time data acquisition and processing framework for large-scale robot skin. **Robotics and Autonomous Systems**, v. 68, p. 86–103, jun 2015. ISSN 09218890. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0921889015000172>>. Citado na página 49.

ZARZHITSKY, D.; SPEARS, D.; THAYER, D.; SPEARS, W. Agent-Based Chemical Plume Tracing Using Fluid Dynamics. In: HINCHEY, M. G.; RASH, J. L.; TRUSZKOWSKI, W. F.; ROUFF, C. A. (Ed.). **Formal Approaches to Agent-Based Systems**. [S.l.]: Springer Berlin Heidelberg, 2005. cap. 10, p. 146–160. ISBN 978-3-540-24422-6. Citado na página 43.

ZESHAN, F.; MOHAMAD, R.; AHMAD, M. N.; HUSSAIN, S. A. **Ontology for autonomous mobile robot system**. 2014. 1073–1085 p. Citado nas páginas 35 e 115.

ZHU, Y.; WANG, X.; XU, B. Design of vision-based obstacle crossing of high-voltage line inspection robot. In: **2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)**. IEEE, 2016. p. 506–511. ISBN 978-1-5090-2733-0. Disponível em: <<http://ieeexplore.ieee.org/document/7574879/>>. Citado na página 53.

GLOSSÁRIO

advecção : (1) deslocamento de massa de ar no sentido horizontal; (2) transporte de algo de uma região para outra utilizando um fluído.

ambientes estruturados : constituídos por ambientes controlados, como chão de fábrica e configurações experimentais específicas (OLIVER; JEAHEUNG; MARC, 2016) .

ambientes não estruturados : constituídos por ambientes que não foram modificados especificamente para facilitar a execução de uma tarefa por um robô (OLIVER; JEAHEUNG; MARC, 2016) .

analito : Dada uma amostra, analito é uma substância ou componente químico que é alvo de análise em um ensaio.

API : A *Application Programming Interface* (API) é uma forma de integrar sistemas, possibilitando benefícios como a segurança dos dados e facilidade na comunicação. A API viabiliza o acesso às funcionalidades de um sistema por aplicativos que não desejam se envolver nos detalhes de implementação.

canal iônico : Canal em forma de poro presente na membrana celular que permite a passagem de íons entre o meio extra e intra celular.

CLR : Os programas criados para o .NET Framework da Microsoft executam em um ambiente de software (em contraste com um ambiente de hardware) denominado *Common Language Runtime* (CLR), uma máquina virtual de aplicativos que fornece serviços como segurança, gerenciamento de memória e tratamento de exceções .

corpo de cogumelo : Canal em forma de poro presente na membrana celular que permite a passagem de íons entre o meio extra e intra celular.

cromatógrafo : é um instrumento analítico que permite a análise dos diversos compostos de uma amostra. A análise ocorre por separação da mistura em várias partes e posterior comparação do resultado com valores previamente conhecidos .

dessatura : As moléculas químicas persistem e saturam o sensor. Assim, após a fase de aquisição, o sensor necessita recuperar sua capacidade, aguardando eliminação das partículas.

espectrômetro : dispositivo utilizado para análise de substâncias, com base na produção e interpretação de seus espectros de emissão ou absorção de radiações eletromagnéticas. Em geral, sua utilização em um nariz artificial envolve combinação com cromatógrafo de gás. O cromatógrafo separa uma mistura odorante em suas partes constituintes e o espectrômetro de massa gera um espectro de massa para cada pico presente na amostra (NAGLE; GUTIERREZ-OSUNA; SCHIFFMAN, 1998).

GitHub : Plataforma de desenvolvimento para: hospedar e revisar código; gerenciar projetos e desenvolvimento de software em equipe. Nesta pesquisa, o GitHub hospeda, no endereço <github.com/helioaz/OntPerception>, todos os itens de configuração associados à ontologia *OntPerception*.

JNI : *Java Native Interface* (JNI) é uma interface de programação para escrever métodos nativos em Java e incorporar a máquina virtual Java em aplicativos nativos .

JVM : *Java Virtual Machine* (JVM) implementa a máquina virtual que executa *bytecodes* gerados pela compilação de código Java. A utilização de *bytecodes* aumenta a portabilidade do executável mas pode gerar perda de desempenho durante a execução.

limiar de discriminação : Representa o menor incremento detectável acima de qualquer intensidade inicial. Exemplo: ao visualizar um flash de luz durante 4seg, somente um flash maior que 4,8s (20% superior a intensidade inicial) terá seu aumento de duração percebido por um humano.

odorante : substância química capturada pelo quimiorreceptor, processada pelo cérebro e interpretada como um odor específico.

OWL : *Web Ontology Language* (OWL) representa um esforço da W3C para contornar as limitações da linguagem RDF quando aplicada em Web Semântica. A OWL é definida por três níveis de complexidade computacional: OWL-Full, OWL-DL e OWL-Lite. A OWL-DL (Description Logic) é mais expressiva que lógica proposicional mas, menos expressiva que lógica de predicados (lógica de primeira ordem) utilizada pela OWL-Full. A vantagem de DLs é serem usualmente decidíveis viabilizando seu processamento de forma eficiente (ANTONIOU *et al.*, 2012). Atualmente a lógica da descrição tornou-se um pedra angular da Web Semântica pelo seu uso no projeto de ontologias.

QTC : A *Quantum Tunneling Composite* (QTC) foi desenvolvida pelo *MIT Media Lab*. Esse composto sintético tira proveito de propriedades em nanoescala criando um material que é inteiramente “sensorial”. Sua resistência elétrica é alterada em função da intensidade da força aplicada. Trata-se de um material flexível podendo ser produzido em qualquer formato desejado .

RDF : *Resource Description Framework* (RDF) é essencialmente um modelo leve de dados baseado em grafos que permite o uso de URI para mapear relacionamentos entre nós presentes na Web. Seu bloco de construção básico é chamando *statement*. Um *statement* é representado por uma tripla $\langle \textit{objeto}, \textit{atributo}, \textit{valor} \rangle$ associados respectivamente a um recurso, uma propriedade e um valor. O valor pode ser um recurso ou um literal. O RDF pode ser descrito de diversas formas, conhecidas como serializadoras. Como exemplo de serializações temos: RDFXML, Notation-3 (N3), Turtle, NTriples, RDFa, and RDF/JSON. O modelo RDF é independente de domínio, cabe ao usuário definir sua própria terminologia em uma linguagem de esquema, denominada RDF Schema (RDFS) (ANTONIOU *et al.*, 2012).

regras de produção : As regras de produção consistem de duas partes: uma pré-condição sensorial (declaração *if*) e uma ação (*then*). Se a pré-condição de uma produção corresponder ao estado atual do mundo, a produção será acionada causando o disparo da regra. Esse mecanismo é um componente básico de “sistemas de produção”.

REST : *Representational State Transfer* (REST) é um estilo de arquitetura de software que fornece uma abordagem conveniente e consistente para requisitar e modificar dados. Em um sistema RESTful, recursos são armazenados em uma base de dados, onde, um cliente envia requisições, o servidor executa a ação e produz a resposta. Na API RESTful as ações são requisitadas através de verbos como *POST*, *GET*, *PUT*, e *DELETE*.

saliência : conceito da neurociência associado a qualquer aspecto de um estímulo que, por alguma razão, se sobressai dos demais. A detecção de saliência representa um mecanismo para o aprendizado e a sobrevivência, permitindo que os organismos concentrem seus recursos perceptivos e cognitivos no subconjunto mais pertinente dos dados sensoriais disponíveis.

SPARQL : *Simple Protocol and RDF Query Language* (SPARQL) é uma linguagem de consulta para o RDF recomendada pela W3C amplamente aceita pela comunidade (ANTONIOU *et al.*, 2012).

taxis : (origem Grega) Movimento de um organismo em resposta a um estímulo tal como luz ou presença de comida. Fonte: wikipedia .

triple store : é uma base de dados de grafos utilizada para armazenar e consultar dados altamente interconectados armazenadas como triplas RDF..

umami : Um dos cinco sabores associados ao paladar. A umami está associado ao reconhecimento das substâncias glutamato livre, inosinato e guanilato. Quando em contato com receptores na língua, essas substâncias aumentam a sensação do sabor do alimento, ou seja, o torna mais agradável ao paladar .

UML : A *Unified Modeling Language* (UML) representa uma linguagem para modelar o projeto de software com aplicação relevante em orientação a objetos. Ela é composta por um conjunto de diagramas que permitem padronizar os detalhes da implementação desejada (FOWLER, 2003).

validação : O processo de avaliar o software durante ou ao final do ciclo de desenvolvimento para determinar se ele satisfaz os requisitos especificados pelo cliente. Estamos construindo o produto certo?.

verificação : O processo de avaliar software para determinar se os produtos de uma dada fase de desenvolvimento satisfazem as condições impostas no início da referida fase. Em resumo: Estamos construindo o produto corretamente?.

vibrissa : Qualquer um dos longos pelos duros que crescem ao redor da boca ou em qualquer outro lugar no rosto de muitos mamíferos, usados como órgãos de toque; bigodes.

Web Semântica : Viabiliza que o acesso ao conteúdo da Web seja realizado por máquinas processadoras facilitando a extração de conhecimento. Esse acesso somente pode ocorrer se os dados da Web forem formalmente descritos e relacionados. Essa formalização é alcançada através de um conjunto de recomendações da W3C (W3C, 2018). A força motora da Web Semântica é o pesquisador Tim Berners-Lee, um dos criadores da Web no final dos anos 80 (ANTONIOU *et al.*, 2012).

DETALHES DE MODELOS DO SISTEMA

Figura 33 – Diagrama de classes da API *OntPercept*

erception

A.2 Diagrama detalhado de classes do programa “Co-gArc”

Figura 34 – Diagrama detalhado de classes do programa “CogArc”

Fonte: Elaborada pelo autor.

A.3 *ontPercept* representado em linguagem Manchester

Código-fonte 3 – Código da *OntPercept* em linguagem Manchester

```

1:  #  //////////////////////////////////////////
2:  #  //Header
3:  #  //////////////////////////////////////////
4:  Ontology: <http://example.org/sense>
5:
6:  Import: <http://www.inf.ufrgs.br/phi-group/ontologies/basicPos.owl>
7:  Import: <http://www.inf.ufrgs.br/phi-group/ontologies/cora.owl>
8:
9:  Annotations:
10:      rdfs:comment "ontPercept represents the resources necessary to model the
11:                    robot's perception of the environment."
12:
13:  AnnotationProperty: rdfs:comment
14:
15:  Prefix: : <http://example.org/sense#>
16:  Prefix: basicPos: <http://www.inf.ufrgs.br/phi-group/ontologies/basicPos.owl#>
17:  Prefix: owl: <http://www.w3.org/2002/07/owl#>
18:  Prefix: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
19:  Prefix: rdfs: <http://www.w3.org/2000/01/rdf-schema#>
20:  Prefix: xml: <http://www.w3.org/XML/1998/namespace>
21:  Prefix: xsd: <http://www.w3.org/2001/XMLSchema#>
22:
23:  \#  //////////////////////////////////////////
24:  \#  //Data Type
25:  \#  //////////////////////////////////////////
26:  Datatype: rdf:langString
27:  Datatype: xsd:anyURI
28:  Datatype: xsd:dateTime
29:  Datatype: xsd:double
30:  Datatype: xsd:long
31:  Datatype: xsd:string
32:
33:
34:  \#  //////////////////////////////////////////
35:  \#  //Object Property
36:  \#  //////////////////////////////////////////
37:  ObjectProperty: generateByObject
38:      SubPropertyOf:
39:          owl:topObjectProperty
40:      Characteristics:
41:          Functional
42:      Range:
43:          <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Object>
44:
45:  ObjectProperty: hasCaptureTime
46:      SubPropertyOf:
47:          owl:topObjectProperty
48:      Characteristics:
49:          Functional
50:      Range:
51:          <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#TimePoint>

```

```

52:
53:   ObjectProperty: hasColor
54:     Annotations:
55:       rdfs:comment "defines the object's main color"@en
56:     Characteristics:
57:       Functional
58:     Domain:
59:       <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Object>
60:
61:   ObjectProperty: hasEmotionalState
62:     Characteristics:
63:       Functional
64:     Range:
65:       EmotionalState
66:
67:   ObjectProperty: hasInternalState
68:     SubPropertyOf:
69:       owl:topObjectProperty
70:     Range:
71:       InternalAttribute
72:
73:   ObjectProperty: hasSmellType
74:     SubPropertyOf:
75:       owl:topObjectProperty
76:     Characteristics:
77:       Functional
78:
79:   ObjectProperty: hasSoundType
80:     Characteristics:
81:       Functional
82:     Range:
83:       HearingAttribute
84:
85:   ObjectProperty: hasTasteType
86:     Characteristics:
87:       Functional
88:
89:   ObjectProperty: isMadeOf
90:     Annotations:
91:       rdfs:comment "type of material that composes the object"@en
92:     SubPropertyOf:
93:       owl:topObjectProperty
94:     Characteristics:
95:       Functional
96:     Domain:
97:       <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Object>
98:     Range:
99:       Material
100:
101:   ObjectProperty: isPositionedAt
102:     Annotations:
103:       rdfs:comment "defines the object coordinates"@en
104:     Characteristics:
105:       Functional
106:     Range:
107:       basicPos:CartesianPositionPoint
108:

```

```

109: ObjectProperty: isSenseOf
110:     Characteristics:
111:         Functional
112:     Domain:
113:         RobotPerceptionEvent
114:     Range:
115:         <http://www.inf.ufrgs.br/phi-group/ontologies/cora.owl#Robot>
116:
117: ObjectProperty: owl:topObjectProperty
118:
119:
120:
121: \# //////////////////////////////////////
122: \# //Data Property
123: \# //////////////////////////////////////
124: DataProperty: RGBValue
125:     Domain:
126:         RGBColor
127:     Range:
128:         xsd:double[>= "0.0"^^xsd:double , <= "1.0"^^xsd:double]
129:     SubPropertyOf:
130:         owl:topDataProperty
131:
132: DataProperty: associateURI
133:     Domain:
134:         <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Object>
135:     Range:
136:         xsd:anyURI
137:     SubPropertyOf:
138:         owl:topDataProperty
139:
140: DataProperty: basicPos:cartesianX
141:     SubPropertyOf:
142:         cartesianPosition
143:
144: DataProperty: basicPos:cartesianY
145:     SubPropertyOf:
146:         cartesianPosition
147:
148: DataProperty: basicPos:cartesianZ
149:     SubPropertyOf:
150:         cartesianPosition
151:
152: DataProperty: bitterness
153:     Annotations:
154:         rdfs:comment "One of the four basic taste sensations that is
155: typically induced by acrid and is mediated especially by receptors in
156: taste buds at the back of the tongue."@en
157:     Characteristics:
158:         Functional
159:     SubPropertyOf:
160:         tasteSense
161:
162: DataProperty: blue
163:     SubPropertyOf:
164:         RGBValue

```

```
164: DataProperty: cartesianPosition
165:   Annotations:
166:     rdfs:comment "coordinates associated with cartesian position"
167:   Range:
168:     xsd:double
169:
170: DataProperty: chemicalLevel
171:   Annotations:
172:     rdfs:comment "represents the level of chemical odorant present in
the environment. This level of odorant, along with nine others,
completely identifies a specific odor. See DOI: 10.1371 / journal.pone
.0073289"@en
173:   Characteristics:
174:     Functional
175:   SubPropertyOf:
176:     smellSense
177:
178: DataProperty: decaynedLevel
179:   Annotations:
180:     rdfs:comment "represents the level of decayned odorant present in
the environment. This level of odorant, along with nine others,
completely identifies a specific odor. see DOI: 10.1371 / journal.pone
.0073289"@en
181:   Characteristics:
182:     Functional
183:   SubPropertyOf:
184:     smellSense
185:
186: DataProperty: detail
187:   Annotations:
188:     rdfs:comment "dependent on the identity of the sound, examples: for
the case of the speech, it presents the sentence being pronounced; for
the case of music, it presents the music being performed, for birds, it
presents the type of bird; for barking, there is no detailed information.
"@en
189:   Characteristics:
190:     Functional
191:   Range:
192:     xsd:string
193:   SubPropertyOf:
194:     hearSense
195:
196: DataProperty: fragrantLevel
197:   Annotations:
198:     rdfs:comment "represents the level of fragrante odorant present in
the environment. This level of odorant, along with nine others,
completely identifies a specific odor. see DOI: 10.1371 / journal.pone
.0073289"@en
199:   Characteristics:
200:     Functional
201:   SubPropertyOf:
202:     smellSense
203:
204: DataProperty: fruityLevel
205:   Annotations:
206:     rdfs:comment "represents the level of fruity odorant present in the
environment. This level of odorant, along with nine others, completely
```

```

    identifies a specific odor. see DOI: 10.1371 / journal.pone.0073289"@en
207:   Characteristics:
208:       Functional
209:   SubPropertyOf:
210:       smellSense
211:
212:   DataProperty: generateBy
213:       Range:
214:           xsd:long
215:
216:   DataProperty: green
217:
218:       SubPropertyOf:
219:           RGBValue
220:
221:   DataProperty: hardness
222:       Annotations:
223:           rdfs:comment "Hardness is a measure of how resistant solid matter is
    to various kinds of permanent shape change when a compressive force is
    applied, parameterized from 0 to 1,"@en
224:       Range:
225:           xsd:double[>= "0.0"^^xsd:double , <= "1.0"^^xsd:double]
226:       SubPropertyOf:
227:           touchSense
228:
229:   DataProperty: hearSense
230:       Annotations:
231:           rdfs:comment "The sense of hearing processes sound information
    captured by microphones present in the robot."@en
232:       Domain:
233:           RobotHear
234:       SubPropertyOf:
235:           owl:topDataProperty
236:
237:   DataProperty: lemonLevel
238:       Annotations:
239:           rdfs:comment "represents the level of lemmon odorant present in the
    environment. This level of odorant, along with nine others, completely
    identifies a specific odor. see DOI: 10.1371 / journal.pone.0073289"@en
240:       Characteristics:
241:           Functional
242:       SubPropertyOf:
243:           smellSense
244:
245:   DataProperty: mintyLevel
246:       Annotations:
247:           rdfs:comment "represents the level of minty odorant present in the
    environment. This level of odorant, along with nine others, completely
    identifies a specific odor. see DOI: 10.1371 / journal.pone.0073289"@en
248:       Characteristics:
249:           Functional
250:       SubPropertyOf:
251:           smellSense
252:
253:   DataProperty: moisture
254:       Annotations:

```

```

255:         rdfs:comment "moisture present in the object, parameterized from 0
to 1"@en
256:     Range:
257:         xsd:double[>= "0.0"^^xsd:double , <= "1.0"^^xsd:double]
258:     SubPropertyOf:
259:         touchSense
260:
261:     DataProperty: name
262:         Characteristics:
263:             Functional
264:         Domain:
265:             <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Object>
266:         Range:
267:             xsd:string
268:
269:     DataProperty: objectId
270:         Annotations:
271:             rdfs:comment "Identification of the object being processed. This is
a functional data property, considering that each object has only one id.
"@en
272:         Characteristics:
273:             Functional
274:         Domain:
275:             <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Object>
276:         Range:
277:             xsd:long
278:
279:     DataProperty: occursAt
280:         Annotations:
281:             rdfs:comment "Instant of Data and time such as the perception occurs
."
282:         Characteristics:
283:             Functional
284:         Domain:
285:             <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#TimePoint>
286:         Range:
287:             xsd:dateTime
288:         SubPropertyOf:
289:             owl:topDataProperty
290:
291:     DataProperty: owl:topDataProperty
292:
293:     DataProperty: popcornLevel
294:         Annotations:
295:             rdfs:comment "represents the level of popcorn odorant present in the
environment. This level of odorant, along with nine others, completely
identifies a specific odor. see DOI: 10.1371 / journal.pone.0073289"@en
296:         Characteristics:
297:             Functional
298:         SubPropertyOf:
299:             smellSense
300:
301:     DataProperty: pressure
302:         Annotations:
303:             rdfs:comment "pressure exerted by the object, parameterized from 0
to 1"@en
304:         Range:

```



```

305:         xsd:double[>= "0.0"^^xsd:double , <= "1.0"^^xsd:double]
306:     SubPropertyOf:
307:         touchSense
308:
309:     DataProperty: pungentLevel
310:         Annotations:
311:             rdfs:comment "represents the level of pungent odorant present in the
environment. This level of odorant, along with nine others, completely
identifies a specific odor. see DOI: 10.1371 / journal.pone.0073289"@en
312:         Characteristics:
313:             Functional
314:         SubPropertyOf:
315:             smellSense
316:
317:     DataProperty: red
318:         SubPropertyOf:
319:             RGBValue
320:
321:     DataProperty: roughness
322:         Annotations:
323:             rdfs:comment "roughness level of the object, parameterized from 0 to
1"@en
324:         Range:
325:             xsd:double[>= "0.0"^^xsd:double , <= "1.0"^^xsd:double]
326:         SubPropertyOf:
327:             touchSense
328:
329:     DataProperty: saltiness
330:         Annotations:
331:             rdfs:comment "One of the four basic taste sensations that is
typically induced by NaCl and is mediated especially by receptors in
taste buds at the front lateral of the tongue."@en
332:         Characteristics:
333:             Functional
334:         SubPropertyOf:
335:             tasteSense
336:
337:     DataProperty: smellSense
338:         Domain:
339:             RobotSmell
340:         Range:
341:             xsd:double[>= "0.0"^^xsd:double , <= "1.0"^^xsd:double]
342:         SubPropertyOf:
343:             owl:topDataProperty
344:
345:     DataProperty: sourness
346:         Annotations:
347:             rdfs:comment "One of the four basic taste sensations that is
typically induced chiefly by acids and is mediated especially by
receptors in taste buds at the lateral of the tongue."@en
348:         Characteristics:
349:             Functional
350:         SubPropertyOf:
351:             tasteSense
352:
353:     DataProperty: sweetLevel
354:         Annotations:

```

```

355:         rdfs:comment "represents the level of sweet odorant present in the
environment. This level of odorant, along with nine others, completely
identifies a specific odor. see DOI: 10.1371 / journal.pone.0073289"@en
356:     Characteristics:
357:         Functional
358:     SubPropertyOf:
359:         smellSense
360:
361: DataProperty: sweetness
362:     Annotations:
363:         rdfs:comment "One of the four basic taste sensations that is
typically induced by disaccharides and is mediated especially by
receptors in taste buds at the front of the tongue."@en
364:     Characteristics:
365:         Functional
366:     SubPropertyOf:
367:         tasteSense
368:
369: DataProperty: tag
370:     Annotations:
371:         rdfs:comment "a word giving the main characteristic of someone or
something"@en
372:     Characteristics:
373:         Functional
374:     Domain:
375:         <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Object>
376:     Range:
377:         xsd:string
378:
379: DataProperty: tasteSense
380:     Domain:
381:         RobotTaste
382:     Range:
383:         xsd:double[>= "0.0"^^xsd:double , <= "1.0"^^xsd:double]
384:     SubPropertyOf:
385:         owl:topDataProperty
386:
387: DataProperty: temperature
388:     Annotations:
389:         rdfs:comment "temperature in Celsius degrees"@en
390:     Range:
391:         xsd:double[>= "-100.0"^^xsd:double , <= "1000.0"^^xsd:double]
392:     SubPropertyOf:
393:         touchSense
394:
395: DataProperty: touchSense
396:     Domain:
397:         RobotTouch
398:     SubPropertyOf:
399:         owl:topDataProperty
400:
401: DataProperty: type
402:     Annotations:
403:         rdfs:comment "Defines the type of object observed by the robot,
examples: mug, person, door, cookie, pan, etc."@en
404:     Characteristics:
405:         Functional

```

```

406:     Domain:
407:         <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Object>
408:     Range:
409:         xsd:string
410:     SubPropertyOf:
411:         visionSense
412:
413:     DataProperty: umami
414:     Annotations:
415:         rdfs:comment "a taste sensation that is meaty or savory and is
produced by several amino acids and nucleotides (such as glutamate and
aspartate)"@en
416:     Characteristics:
417:         Functional
418:     SubPropertyOf:
419:         tasteSense
420:
421:     DataProperty: visionSense
422:     Annotations:
423:         rdfs:comment "The sense of view processes the information captured
by cameras in the robot."@en
424:     Domain:
425:         RobotVision
426:
427:     DataProperty: volume
428:     Annotations:
429:         rdfs:comment "Refers to the auditory sensation perceived by the
listener. This sensation depends on the sound frequency and intensity,
parameterized from 0 to 1."@en
430:     Characteristics:
431:         Functional
432:     Range:
433:         xsd:double[>= "0.0"^^xsd:double , <= "1.0"^^xsd:double]
434:     SubPropertyOf:
435:         hearSense
436:
437:     DataProperty: woodyLevel
438:     Characteristics:
439:         Functional
440:     SubPropertyOf:
441:         smellSense
442:
443:
444:
445:
446: \# //////////////////////////////////////
447: \# // Class
448: \# //////////////////////////////////////
449: Class: <http://www.inf.ufrgs.br/phi-group/ontologies/cora.owl#Robot>
450:     SubClassOf:
451:         hasEmotionalState some EmotionalState
452:
453: Class: <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Abstract>
454:
455: Class: <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Agent>
456:     Annotations:

```

```

457:         rdfs:comment "Something or someone that can act on its own and
           produce changes in the world."@en
458:
459: Class: <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Attribute>
460:
461: Class: <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Object>
462:     Annotations:
463:         rdfs:comment "Corresponds roughly to the class of ordinary objects."
           @en
464:     SubClassOf:
465:         objectId exactly 1 xsd:long
466:     HasKey:
467:         objectId
468:
469: Class: <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#TimePoint>
470:
471: Class: BiologicalAttribute
472:     SubClassOf:
473:         InternalAttribute
474:     DisjointWith:
475:         PerceptualAttribute
476:
477: Class: CognitiveAgent
478:     Annotations:
479:         rdfs:comment "A CognitiveAgent is an Agent that has the ability to
           reason, deliberate, make plans, and experience emotions. Although Human
           is a subclass of CognitiveAgent, there may be instances of CognitiveAgent
           which are not also instances of Human. For example, Primates, dolphins,
           whales, and some extraterrestrials (if they exist) might be considered
           CognitiveAgents."@en
480:     SubClassOf:
481:         SentientAgent
482:
483: Class: ColorAttribute
484:     SubClassOf:
485:         InternalAttribute
486:
487: Class: EmotionalState
488:     Annotations:
489:         rdfs:comment "A emotional state associated with the object."
490:     SubClassOf:
491:         StateofMind
492:
493: Class: HearingAttribute
494:     SubClassOf:
495:         PerceptualAttribute
496:
497: Class: Human
498:     Annotations:
499:         rdfs:comment "Modern man, the only remaining species of the Homo
           genus."@en
500:     SubClassOf:
501:         CognitiveAgent,
502:         hasEmotionalState some EmotionalState
503:
504: Class: InternalAttribute
505:     Annotations:

```

```

506:         rdfs:comment "Any Attribute whose presence is detected by an act of
           Perception."@en
507:     SubClassOf:
508:         <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Attribute>
509:
510:     Class: Material
511:     SubClassOf:
512:         InternalAttribute
513:
514:     Class: PerceptualAttribute
515:     SubClassOf:
516:         InternalAttribute
517:     DisjointWith:
518:         BiologicalAttribute
519:
520:     Class: PhysicalState
521:     SubClassOf:
522:         InternalAttribute
523:
524:     Class: PsychologicalAttribute
525:     SubClassOf:
526:         BiologicalAttribute
527:
528:     Class: RGBColor
529:     SubClassOf:
530:         ColorAttribute,
531:         blue exactly 1 xsd:double,
532:         green exactly 1 xsd:double,
533:         red exactly 1 xsd:double
534:
535:     Class: RobotHear
536:     Annotations:
537:         rdfs:comment "The subclass of RobotPerception in which the sensing
           is done by an auditory sensor."@en
538:     EquivalentTo:
539:         RobotPerceptionEvent
540:         and (hasSoundType exactly 1 HearingAttribute)
541:     SubClassOf:
542:         RobotPerceptionEvent,
543:         isPositionedAt some basicPos:CartesianPositionPoint
544:
545:     Class: RobotPerceptionEvent
546:     Annotations:
547:         rdfs:comment "Represents the robot's perception about the
           environment."
548:     SubClassOf:
549:         <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Abstract>,
550:         hasCaptureTime exactly 1 <http://www.inf.ufrgs.br/phi-group/
           ontologies/sumo.owl#TimePoint>,
551:         isSenseOf exactly 1 <http://www.inf.ufrgs.br/phi-group/ontologies/
           cora.owl#Robot>,
552:         generateBy exactly 1 xsd:long
553:
554:     Class: RobotSmell
555:     Annotations:
556:         rdfs:comment "The subclass of RobotPerception in which the sensing
           is done by an olfactory sensor."@en

```

```

557:     SubClassOf:
558:         RobotPerceptionEvent,
559:         isPositionedAt some basicPos:CartesianPositionPoint
560:
561: Class: RobotTaste
562:     Annotations:
563:         rdfs:comment "The subclass of RobotPerception in which the sensing
564:         is done by of a sensor which can discriminate various tastes."@en
565:     SubClassOf:
566:         RobotPerceptionEvent
567:
568: Class: RobotTouch
569:     Annotations:
570:         rdfs:comment "The subclass of RobotPerception in which the sensing
571:         is done by an haptic technology."@en
572:     SubClassOf:
573:         RobotPerceptionEvent,
574:         isPositionedAt some basicPos:CartesianPositionPoint
575:
576: Class: RobotVision
577:     Annotations:
578:         rdfs:comment "The subclass of RobotPerception in which the sensing
579:         is done by a vision sensor."@en
580:     SubClassOf:
581:         RobotPerceptionEvent
582:
583: Class: SentientAgent
584:     Annotations:
585:         rdfs:comment "A SentientAgent is an Agent that is capable of
586:         Perception and experiences some level of consciousness (see
587:         ConsciousnessAttribute). If the Agent is able to reason at a
588:         comparatively high level (evinced by the ability to learn, plan, and feel
589:         emotions), then it is also an instance of CognitiveAgent. A
590:         DomesticAnimal is a SentientAgent, but may or may not be a CognitiveAgent
591:         , depending on the type of Animal."@en
592:     SubClassOf:
593:         <http://www.inf.ufrgs.br/phi-group/ontologies/sumo.owl#Agent>
594:
595: Class: StateofMind
596:     SubClassOf:
597:         PsychologicalAttribute
598:
599: Class: basicPos:CartesianPositionPoint
600:     Annotations:
601:         rdfs:comment "A CartesianPositionPoint denotes the quantitative
602:         position of an object in a cartesian coordinate system."@en
603:
604: \# //////////////////////////////////////
605: \# //Individual
606: \# //////////////////////////////////////
607: Individual: angerEmotion
608:     Annotations:
609:         rdfs:comment "Anger is a negative emotion, characterised by feelings
610:         of unpleasantness and high arousal, in the form of antagonistic feelings
611:         and action tendencies. [Source: OCEAS]"@en

```

```
602:     Types:
603:         EmotionalState
604:
605: Individual: barkingSound
606:     Types:
607:         HearingAttribute
608:
609: Individual: bellSound
610:     Types:
611:         HearingAttribute
612:
613: Individual: birdsSound
614:     Types:
615:         HearingAttribute
616:
617: Individual: closeState
618:     Types:
619:         PhysicalState
620:
621: Individual: clothMaterial
622:     Types:
623:         Material
624:
625: Individual: disgustEmotion
626:     Annotations:
627:         rdfs:comment "A negative emotion which guards the body against
        dangerous foods and microbial infections. The basic physiological disgust
        has also culturally evolved to incorporate moral disgust which protects
        the self from nonphysical threats such as moral violations. [Source:
        OCEAS]"@en
628:     Types:
629:         EmotionalState
630:
631: Individual: fearEmotion
632:     Annotations:
633:         rdfs:comment "An activated, aversive emotion that motivates attempts
        to cope with events that provide threats to the survival or well-being
        of organisms. Characterised by feelings of threat and impending doom, and
        by an urge to get out of the situation. [Source: OCEAS]"@en
634:     Types:
635:         EmotionalState
636:
637: Individual: glassMaterial
638:     Types:
639:         Material
640:
641: Individual: happinessEmotion
642:     Annotations:
643:         rdfs:comment "A positive emotion which is experienced in reaction to
        a positive experience or event. [Source: OCEAS]"@en
644:     Types:
645:         EmotionalState
646:
647: Individual: humanVoiceSound
648:     Types:
649:         HearingAttribute
650:
```

```
651: Individual: liquidFlowingSound
652:   Types:
653:     HearingAttribute
654:
655: Individual: metalMaterial
656:   Types:
657:     Material
658:
659: Individual: motorSound
660:   Types:
661:     HearingAttribute
662:
663: Individual: musicSound
664:   Types:
665:     HearingAttribute
666:
667: Individual: neutralEmotion
668:   Types:
669:     EmotionalState
670:
671: Individual: noneState
672:   Types:
673:     PhysicalState
674:
675: Individual: offState
676:   Types:
677:     PhysicalState
678:
679: Individual: onState
680:   Types:
681:     PhysicalState
682:
683: Individual: openState
684:   Types:
685:     PhysicalState
686:
687: Individual: organicMaterial
688:   Types:
689:     Material
690:
691: Individual: paperMaterial
692:   Types:
693:     Material
694:
695: Individual: plasticMaterial
696:   Types:
697:     Material
698:
699: Individual: robotVoiceSound
700:   Types:
701:     HearingAttribute
702:
703: Individual: rockMaterial
704:   Types:
705:     Material
706:
707: Individual: rubberMaterial
```

```

708:     Types:
709:         Material
710:
711: Individual: sadnessEmotion
712:     Annotations:
713:         rdfs:comment "A negative emotion felt when an event is appraised as
unpleasant, obstructive to one's goals and concerns, and one feels unable
to cope with it or modify it. [Source: OCEAS]"@en
714:     Types:
715:         EmotionalState
716:
717: Individual: surpriseEmotion
718:     Annotations:
719:         rdfs:comment "Surprise is a neutral emotion caused by encountering
unexpected events of all kinds. May be experienced as negative if the
surprising event is evaluated as negative, or positive if the surprising
event is evaluated as positive. [Source: OCEAS]"@en
720:     Types:
721:         EmotionalState
722:
723: Individual: tvSound
724:     Types:
725:         HearingAttribute
726:
727: Individual: unknownMaterial
728:     Types:
729:         Material
730:
731: Individual: unknownSound
732:     Types:
733:         HearingAttribute
734:
735: Individual: woodMaterial
736:     Types:
737:         Material
738:
739: \# //////////////////////////////////////
740: \# //Axiom
741: \# //////////////////////////////////////
742: DisjointClasses:
743:     RobotHear,RobotSmell,RobotTaste,RobotTouch,RobotVision

```

A.4 Soar: proposição de operadores

Código-fonte 4 – Código de proposição de operadores para o Soar

```

1: #####0
2: # Propose*cmdReception
3: # If the human gives an task and step equal 0,
4: # then propose process new command
5: sp {Propose*cmdReception
6: (state <s> ^name butler
7:     ^io.input-link <i1>
8:     ^robot <r1>)
9: (<i1> ^action.task <> null)
10: (<r1> ^step 0)
11: -->
12:     (<s> ^operator <o> + =)
13:     (<o> ^name cmdReception)
14: }
15:
16:
17:
18: #####1
19: # Propose*getKitchenCracker
20: # If the task is Get and the object is Cracker and empty hands and step
    equal 11,
21: # then propose move to Kitchen.
22: sp {Propose*getKitchenCracker
23: (state <s> ^name butler ^robot <r1>)
24: (<r1> ^task get ^object Cracker ^step 11 )
25: -->
26:     (<s> ^operator <o> + =)
27:     (<o> ^name getKitchenCracker)
28: }
29:
30:
31: # Propose*getKitchenCracker*remove
32: # If the task is Get and the object is Cracker and step equal 12 and
33: # the command is complete
34: # then propose remove move.
35: sp {Propose*getKitchenCracker*remove
36: (state <s> ^name butler ^robot <r1> ^io.output-link <output>)
37: (<r1> ^task get ^object Cracker ^step 12 )
38: (<output> ^Move <move>)
39: (<move> ^status complete)
40: -->
41:     (<s> ^operator <o> + =)
42:     (<o> ^name getKitchenCracker*remove)
43: }
44:
45:
46:
47: #####2
48: # ontPercept Project
49: # 03/2018
50: # Phase 2x - lookForCracker
51:

```

```

52:
53: # Propose*lookForCracker
54: # If   the task is Get and the object is Cracker and
55: #     the result of the last command is success and
56: #     the step is 21
57: # then propose lookForCracker.
58: sp {Propose*lookForCracker
59:   (state <s> ^name butler ^robot <r1>)
60:   (<r1> ^task get ^object Cracker ^step 21 )
61:   (<r1> ^lastCmdResult success )
62: -->
63:   (<s> ^operator <o> + =)
64:   (<o> ^name lookForCracker)
65: }
66:
67:
68: # Propose*lookForCracker*remove
69: # If   the task is Get and the object is Cracker and the step is 22 and
70: #     it saw a Cracker and the the command is complete
71: # then propose remove lookForCracker.
72: sp {Propose*lookForCracker*remove
73:   (state <s> ^name butler ^io.output-link <output> ^robot <r1>
74:     ^io.input-link.vision <vision>)
75:   (<r1> ^task get ^object Cracker ^step 22 )
76:   (<vision> ^(<thing>.tagInfo Cracker)
77:   (<output> ^LookFor <cmd>)
78:   (<cmd> ^status complete)
79: -->
80:   (<s> ^operator <o> + =)
81:   (<o> ^name lookForCracker*remove)
82: }
83:
84:
85:
86: #####3
87: # Propose*moveToCracker
88: # If   the task is Get and the object is Cracker and
89: #     the result of the last command is success and
90: #     the step is 31
91: # then propose moveToCracker.
92: sp {Propose*moveToCracker
93:   (state <s> ^name butler ^robot <r1>)
94:   (<r1> ^task get ^object Cracker ^step 31 )
95:   (<r1> ^lastCmdResult success )
96: -->
97:   (<s> ^operator <o> + =)
98:   (<o> ^name moveToCracker)
99: }
100:
101:
102: # Propose*moveToCracker*remove
103: # If   the task is Get and the object is Cracker and
104: #     the command is complete
105: #     the step is 32
106: # then propose remove moveToCracker.
107: sp {Propose*moveToCracker*remove
108:   (state <s> ^name butler ^robot <r1> ^io.output-link <output>)

```

```

109:    (<r1> ^task get ^object Cracker ^step 32 )
110:    (<output> ^MoveID <cmd>)
111:    (<cmd> ^status complete)
112: -->
113:    (<s> ^operator <o> + =)
114:    (<o> ^name moveToCracker*remove)
115: }
116:
117:
118:
119: #####4
120: # Propose*takeCracker
121: # If   the task is Get and the object is Cracker and
122: #     the result of the last command is success and
123: #     the step is 41
124: # then propose takeCracker.
125: sp {Propose*takeCracker
126:    (state <s> ^name butler ^robot <r1>)
127:    (<r1> ^task get ^object Cracker ^step 41 )
128:    (<r1> ^lastCmdResult success )
129: -->
130:    (<s> ^operator <o> + =)
131:    (<o> ^name takeCracker)
132: }
133:
134:
135: # Propose*takeCracker*remove
136: # If   the task is Get and the object is Cracker and
137: #     the step is 42
138: # then propose remove takeCracker.
139: sp {Propose*takeCracker*remove
140:    (state <s> ^name butler ^robot <r1> ^io.output-link <output>)
141:    (<r1> ^task get ^object Cracker ^step 42 )
142:    (<output> ^TakeRight <cmd>)
143:    (<cmd> ^status complete)
144: -->
145:    (<s> ^operator <o> + =)
146:    (<o> ^name takeCracker*remove)
147: }
148:
149:
150: # Propose*headReset
151: # If   the task is Get and the object is Cracker and
152: #     the result of the last command is success and
153: #     the step is 43
154: # then propose headReset.
155: sp {Propose*headReset
156:    (state <s> ^name butler ^robot <r1>)
157:    (<r1> ^task get ^object Cracker ^step 43 )
158:    (<r1> ^lastCmdResult success )
159: -->
160:    (<s> ^operator <o> + =)
161:    (<o> ^name headReset)
162: }
163:
164:
165: # Propose*headReset*remove

```

```

166: # If    the task is Get and the object is Cracker and
167: #       the step is 44
168: # then propose remove headReset.
169: sp {Propose*headReset*remove
170:   (state <s> ^name butler ^robot <r1> ^io.output-link <output>)
171:   (<r1> ^task get ^object Cracker ^step 44 )
172:   (<output> ^HeadReset <cmd>)
173:   (<cmd> ^status complete)
174: -->
175:   (<s> ^operator <o> + =)
176:   (<o> ^name headReset*remove)
177: }
178:
179:
180: # Apply*headReset*remove:
181: # If the operator is headReset,
182: # and there is a completed head reset command on the output link,
183: # then remove that command and go to next phase: moveLivingRoom
184: sp {Apply*headReset*remove
185:   (state <s> ^name butler ^operator <o>
186:     ^io.output-link <output> ^robot <r1>)
187:   (<o> ^name headReset*remove )
188:   (<r1> ^lastCmdResult inProgress ^step 44)
189:   (<output> ^HeadReset <cmd>)
190:   (<cmd> ^status complete)
191: -->
192:   (<r1> ^step 51)
193:   (<r1> ^step 44 -)
194:   (<r1> ^lastCmdResult success)
195:   (<r1> ^lastCmdResult inProgress - )
196:   (<output> ^HeadReset <cmd> -)}
197:
198:
199:
200: #####5
201: # Propose*moveLivingRoom
202: # If    the task is Get and the object is Cracker and
203: #       the result of the last command is success and
204: #       the step is 51
205: # then propose moveLivingRoom.
206: sp {Propose*moveLivingRoom
207:   (state <s> ^name butler ^robot <r1>)
208:   (<r1> ^task get ^object Cracker ^step 51 )
209:   (<r1> ^lastCmdResult success )
210: -->
211:   (<s> ^operator <o> + )
212:   (<o> ^name moveLivingRoom) }
213:
214:
215: # Propose*moveLivingRoom*remove
216: # If    the task is Get and the object is Cracker and
217: #       the command is complete
218: #       the step is 52
219: # then propose remove moveLivingRoom.
220: sp {Propose*moveLivingRoom*remove
221:   (state <s> ^name butler ^robot <r1> ^io.output-link <output> )
222:   (<r1> ^task get ^object Cracker ^step 52 )

```

```

223:         (<output> ^Move <cmd>)
224:         (<cmd> ^status complete)
225: -->
226:         (<s> ^operator <o> + =)
227:         (<o> ^name moveLivingRoom*remove)
228:     }
229:
230:
231:
232: #####6
233: # Propose*lookForHuman
234: # If   the task is Get and the object is Cracker and
235: #      the result of the last command is success and
236: #      the step is 61
237: # then propose lookForHuman.
238: sp {Propose*lookForHuman
239:   (state <s> ^name butler ^robot <r1>)
240:   (<r1> ^task get ^object Cracker ^step 61 )
241:   (<r1> ^lastCmdResult success )
242: -->
243:   (<s> ^operator <o> + =)
244:   (<o> ^name lookForHuman)
245: }
246:
247:
248: # Propose*lookForHuman*remove
249: # If   the task is Get and the object is Cracker and the step is 62
250: #      it saw a Human and the command is complete
251: #
252: # then propose remove moveToCracker.
253: sp {Propose*lookForHuman*remove
254:   (state <s> ^name butler ^robot <r1> ^io.output-link <output>
255:     ^io.input-link.vision <vision> )
256:   (<r1> ^task get ^object Cracker ^step 62 )
257:   (<vision> ^<thing>.tagInfo Human)
258:   (<output> ^LookFor <cmd>)
259:   (<cmd> ^status complete)
260: -->
261:   (<s> ^operator <o> + =)
262:   (<o> ^name lookForHuman*remove)
263: }
264:
265:
266:
267: #####7
268: # Propose*moveToHuman
269: # If   the task is Get and the object is Cracker and
270: #      the result of the last command is success and
271: #      the step is 71
272: # then propose moveToHuman.
273: sp {Propose*moveToHuman
274:   (state <s> ^name butler ^robot <r1>)
275:   (<r1> ^task get ^object Cracker ^step 71 )
276:   (<r1> ^lastCmdResult success )
277: -->
278:   (<s> ^operator <o> + =)
279:   (<o> ^name moveToHuman)

```

```

280: }
281:
282:
283: # Propose*moveToHuman*remove
284: # If   the task is Get and the object is Cracker and
285: #     the command is complete
286: #     the step is 72
287: # then propose remove moveToHuman.
288: sp {Propose*moveToHuman*remove
289:   (state <s> ^name butler ^robot <r1> ^io.output-link <output> )
290:   (<r1> ^task get ^object Cracker ^step 72 )
291:   (<output> ^MoveID <cmd>)
292:   (<cmd> ^status complete)
293: -->
294:   (<s> ^operator <o> + =)
295:   (<o> ^name moveToHuman*remove)
296: }
297:
298:
299:
300: #####8
301: # Propose*deliveryCracker
302: # If   the task is Get and the object is Cracker and
303: #     the result of the last command is success and
304: #     the step is 81
305: # then propose deliveryCracker.
306: sp {Propose*deliveryCracker
307:   (state <s> ^name butler ^robot <r1>)
308:   (<r1> ^task get ^object Cracker ^step 81 )
309:   (<r1> ^lastCmdResult success )
310: -->
311:   (<s> ^operator <o> + )
312:   (<o> ^name deliveryCracker)
313: }
314:
315:
316: # Propose*deliveryCracker*remove
317: # If   the task is Get and the object is Cracker and
318: #     the command is complete
319: #     the step is 82
320: # then propose remove deliveryCracker.
321: sp {Propose*deliveryCracker*remove
322:   (state <s> ^name butler ^robot <r1> ^io.output-link <output>)
323:   (<r1> ^task get ^object Cracker ^step 82 )
324:   (<output> ^Speech <cmd>)
325:   (<cmd> ^status complete)
326: -->
327:   (<s> ^operator <o> + =)
328:   (<o> ^name deliveryCracker*remove)
329: }

```

CRÍTICAS À WEB SEMÂNTICA

Nos dias 10 e 11 de outubro de 2009, um dos criadores da *Basic Formal Ontology* (BFO), o filósofo Barry Smith da *University at Buffalo* ministrou o curso *An Introduction to Ontology: From Aristotle to the Universal Core* com oito aulas (SMITH, 2009).

Na sexta aula, intitulada *Ontology and the Semantic Web* foram apresentados 5 problemas graves que podem comprometer a evolução da [Web Semântica](#). Naturalmente, essa é uma visão filosófica do tema que deve ser analisada com cuidado. A expansão da [Web Semântica](#) mostra a viabilidade da proposta. A visão do Professor Barry deve ser interpretada como um alerta, além de ressaltar a necessidade de fortalecer as organizações que buscam padrões para a área. A seguir são listados os pontos levantados na apresentação.

Problem 1: People Lie

- *Meta utopia is a world of reliable metadata*
- *But poisoning the well can confer benefits to the poisoners*
- *Metadata exists in a competitive world*
- *Some people are crooks (crook: a very dishonest person, especially a criminal or a cheat)*
- *Some people are cranks (A crank will usually maintain his viewpoint despite — or perhaps because of — evidence to the contrary.)*
- *Some people are French philosophers*
- *Who will policy the coding?*

Problem 2: People are lazy

- *How many pages on the web are titled: "Please title this page"*

Problem 3: People are stupid

- *The vast majority of the Internet's users (even those who are native speakers of English) cannot spell or punctuate*
- *Will internet users learn to accurately tag their information with whatever DL-hierarchy they are supposing to be using?*

Problem 4: Multiple descriptions

- *"Requiring everyone to use the same vocabulary denudes the cognitive landscape, enforces homogeneity in ideas"*

Problem 5: Ontology Impedance

- *Represents the semantic mismatch between ontologies being merged*
 - *Solution 1: treat it as inevitable, and learn to find ways to cope with the disturbance which it brings*
 - *Solution 2: resolve the impedance problem by hand on a case-by-case basis*
- *Both these solutions fails:*
 - *Problem with solution 1: treating mismatches as "impedance" ignores the problem of error propagation*
 - *Problem with solution 2: resolving impedance in a case-by-case basis defeats the very purpose of the semantic web: "it must resolve the problems automatically on the fly"*