

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

JÚLIA KIKUYE KAMBARA DA SILVA

**Instanciação de Atividades de Atuação em um Modelo de Processo de
Negócio e Seleção de Serviços Web em Ambientes de *Home Care***

Dissertação apresentada como requisito parcial para a
obtenção do grau de Mestre em Ciência da
Computação.

Orientador: Prof. Dr. Leandro Krug Wives

Porto Alegre
2015

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Kambara-Silva, Júlia Kikuye

Instanciação de Atividades de Atuação em um Modelo de Processo de Negócio e Seleção de Serviços Web em Ambientes de *Home Care* [manuscrito] / Júlia Kikuye Kambara da Silva. – 2015.

161 f.:il.

Orientador: Leandro Krug Wives.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2014.

1.Serviços Web. 2.Seleção. 3.Instanciação. 4. Sensibilidade ao contexto. 5. Home care. I. Wives, Leandro Krug. II. Instanciação de Atividades de Atuação em um Modelo de Processo de Negócio e Seleção de Serviços Web em Ambientes de *Home Care*.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do PPGC: Prof. Luigi Carro

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço à minha mãe, por ter me apoiado, por nunca me deixar desistir e ter estado sempre ao meu lado.

Agradeço ao restante da minha família e aos amigos queridos, pelo carinho, compreensão, amizade e companheirismo, por estarem sempre dispostos a me ouvir, ajudar e sorrir.

Agradeço ao meu namorado pelo apoio constante, que me acompanhou desde o começo do curso de graduação.

Agradeço aos colegas do laboratório 213, em especial: Guilherme, Diego, Jonas, Carlos, pelas discussões construtivas, momentos de descontração, e companheirismo.

Agradeço ao meu orientador, Prof. Leandro Krug Wives, pelo apoio, orientação e compreensão, com que me guiou no desenvolvimento deste trabalho.

Agradeço à UFRGS e todos os professores e técnicos administrativos que me auxiliaram durante as aulas ou durante problemas administrativos.

RESUMO

Atualmente existem muitos estudos sendo conduzidos na área de *Home Care*, onde casas providas com sensores e atuadores podem ajudar o paciente em suas tarefas diárias, mesmo aqueles portadores de doenças crônicas ou limitações cognitivas e físicas. No contexto de *Home Care*, dispositivos domésticos controláveis podem agir como atuadores. Um importante desafio nessa área é a coordenação desses atuadores, para que os mesmos possam ser utilizados na prestação de cuidados ao paciente em sua casa. Para prestar auxílio em tarefas, deve ser levado em consideração o contexto do paciente e dos dispositivos presentes na casa. Este trabalho apresenta uma nova abordagem, sensível ao contexto, para selecionar a funcionalidade, de um dispositivo, mais adequada para a execução de uma atuação em um ambiente de *Home Care*. Nessa abordagem, as funcionalidades dos dispositivos são representadas por serviços Web, e modelos de processos de negócio são utilizados como roteiros das atividades que devem ser realizadas para gerir uma determinada situação. Desta forma, o problema de selecionar dispositivos e funcionalidades adequadas para executar uma determinada atividade de atuação de um modelo de processo de negócio é tratado através da seleção de serviços Web com base em aspectos sintáticos, semânticos e contextuais. A abordagem é avaliada através da simulação de cenários de aplicação extraídos da literatura, de forma a verificar a sua viabilidade.

Palavras-chave: Serviços Web; Seleção; Instanciação; Sensibilidade ao Contexto; *Home Care*.

Instantiation of Actuation Activities in a Business Process Model and Selection of Web Services in Home Care Environments

ABSTRACT

There are many studies currently being conducted within the field of Home Care, where houses fulfilled with actuators and sensors can help patients in their daily lives, even the ones who suffer from chronicle diseases or cognitive and physical disabilities. In the Home Care context, controllable domestic devices can perform as actuators. An important challenge to this area refers to the coordination of such actuators so they can be used to assist patients in their home. In order to assist the patient daily live we need to take into consideration the patient and the context of household devices. This work presents a novel context-aware approach for selecting the most appropriate device functionality to execute the actuation needed in a Home Care environment. In this approach, device functionalities are described and represented by Web services, and business process models are used as guidelines for the activities that should be realized to manage a determined situation. Therefore, the issue of selecting devices and adequate functionalities to execute determined activity in a business process model is treated as an approach to select Web services based on their syntactic and semantic aspects in a context-aware manner. The approach is evaluated by simulation of application scenarios extracted from the literature in order to check its viability.

Keywords: Web Service; Selection; Instantiation; Context-Awareness; Home Care.

LISTA DE FIGURAS

Figura 2.1 – Arquitetura padrão do serviço Web	31
Figura 3.1 – Visão geral do modelo para aplicações <i>Home Care</i>	49
Figura 3.2 – Modelo de Pessoa.....	50
Figura 3.3 – Modelo de Organização	52
Figura 3.4 – Modelo de Localização	53
Figura 3.5 – Modelo do Dispositivo.....	54
Figura 3.6 – Exemplo de funcionalidade com características pré-determinadas	55
Figura 3.7 – Modelo de Controlador	56
Figura 3.8 – Ações de Atuação são executadas por algum tipo de funcionalidade do dispositivo.....	57
Figura 3.9 – Conceitos folhas das Ações de Atuação possuem informação agregada.....	58
Figura 3.10 – Modelo de PN sem a utilização de tripla no <i>label</i> das atividades.....	59
Figura 3.11 – Modelo de PN com a utilização de tripla no <i>label</i> das atividades	60
Figura 3.12 – Classificação de sujeito, ação e objeto.....	61
Figura 3.13 – Framework proposto para objeto-único com ação referente a (uma) Pessoa.....	64
Figura 3.14 – Framework proposto para objeto-único com ação referente a (um) Dispositivo	65
Figura 3.15 – Framework proposto para multiobjeto com ação referente a Dispositivos.....	66
Figura 4.1 – Modelo do PN para tratar do estado de agitação do paciente (em BPMN 2.0) ...	70
Figura 4.2 – Modelo do PN para tratar quando se esquece o fogão ligado (em BPMN 2.0) ...	72
Figura 4.3 – Modelo do PN para tratar quando a pessoa entra em um cômodo (em BPMN 2.0)	73
Figura 4.4 – Modelo do PN para tratar alto nível de monóxido de carbono no ar (em BPMN 2.0).....	74
Figura 5.1 – Regra 1: “Limitação Visual” quando o nível é “Alta” ou “Média”	77
Figura 5.2 – Regra 2: “Limitação Auditiva” quando o nível é “Alta” ou “Média”.....	78
Figura 5.3 – Regra 3: “Limitação Auditiva” e “Limitação Visual” quando os níveis são “Alta” ou “Média”	79
Figura 5.4 – Regra 4: “Limitação Auditiva” e “Limitação Visual” quando os níveis são “Baixa” ou “Zero”	80
Figura 5.5 – Regra 5: Quando o cuidador não está na casa.....	81
Figura 5.6 – Regra 6: “Ação Ligar” para Luz do Teto.....	82
Figura 5.7 – Regra 7: “Ação Destrancar” para Grupo de Janela Inteligente.....	83
Figura 5.8 – Regra 8: “Ação Abrir” para Grupo de Janela Inteligente	83
Figura 5.9 – Regra 9: Interação com o Provedor de Saúde	84
Figura 5.10 – Tela de configuração inicial do <i>HouseSimulator</i>	86
Figura 5.11 – Tela de <i>feedback</i> do <i>HouseSimulator</i>	87
Figura 5.12 – Visão geral da interação entre o controlador e o <i>HouseSimulator</i>	88
Figura 5.13 – Implementação do framework proposto para instanciação de atividade com ação do tipo Atuação.....	89
Figura 5.14 – Áreas do controlador.....	90
Figura 5.15 – Configuração do simulador para a execução dos cenários.....	91
Figura 5.16 – Paciente John no estado agitado.....	92
Figura 5.17 – Controlador detecta estado de agitação no paciente	92
Figura 5.18 – Cenário 1, atividade (a) a ser executada.....	93
Figura 5.19 – Cenário 1, instâncias de “Informações” contidas na instância de “Ação Notificar Agitação”	93

Figura 5.20 – Cenário 1, instância informação do tipo “Texto” com conteúdo do tipo <i>string</i>	94
Figura 5.21 – Cenário 1, instância informação do tipo “Imagem” com conteúdo do tipo <i>base64Binary</i>	94
Figura 5.22 – Cenário 1, instância informação do tipo “Áudio” com conteúdo do tipo <i>base64Binary</i>	94
Figura 5.23 – Cenário 1, instância informação do tipo “Vídeo” com conteúdo do tipo <i>base64Binary</i>	94
Figura 5.24 – Cenário 1, instancia de atividade atualizada para atividade (a)	95
Figura 5.25 – Cenário 1, resultado das regras de inferência para atividade (g)	95
Figura 5.26 – Cenário 1, resultado das análises de similaridade sintática e semântica para “Ação Notificar Agitação”	96
Figura 5.27 – Cenário 1, atividade (a) executada pelo controlador	97
Figura 5.28 – Cenário 1, atividade (a) executada no simulador	97
Figura 5.29 – Cenário 1, atividade (b) executada	97
Figura 5.30 – Cenário 1, atividade (f) executada	98
Figura 5.31 – Cenário 1, atividade (g) a ser executada	98
Figura 5.32 – Cenário 1, instâncias de “Informações” contidas na “Ação Informar uso de drogas”	99
Figura 5.33 – Cenário 1, instância informação do tipo “URL do Vídeo” com conteúdo do tipo <i>string</i>	99
Figura 5.34 – Cenário 1, instância informação do tipo “URL da Imagem” com conteúdo do tipo <i>string</i>	99
Figura 5.35 – Cenário 1, instância informação do tipo “URL do Som” com conteúdo do tipo <i>string</i>	99
Figura 5.36 – Cenário 1, resultado das regras de inferência para atividade (g)	100
Figura 5.37 – Cenário 1, resultado das análises de similaridade sintática e semântica para “Ação Informar Uso de Drogas”	101
Figura 5.38 – Cenário 1, atividade (g) executada pelo controlador	102
Figura 5.39 – Cenário 1, atividade (g) executada no simulador	102
Figura 5.40 – Cenário 1, atividade (h) a ser executada	103
Figura 5.41 – Cenário 1, instâncias de “Informações” contidas na instância de “Ação Acalmar Agitação”	103
Figura 5.42 – Cenário 1, instância informação do tipo “Áudio” com conteúdo do tipo <i>base64Binary</i>	103
Figura 5.43 – Cenário 1, instância informação do tipo “Vídeo” com conteúdo do tipo <i>base64Binary</i>	104
Figura 5.44 – Cenário 1, resultado das regras de inferência para atividade (h)	104
Figura 5.45 – Cenário 1, resultado das análises de similaridade sintática e semântica para “Ação Acalmar Agitação”	105
Figura 5.46 – Cenário 1, atividade (h) executada pelo controlador	105
Figura 5.47 – Cenário 1, atividade (h) executada no simulador	106
Figura 5.48 – Cenário 1, atividade (i) a ser executada	106
Figura 5.49 – Cenário 1, instância de “Informação” contidas na instância de “Ação Requisita Assistência Agitação”	107
Figura 5.50 – Cenário 1, instância informação do tipo “Texto” com conteúdo do tipo <i>string</i>	107
Figura 5.51 – Cenário 1, resultado das regras de inferência para atividade (i)	107
Figura 5.52 – Cenário 1, resultado das análises de similaridade sintática e semântica para “Ação Requisita Assistência Agitação”	108
Figura 5.53 – Cenário 1, atividade (i) executada pelo controlador	108

Figura 5.54 – Cenário 1, atividade (i) executada no simulador.....	108
Figura 5.55 – Cenário 1, atividade (j) executada.....	109
Figura 5.56 – Cenário 1, fim do PN	109
Figura 5.57 – Fogão no estado ligado	110
Figura 5.58 – Cenário 2, controlador detecta estado ligado do fogão	110
Figura 5.59 – Cenário 2, atividade (a) executada	111
Figura 5.60 – Cenário 2, evento de tempo executado	111
Figura 5.61 – Cenário 2, atividade (c) executada	111
Figura 5.62 – Cenário 2, atividade (d) executada.....	111
Figura 5.63 – Cenário 2, atividade (e) executada.....	112
Figura 5.64 – Cenário 2, resultado das regras de inferência para atividade (e) foco no paciente	112
Figura 5.65 – Cenário 2, instâncias de “Informação” contidas na instância de “Ação Notificar Fogão Ligado”	113
Figura 5.66 – Cenário 2, instância de “Informação” do tipo “Texto” com conteúdo do tipo <i>string</i>	113
Figura 5.67 – Cenário 2, instância de “Informação” do tipo “Imagem” com conteúdo do tipo <i>base64Binary</i>	113
Figura 5.68 – Cenário 2, instância de “Informação” do tipo “Áudio” com conteúdo do tipo <i>base64Binary</i>	114
Figura 5.69 – Cenário 2, instância de “Informação” do tipo “Vídeo” com conteúdo do tipo <i>base64Binary</i>	114
Figura 5.70 – Cenário 2, atividade (e) com foco no paciente executada pelo controlador	115
Figura 5.71 – Cenário 2, atividade (e) com foco no paciente executada no simulador.....	116
Figura 5.72 – Cenário 2, resultado das regras de inferência para atividade (e) com foco na cuidadora	117
Figura 5.73 – Cenário 2, atividade (e) com foco na cuidadora executada pelo controlador ..	118
Figura 5.74 – Cenário 2, atividade (e) com foco na cuidadora executada no simulador	119
Figura 5.75 – Cenário 2, fim do PN	119
Figura 5.76 – Janela inteligente no estado fechado e a luz do teto no estado desligado	120
Figura 5.77 – Cenário 3, detectado quando o paciente entra no cômodo.....	120
Figura 5.78 – Cenário 3, atividade (a) executada	121
Figura 5.79 Cenário 3, evento de tempo executado	121
Figura 5.80 – Cenário 3, atividade (b) executada.....	121
Figura 5.81 – Cenário 3, atividade (c) a ser executada.....	122
Figura 5.82 – Cenário 3, resultado das regras de inferência para atividade (c).....	122
Figura 5.83 – Cenário 3, atividade (c) executada pelo controlador.....	122
Figura 5.84 – Cenário 3, atividade (c) executada no simulador	123
Figura 5.85 – Cenário 3, fim do PN	123
Figura 5.86 – Cenário 4, detectado alto índice de monóxido de carbono no ar	124
Figura 5.87 – Cenário 4, atividade (a) a ser executada.....	124
Figura 5.88 – Cenário 4, resultado das regras de inferência para atividade (a).....	125
Figura 5.89 – Cenário 4, atividade (a) executada pelo controlador.....	125
Figura 5.90 – Cenário 4, atividade (a) executada no simulador	126
Figura 5.91 – Cenário 4, atividade (b) a ser executada	126
Figura 5.92 – Cenário 4, resultado das regras de inferência para atividade (b)	127
Figura 5.93 – Cenário 4, atividade (b) executada pelo controlador	127
Figura 5.94 – Cenário 4, atividade (b) executada no simulador.....	128
Figura 5.95 – Cenário 4, atividade (c) a ser executada.....	128

Figura 5.96 – Cenário 4, resultado das regras de inferência para atividade (c) com foco no paciente.....	129
Figura 5.97 – Cenário 4, instâncias de “Informação” contidas na instância de “Ação Notificar Alto Nível de Monóxido de Carbono”	129
Figura 5.98 – Cenário 4, instância informação do tipo “Texto” com conteúdo do tipo <i>string</i>	129
Figura 5.99 – Cenário 4, instância de “Informação” do tipo “Imagem” com conteúdo do tipo <i>base64Binary</i>	130
Figura 5.100 – Cenário 4, instância de “Informação” do tipo “Áudio” com conteúdo do tipo <i>string</i>	130
Figura 5.101 – Cenário 4, instância de “Informação” do tipo “Vídeo” com conteúdo do tipo <i>string</i>	130
Figura 5.102 – Cenário 4, resultado das análises de similaridade sintática e semântica para “Ação Notificar Alto Nível de Monóxido de Carbono” com foco no paciente	131
Figura 5.103 – Cenário 4, atividade (c) com foco no paciente executada pelo controlador ..	132
Figura 5.104 – Cenário 4, atividade (c) com foco no paciente executada no simulador.....	132
Figura 5.105 – Cenário 4, resultado das regras de inferência para atividade (c) com foco na cuidadora	133
Figura 5.106 – Cenário 4, atividade (c) com foco na cuidadora executada pelo controlador	134
Figura 5.107 – Cenário 4, atividade (c) com foco na cuidadora executada no simulador	134
Figura 5.108 – Cenário 4, fim do PN	134

LISTA DE TABELAS

Tabela 2.1 – Valores de correlações de medidas.....	29
Tabela 5.1 – Cenário 1, ordenação dos pares para a atividade (a) através da similaridade semântica	96
Tabela 5.2 – FuncDip localizados no mesmo cômodo do paciente e que possuem funcionalidades do tipo “Interação com Pessoa”	100
Tabela 5.3 – Cenário 1, ordenação dos pares para a atividade (g) através da similaridade semântica	102
Tabela 5.4 – Cenário 1, ordenação dos pares para a atividade (h) através da similaridade semântica	105
Tabela 5.5 – Cenário 2, ordenação dos pares para a atividade (e) com foco no paciente através da similaridade semântica.....	114
Tabela 5.6 – Cenário 2, ordenação dos pares para a atividade (e) com foco na cuidadora através da similaridade semântica	117
Tabela 5.7 – Cenário 4, ordenação dos pares para a atividade (c) com foco no paciente através da similaridade semântica.....	131
Tabela 5.8 – Cenário 4, ordenação dos pares para a atividade (c) com foco na cuidadora através da similaridade semântica	133

LISTA DE QUADROS

Quadro 2.1 – Abordagens de matching semântico de SW e os algoritmos utilizados para o cálculo de similaridade entre os conceitos	35
Quadro 2.2 – Comparativo dos trabalhos relacionados de matching de SW baseado em pragmático	39
Quadro 2.3 – Comparativo dos trabalhos relacionados de coordenação de serviços em <i>Smart Home</i> e <i>Home Care</i>	45
Quadro 4.1 – Tipos de triplas e cenários para sua exemplificação.....	69
Quadro 5.1 – Comparativo entre a abordagem proposta e os trabalhos relacionados de matching de SW baseado em pragmático.....	135
Quadro 5.2 – Comparativo entre a abordagem proposta e os trabalhos relacionados de coordenação de serviços em <i>Smart Home</i> e <i>Home Care</i>	136

LISTA DE ABREVIATURAS E SIGLAS

AmI	<i>Ambient Intelligence</i>
BPMN	<i>Business Process Model Notation</i>
CIM	<i>Computation Independent Model</i>
HCS	<i>Home Care Systems</i>
OWL	<i>Web Ontology Language</i>
OWL-S	<i>OWL Web Ontology Language for Services</i>
PIM	<i>Platform Independent Model</i>
PN	Processo de Negócio
PSM	<i>Platform Specific Model</i>
SOAP	<i>Simple Object Access Protocol</i>
SW	Serviço Web
SWRL	<i>Semantic Web Rule Language</i>
UDDI	<i>Universal Description, Discovery, and Integration</i>
WSDL	<i>Web Services Description Language</i>

SUMÁRIO

AGRADECIMENTOS	5
RESUMO	7
ABSTRACT	9
LISTA DE FIGURAS	11
LISTA DE TABELAS	15
LISTA DE QUADROS	17
LISTA DE ABREVIATURAS E SIGLAS	19
SUMÁRIO	21
1 INTRODUÇÃO	23
1.1 Objetivo Geral	25
1.2 Objetivos Específicos	25
2 FUNDAMENTAÇÃO TEÓRICA	27
2.1 Ontologia com foco em <i>Matching</i> de Ontologia	27
2.2 Serviços Web com foco em <i>Matching</i> de Serviços Web	31
2.2.1 Baseado em Palavras	34
2.2.2 Baseado em Sintaxe.....	34
2.2.3 Baseado em Semântica	34
2.2.4 Baseado em Pragmática.....	36
2.3 Processo de Negócio	40
2.4 <i>Smart Home</i> e <i>Home Care</i> com foco em Coordenação de Serviços	41
2.4.1 Trabalhos Relacionados.....	42
2.5 Resumo do Capítulo 2	45
3 CENÁRIO MOTIVACIONAL E ABORDAGEM PROPOSTA	47
3.1 Cenário Motivacional	47
3.2 Abordagem Proposta	48
3.2.1 Modelo para aplicações <i>Home Care</i>	48
3.2.2 Rótulo de atividade de modelo de PN	59
3.2.3 Instanciação das atividades de modelo de PN	60
3.3 Resumo do Capítulo 3	67
4 CENÁRIOS DE APLICAÇÃO	69
4.1 Cenário 1: Tratar estado de agitação	70
4.2 Cenário 2: Notificação do fogão ligado	71
4.3 Cenário 3: Gestão/control de iluminação	72
4.4 Cenário 4: Tratar alto nível de monóxido de carbono no ar	73
4.5 Resumo do Capítulo 4	74
5 AVALIAÇÃO BASEADA EM ESTUDOS DE CASO	75
5.1 Etapa 1: Criação e Configuração do Ambiente;	75
5.1.1 Implementação do modelo para aplicações <i>Home Care</i>	76
5.1.2 Especificação de um Simulador de Casa	84
5.1.3 Especificação do Conjunto de Descrições Semânticas para as Funcionalidades dos Dispositivos	87
5.1.4 Construção de um Controlador.....	88
5.2 Etapa 2: Execução dos Processos de Negócios	91
5.2.1 Execução do Cenário 1	92
5.2.2 Execução do Cenário 2	109
5.2.3 Execução do Cenário 3	120

5.2.4	Execução do Cenário 4.....	124
5.3	Análise da Abordagem.....	135
5.4	Resumo do Capítulo 5.....	136
6	CONCLUSÕES.....	139
	REFERÊNCIAS	141
	APÊNDICE A – DISPOSITIVOS E SUAS FUNCIONALIDADES.....	149
	APÊNDICE B – EXEMPLO DE DESCRIÇÃO SEMÂNTICA	153
	ANEXO A – HIERARQUIA DOS DATATYPES DO XML.....	161

1 INTRODUÇÃO

A população mundial é atualmente maior que 7 bilhões de habitantes, onde 8,2% dessas pessoas tem mais de 65 anos (WORDOMETERS, 2014). Ainda, de acordo com o *Department of Economic and Social Affairs of the United Nations* (2013), a expectativa de vida está aumentando. Enquanto não há alternativas para hospitais no tratamento de pacientes com doença terminal, idosos que são saudáveis ainda precisam de assistência para viver com independência (PUNG et al., 2009). Com essa perspectiva, tecnologias de inovação são necessárias para que idosos possam permanecer em suas próprias casas (BASTIDE; ZEFOUNI; LAMINE, 2010).

Nesse contexto, *Home Care Systems* (HCS) emergiram. De acordo com McGee-Lennon (2008), HCS podem ser definidos como uma tecnologia usada para dar suporte na realização de tarefas de rede, fornecendo os meios para coletar, distribuir, analisar e gerenciar informações relacionadas aos cuidados. Tal tecnologia tipicamente inclui sensores, dispositivos, telas, dados, redes e infraestrutura computacional. Desse modo, HCS visam capacitar pessoas que necessitam de assistência para que continuem a viver em suas próprias casas, enquanto que sua saúde e autonomia se deterioram (AUVINEN et al., 2011).

A fim de gerenciar eficientemente a assistência domiciliar de saúde, uma conexão virtual entre o paciente e o hospital tem que ser estabelecida, suportando atividades de monitoramento, requisição de serviços e gerenciamento de protocolos médicos ao qual o paciente está sujeito (ARDISSONO et al., 2006).

HCS pode ser visto como uma área dentro de Inteligência do Ambiente, em inglês *Ambient Intelligence* (AmI), que, segundo Cook et al. (2009), possui as funções básicas: coletar as informações do ambiente (sensoriamento), raciocinar sobre essas informações, e atuar conforme esse raciocínio.

Alguns trabalhos, tais como Ardissono et al. (2006), Cho et al. (2007), Lamine et al. (2010) e Kao e Yuan (2012), propõem a utilização de modelos de processos de negócios para descrever as atividades (de sensoriamento e de atuação) que devem ser realizadas em um HCS. Tal modelo de processo de negócio é acionado quando uma situação de interesse é detectada. Neste trabalho, situação de interesse refere-se a situações indesejadas que podem acontecer a um idoso em um contexto *Home Care* e são relevantes para serem monitoradas pelo sistema. Ainda, quando uma situação de interesse ocorre, a aplicação deveria reagir de forma a retornar a situação ao normal.

Uma atividade em processo de negócio, frequentemente é executada através de um serviço Web. Em HCS, dispositivos domésticos e suas funcionalidades podem ser representados como serviço Web. Além disso, a execução da atividade em HCS deveria ser sensível ao contexto, de forma que o sistema capture informações contextuais, raciocine sobre elas, e se adapte.

Nesse sentido, as atividades de atuação (ver Seção 3.2.1.5) especificadas por um modelo de processo de negócio poderiam ser executadas por uma funcionalidade de um dispositivo doméstico. Contudo, em um ambiente onde dispositivos possuem diferentes características e estão dispersos pela casa, o desafio é identificar o dispositivo doméstico que pode executar a determinada atividade, levando em consideração as características do paciente e do dispositivo (informações contextuais).

Com isso, a questão de pesquisa que este trabalho visa solucionar é: “Dada uma atividade de atuação de um modelo de processo de negócio em ambiente de *Home Care*, como podemos instanciar essa atividade através de funcionalidade de dispositivos domésticos, representados por serviços Web, de maneira sensível ao contexto?”.

Para responder essa questão, é proposto uma abordagem, onde as orientações para tratar cada situação de interesse são descritas por um modelo de processo de negócio. Para cada atividade de um modelo de processo de negócio é atualizado o modelo ontológico ao qual é descrito o contexto, e para as atividades de atuação é raciocinado quais possíveis funcionalidades de dispositivos domésticos poderiam executar a atividade (pré-seleção sensível ao contexto). Após, com auxílio de um modelo ontológico, é criado um conjunto de serviços Web abstratos, e com esses serviços é feita a filtragem das funcionalidades de dispositivos (representadas por serviços Web) previamente selecionadas, utilizando técnicas de *matching* de serviços Web (sintático e semântico), para garantir que o controlador consiga acionar o serviço Web. Por fim, a partir do conjunto de serviços Web selecionados é executada a atividade de atuação do processo de negócio.

Para realizar a seleção de serviços Web, levou-se em consideração as descrições semânticas e os tipos de dados do serviço Web. Dessa forma, tentou-se selecionar o serviço Web mais próximo do serviço Web abstrato de maneira semântica e sintática. Ainda, considerou-se aspectos contextuais, onde, de acordo Rong e Liu (2010), contexto é qualquer informação que afeta explicitamente ou implicitamente a requisição ao serviço Web.

A avaliação da abordagem foi feita através da simulação de 4 cenários de aplicação distintos retirados da literatura onde se exemplificou o comportamento da abordagem. A

avaliação utilizou o simulador *HouseSimulator*, desenvolvido por Pötter (2013), ao qual simula uma casa com dispositivos que recebem e enviam dados através de serviços Web

Por fim, as principais contribuições deste trabalho são a especificação de um modelo ontológico e de rótulos para atividades de modelo de processo de negócio para dar suporte ao framework que visa instanciar atividades de atuação de um processo de negócio através de funcionalidades de dispositivos, representadas por serviço Web, em um ambiente *Home Care* sensível ao contexto. A seleção das funcionalidades de dispositivos é baseada no contexto do usuário e dos dispositivos, e a filtragem dos serviços Web (que representam as funcionalidades de um dispositivo) considera os aspectos semânticos e sintáticos, permitindo que o sistema se adapte ao contexto do usuário e dos dispositivos. Assim, casas podem incorporar novos dispositivos e ainda gerenciá-los adequadamente.

1.1 Objetivo Geral

O objetivo geral é a criação de uma abordagem que permita a instanciação de atividades de atuação de um modelo de processo de negócio, em ambiente de *Home Care*, através de funcionalidade de dispositivos domésticos, representados por serviços Web, de maneira sensível ao contexto.

1.2 Objetivos Específicos

Para alcançar o objetivo geral, os seguintes objetivos específicos foram visados:

- Definir uma normalização para os rótulos de modelo de processo de negócio que descrevem as orientações das situações de interesse;
- Definir semânticas diferentes para cada tipo de rótulos;
- Definir um modelo ontológico para dar suporte à criação do serviço Web abstrato referente a ação indicada na atividade de um modelo de processo de negócio e à seleção do serviço Web que descreve a funcionalidade do dispositivo;
- Definir um algoritmo de matching semântica de serviço Web;
- Definir um algoritmo de matching sintática de serviço Web;
- Definir uma abordagem para seleção de serviço Web que leve em consideração o contexto;

1.3 Estrutura do Texto

Este trabalho está organizado da seguinte maneira: Seção 2 traz a fundamentação teórica e trabalhos relacionados; Seção 3 descreve o cenário motivacional desse trabalho e explica a abordagem proposta; Seção 4 descreve quatro cenários de aplicação da abordagem; Seção 5 mostra a avaliação da abordagem baseada em estudo de caso; Seção 6 conclui o trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

A seguir será apresentada a fundamentação teórica ao qual este trabalho é baseado, sendo que os seguintes tópicos são abordados:

- Ontologia com foco em *matching* de ontologia;
- Serviços Web com foco em *matching* de serviços Web;
- Processo de negócio;
- *Smart Home* e *Home Care* com foco em coordenação de serviços.

A importância desses tópicos neste trabalho se deve ao fato de que eles são utilizados em diversas situações importantes, a saber: utiliza-se ontologia para modelar pessoa, organização, controlador, dispositivo e localização, que são conceitos de domínio fundamentais nesta abordagem; aplica-se o *matching* de ontologias, por ser uma técnica que pode ser empregada no *matching* de serviços Web; utiliza-se serviços Web por serem aplicados para representar as funcionalidades dos dispositivos domésticos e, dessa forma, possibilitar a interação com esses dispositivos; emprega-se o *matching* de serviços Web, pois ele é necessário para invocar o serviço disponibilizado pelo dispositivo em um ambiente onde não se sabe de forma automática quais argumentos (em termos de o que eles representam) devem ser passados para invocar o serviço (dispositivos domésticos podem ser construídos por diferentes fabricantes sem padronização); aproveita-se de processos de negócio por serem utilizados para modelar as orientações de gerenciamento de situação de interesse; e, por fim, contextualiza-se as áreas *Smart Home* e *Home Care* (que é o domínio deste trabalho), focando na coordenação de serviço, pois é a problemática deste trabalho.

2.1 Ontologia com foco em *Matching* de Ontologia

De acordo com Gruber (1993), ontologias são uma “especificação formal e explícita de uma conceituação compartilhada”, sendo que é uma conceituação porque refere-se a um modelo abstrato do mundo real, é explícita porque os tipos de conceitos utilizados e suas restrições de uso são explicitamente definidos, é formal porque pode ser entendida por máquina e é compartilhada porque captura um conhecimento consensual, aceito por um grupo.

Ainda, segundo os trabalhos de Vasilecas et al. (2009) e Kalibatiene e Vasilecas (2010), uma ontologia define conceitos básicos, suas definições e seus relacionamentos,

compreendendo o vocabulários de um domínio de negócio, e seus axiomas para restringir os relacionamentos e interpretações dos conceitos.

Conforme o trabalho de Kalibatiene e Vasilecas (2011), ontologias podem ser descritas em várias linguagens ontológicas, tais como: *Knowledge Interchange Format* (KIF), *Frame Logic* (F-logic), *Resource Description Framework* (RDF), DAML+OIL (que combina características da *DARPA Agent Markup Language* e da *Ontology Interchange Language*), *Web Ontology Language* (OWL). Entretanto, ainda de acordo com esse trabalho, a OWL é a linguagem ontológica mais popular.

A OWL fornece, ainda, três sublinguagens com aumento do grau de expressividade do menor para o maior (MCGUINNESS; HARMELEN, 2004):

- OWL-Lite: que fornece uma classificação hierárquica e restrições simples;
- OWL-DL: que fornece a máxima expressividade, mantendo a completude computacional (todas as conclusões são garantidas para serem computáveis) e decidibilidade (todas os cálculos terminam em um tempo finito);
- OWL-Full: que fornece a máxima expressividade e a liberdade sintática do RDF, mas sem a garantias computacionais.

Ainda, regras podem ser utilizadas em conjunto com ontologias, tais como: *Rule Markup Language* (RuleML), *Planning Domain Definition Language* (PDDL) e *Semantic Web Rule Language* (SWRL).

Desta forma, motores de inferência conseguem raciocinar baseando-se em regras e ontologias. Motores de inferência são programas que inferem consequências lógicas a partir de um conjunto de fatos afirmados explicitamente ou axiomas e fornecem suporte automático para tarefas de inferências tais como classificação, *debugging* e consultas. Exemplos de motores de inferência: FaCT++, RacerPro, Pellet, KAON2 e Hoolet, HermeT.

Com isso, ontologias são utilizadas para modelar vários domínios. Entretanto, um mesmo domínio pode ser modelado por ontologias distintas, havendo sobreposição de conceitos. Como ocorre nos trabalhos de Pernas et al. (2012) e Machado e Oliveira (2014), onde ambos tem uma ontologia distinta do domínio de educação.

Surgiu então, a necessidade de *matching* de ontologias, que é, conforme Otero-Cerdeira et al. (2015), um processo complexo que ajuda em reduzir a lacuna semântica entre diferentes representações sobrepostas de um mesmo domínio.

As abordagens de *matching* de ontologias podem ser classificadas do seguinte modo (SÁNCHEZ et al., 2012):

- Medida baseada em contagem de arestas: medem a similaridade entre dois conceitos considerando o comprimento do caminho conectado formado pelos conceitos e/ou a posição dos conceitos na taxonomia;
- Medida baseada em características: a similaridade entre dois conceitos é medida como uma função de suas propriedades ou baseadas nos seus relacionamentos para com outros conceitos similares;
- Medida baseada em conteúdo de informação: exploram a noção de conteúdo de informação (CI), pela associação da probabilidade de aparecer para cada conceito de uma taxonomia, calculada a partir da ocorrência de um dado *corpus*.

A Tabela 2.1 informa o trabalho por quem a abordagem foi proposta, em que a abordagem é baseada, o valor de correlação no *benchmark* de Miller e Charles (M&C), o valor de correlação pelo *benchmark* de Rubenstein e Goodenough (R&G), e onde a abordagem foi avaliada. Os *benchmarks* de Rubenstein e Goodenough e o de Miller e Charles são considerados os testes padrões de fato para avaliar e comparar a acurácia de medidas de similaridade entre palavras. Tal acurácia foi obtida calculando-se a correlação entre as avaliações de similaridade reportadas nestes benchmarks contra aquelas obtidas por uma avaliação computadorizada. Com isso, se duas avaliações são exatamente a mesma, o que indica que a função de similaridade imita o julgamento humano, então a correlação é 1. Caso a correlação seja igual a 0, isso indica que a avaliação automática não está relacionada com a opinião humana

Tabela 2.1 – Valores de correlações de medidas

<i>Trabalho</i>	<i>Medida baseada em</i>	<i>M&C</i>	<i>R&G</i>	<i>Avaliado em</i>
Rada et al. (1989)	cont. de arestas	0.59	N/A	Petrakis et al. (2006)
Wu e Palmer (1994)	cont. de arestas	0.74	N/A	Petrakis et al. (2006)
Leacock and Chodorow (1998)	cont. de arestas	0.74	0.77	Patwardhan e Pedersen (2006)
Li et al. (2003)	cont. de arestas	0.82	N/A	Petrakis et al. (2006)
Tversky (1977)	característica	0.73	N/A	Petrakis et al. (2006)
Rodriguez e Egenhofer (2003)	característica	0.71	N/A	Petrakis et al. (2006)
Petrakis et al. (2006)	característica	0.74	N/A	Petrakis et al. (2006)
Sánchez et al. (2012)	característica	0.83	0.857	Sánchez et al. (2012)
Resnik (1995)	CI	0.72	0.72	Patwardhan e Pedersen (2006)
Resnik (1995) (CI computado como Seco et al. (2004))	CI	N/A	0.829	Zhou et al. (2008)
Resnik (1995) (CI computado como Zhou et al. (2008))	CI	N/A	0.842	Zhou et al. (2008)
Jiang e Conrath (1997)	CI	0.73	0.75	Patwardhan e Pedersen (2006)

<i>Trabalho</i>	<i>Medida baseada em</i>	<i>M&C</i>	<i>R&G</i>	<i>Avaliado em</i>
Jiang e Conrath (1997) (CI computado como Seco et al. (2004))	CI	N/A	0.823	Zhou et al. (2008)
Jiang and Conrath (1997) (CI computado como Zhou et al. (2008))	CI	N/A	0.858	Zhou et al. (2008)
Lin (1998)	CI	0.7	0.72	Patwardhan e Pedersen (2006)
Lin (1998) (CI computado como Seco et al. (2004))	CI	N/A	0.845	Zhou et al. (2008)
Lin (1998) (CI computado como em Zhou et al. (2008))	CI	N/A	0.866	Zhou et al. (2008)
Seco et.al. (2004)	CI	-	-	-
Zhou et. al. (2008)	CI	-	-	-
Seddqui e Aono (2010)	CI	-	-	-

Fonte: tabela modificada de Sánchez et al. (2012) e Fernandez et al. (2007)

Como demonstrado na Tabela 2.1, a abordagem proposta por Lin (1998) com IC computado como em Zhou et al. (2008) é a abordagem que obteve o melhor valor de correlação. Por isso, esta abordagem foi escolhida para implementação deste trabalho (ver Seção 5.1.4) e será melhor detalhada a seguir.

Para Lin (1998) a similaridade entre dois conceitos a e b é medida pela razão da quantidade de informação necessário para relatar semelhanças de a e b e a informação necessária para descrever totalmente o que a e b são:

$$sim(a, b) = \frac{2 * sim_{resnik}(a, b)}{IC(a) + IC(b)}$$

onde:

- $sim_{resnik}(a, b)$: é a função de similaridade proposta por Resnik (1995);
- $IC(a)$ é função que calcula o conteúdo de informação do conceito a .

Entretanto, para Resnik (1995) a similaridade entre dois conceitos a e b é baseada na noção de que quanto mais informação dois conceitos compartilham em comum, mais similar eles são, e a informação compartilhada por dois conceitos é indicada pelo IC dos conceitos que generalizam eles na taxonomia:

$$sim_{resnik}(a, b) = \max_{c \in S(a,b)} [IC(c)]$$

onde:

- $S(a,b)$ é o conjunto de conceitos que são generalizações (conceitos pais) de ambos os conceitos a e b .

O cálculo do CI proposto por Zhou et al. (2008) baseia-se nos *hyponyms* (conceitos filhos) e com a profundidade do conceito na taxonomia:

$$IC_{zhou}(a) = k * \left(1 - \frac{\log(hypo(a) + 1)}{\log(maxNodes)} \right) + (1 - k) * \left(\frac{\log(depth(a))}{\log(maxDepth)} \right)$$

onde:

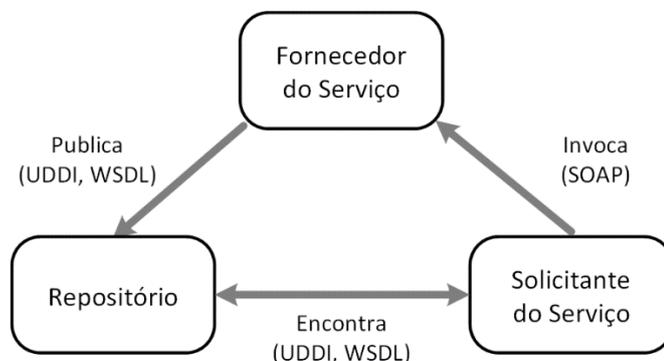
- *hypo(a)*: é uma função que retorna o número de *hyponyms* do conceito *a*;
- *depth(a)*: é a profundidade do conceito *a* na estrutura da ontologia (taxonomia);
- *maxNodes*: é o número máxima de conceitos existentes na ontologia;
- *maxDepth*: é a profundidade máxima da estrutura da ontologia (taxonomia).

2.2 Serviços Web com foco em *Matching* de Serviços Web

Su e Wang (2010) definem serviço Web como uma unidade interoperável de lógica de aplicação que transcende linguagens de programação, sistemas operacionais, protocolos de comunicação, rede, e dependências de representação dados e questões.

Serviços Web (SW) são formados por um conjunto de padrões abertos que definem como os componentes reusáveis devem ser especificados (através de *Web Service Description Language – WSDL*), como podem ser anunciados para a descoberta e reuso (via *Universal Description, Discovery, and Integration API – UDDI*), e como eles podem ser invocados em tempo de execução (via *Simple Object Access Protocol API – SOAP*) (STROULIA, 2003).

Figura 1 – Arquitetura padrão do serviço Web



Fonte: Imagem traduzida de Atkinson et al. (2007, p.1)

A Figura 1 mostra a arquitetura padrão de um SW, onde (ATKINSON et al., 2007; FENSEL et al., 2007):

- o fornecedor do SW, depois de descrever a interface de seu serviço, tipicamente feita em WSDL, publica-o em um repositório central para que este possa ser descoberto;
- o solicitante do SW tem então a capacidade de executar consultas para obter uma referência para o serviço designado, essa interação é feita tipicamente pelo padrão UDDI;
- para invocar um SW, o solicitante necessita saber como fazê-lo (ou seja, precisa da definição da interface), essa informação é fornecida pelo WSDL, e a transmissão de dados é feita através do padrão SOAP.

Os padrões de SW descritos anteriormente resolvem muitos problemas no nível técnico (aspectos sintáticos da interação do serviço), descrevendo *como* o serviço pode ser acessado, mas a semântica do SW, a descrição de *o que* o serviço faz e em que ordem suas operações têm que ser chamadas é somente, se houver, descrita em linguagem natural em entradas ou comentários de uma descrição WSDL ou entrada UDDI (WANG et al., 2004; FENSEL et al., 2007).

Ainda, muitos trabalhos argumentam que o método de descoberta de SW no UDDI não tem poder suficiente para descoberta automatizada pelos seguintes fatores:

- A descoberta de serviços potencialmente relevantes no catalogo UDDI por categoria confia no entendimento do senso comum compartilhado do domínio da aplicação pelo desenvolvedor que publica e o que consume o serviço especificado. Dessa forma, é de responsabilidade do fornecedor publicar os serviços na categoria UDDI apropriada, e do consumidor procurá-los corretamente (STROULIA; WANG, 2005; LIU et al., 2009);
- A descoberta de serviços por pesquisa de palavras-chave não permite nenhuma forma de inferência ou casamento (*matching*) flexível de palavras-chave, e ainda requer algum conhecimento prévio sobre o registro, como número chave da entidade de negócio ou nome (PAOLUCCI et al., 2002; HAO; ZHANG, 2004; SANTOS; WIVES; DE OLIVEIRA, 2009);
- O UDDI não fornece um ranking dos serviços retornados, o que pode ser um problema quando se tem um grande número de resultados retornados (SANTOS; WIVES; DE OLIVEIRA, 2012).

Como afirmam Sivashanmugam et al. (2003), a chave para a descoberta de SW é agregar *semântica* na descrição dos serviços, e então usar algoritmos de *matching* semânticos para encontrar o serviço requerido.

Tipicamente, algoritmos de descoberta e *matching* de SW assumem que, dado um serviço solicitante *R*, deseja-se encontrar, em um conjunto de serviços anunciantes, o serviço *S* tal que este seja o mais similar ao serviço *R* (GANJISAFFAR et al., 2007).

Para resolver esse problema de descoberta surgiu o *Serviço Web Semântico* (SWS), cuja semântica pode ser descrita em termos de Entradas, Saídas, Pré-condições e Efeitos, através de ontologias.

Wang et al. (2004) definem SWS como a descrição das capacidades e do conteúdo de um SW em uma linguagem interpretável por computador e melhorar a qualidade das tarefas existentes, incluindo descoberta, invocação, composição, monitoramento e recuperação de SW. Com isso, os trabalhos nesta área esperam minimizar a intervenção do usuário (FENSEL et al., 2007).

Para habilitar um SW com semântica, muitas abordagens de descrição foram propostas, tais como: *DARPA Agent Markup Language for Services* (DAML-S), *OWL Web Ontology Language for Services* (OWL-S), *Semantic Annotations for WSDL and XML Schema* (SAWDL), *YASA for WSDL* (YASA4WSDL), *Web Service Modeling Ontology* (WSMO) e *Web Service Semantics* (WSDL-S).

Não existe na literatura uma definição única dos tipos de *matching* envolvidos na descoberta de SW. Por isso, a fim de dar uma visão geral, será adotada a visão de Rong e Liu (2010) juntamente com a visão dos trabalhos de Stroulia e Wang (2005) (STROULIA; WANG, 2005), Masuch et al. (2012) (MASUCH et al., 2012) e Klusch e Kapahnke (2012b). Desta forma, a descoberta de SW pode incluir diferentes tipos de *matching*, tais como:

- Baseado em Palavras: emprega mecanismo de similaridade baseado em palavras-chaves para encontrar o SW adequado.
- Baseado em Sintaxe: se importa com a função, a interface, a operação, a estrutura do SW ou qualquer combinação destes.
- Baseado em Semântica: compara os serviços de acordo com sua descrição formal.
- Baseado em Pragmático: é um modelo de *matching* que engloba os anteriores com a habilidade de levar em consideração o contexto para a descoberta de SW.

Entretanto, os trabalhos de descoberta de SW não só se baseiam em um tipo de *matching*, eles utilizam as abordagens combinadas para obter um melhor resultado.

2.2.1 Baseado em Palavras

A intuição básica nesse tipo de *matching* é que quanto mais palavras dois documentos compartilham em comum, mais similares eles são (STROULIA; WANG, 2005). Por isso, a maioria dos trabalhos desse tipo de *matching* são baseados em análise de frequência de termos.

Contudo, conforme Dong et al. (2004), esse tipo de abordagem é insuficiente no contexto de SW porque a documentação textual para operações SW são altamente compactas. Desta forma, trabalhos como Dong et al. (2004), Liu et al. (2010), Xu et al. (2012), Plebanie Pernici (2012), utilizam esse tipo de *matching* em conjunto com outras abordagens.

2.2.2 Baseado em Sintaxe

Tipicamente, *matching* de SW baseado em sintaxe se importa com o tipo de dado dos parâmetros de entrada e saída.

Um abordagem de *matching* de tipo de dado é apresentada em Gao et al. (2002), onde dado dois tipos, t_1 e t_2 , t_1 é similar a t_2 se t_1 e t_2 representam o mesmo tipo de dado; ou se t_1 é um subtipo de t_3 .

Outra abordagem foi proposta por Plebani e Pernici (2009), ao qual o cálculo do valor de similaridade entre tipos de dados retorna um valor entre 0 e 1. Esta abordagem é específica para tipo de dados expressados em XML Schema Definition (ver Anexo A), onde um tipo de dado pode ser *built-in* ou *complex*. Tipos de dados *built-in* são organizados em cinco grupos (*integer*, *real*, *string*, *date*, *boolean*). A função difere quando a combinação de tipo *built-in* e *complex*. Entretanto, como afirma Broens et al. (2004), este tipo de *matching* conduz a resultados de descoberta pobres, pois SW similares sintaticamente podem ser diferentes semanticamente em termos de descrição de serviço.

2.2.3 Baseado em Semântica

Matching semânticos focam principalmente no *matching* de conceitos relacionados aos parâmetros passados como entrada e saída de um SW, sendo esses definidos por conceitos de uma ontologia. Tais conceitos são estruturados hierarquicamente de forma que os algoritmos de *matching* podem verificar a igualdade e as dependências taxonômicas entre os conceitos do SW solicitante e do anunciante.

Tipicamente, o *matching* semântico verifica, para cada conceito presente no parâmetro de entrada/saída de um serviço R, o conceito mais similar a esse contido no conceito do parâmetro de entrada/saída de um serviço S.

O *matching* de conceitos na ontologia pode ser dividido em:

- Baseado em lógica: que retorna a o grau de *matching* entre dois conceitos com base na taxonomia da ontologia;
- Não baseado em lógica: que retorna um valor numérico que informa a similaridade ou a distância entre dois conceitos na ontologia.

Contudo, Lampe e Schulte (2012) alertam para as desvantagens de se utilizar *matching* semântico baseadas em lógica, tais como: o *ranking* dos serviços retornados nessa abordagem possui um granularidade grossa, conseqüentemente, para se ter uma classificação mais refinada, necessita-se o uso de análise de similaridade adicional (geralmente não baseada em lógica); o uso desse tipo de *matching* semântico dificulta ou inibe a combinação com medidas de similaridade adicionais (por exemplo, medida de similaridade com uso de palavras); esse tipo de abordagem faz suposições básicas quanto a generalização e especialização de conceitos semânticos em ontologias que não são necessariamente cumpridas.

Abordagens de *matching* semântico não baseadas em lógica geralmente utilizam algoritmos de *matching* de ontologia (Seção 2.1) para calcular a similaridade entre os conceitos de entrada/saída. O Quadro 2.1 mostra alguns exemplos de trabalhos com *matching* semântico com valor numérico e onde o cálculo de *matching* de conceito foi baseado.

Quadro 2.1 – Abordagens de *matching* semântico de SW não baseados em lógica e os algoritmos utilizados para o cálculo de similaridade entre os conceitos

<i>Trabalho</i>	<i>Cálculo de similaridade entre conceitos baseado em</i>
Guo et al. (2005)	Rodríguez e Egenhofer (2003)
Ganjisaffar et al. (2007)	Zhou et al. (2008)
Xia e Yoshida (2008)	Rada et al. (1989)
Liu et al. (LIU et al., 2009)	Li et al. (2003)
Fethallah et al. (2010)	Wu e Palmer (1994)
Zhang et al. (2011)	Tversky (1977)
Klusck e Kapahnke (2012a)	Li et al. (2003)
Santos et al. (2012)	Li et al. (2003)
Klusck e Kapahnke (2012b)	Li et al. (2003)

Schulte et al. (2012)	Resnik (1995) e Lin (1998)
Mohebbi et al. (2014)	Jiang e Conrath (1997) e Lin (1998)

Fonte: quadro criado pela autora

2.2.4 Baseado em Pragmática

O contexto é utilizado principalmente para selecionar o serviço mais adequado para executar uma ação dado uma situação (TRUONG; DUSTDAR, 2009).

Segundo Dey (2001), contexto é qualquer informação que pode ser usada para caracterizar a situação de uma entidade, sendo que entidade pode ser uma pessoa, um lugar, ou um objeto, e situação é a descrição dos estados das entidades relevantes.

Entretanto, na perspectiva de descoberta de SW, Rong e Liu (2010) definem contexto como qualquer informação que afeta explicitamente ou implicitamente a formação da requisição do SW.

Suraci et al. (2007) cita as duas maiores razões para a utilização de contexto em *matching* de SW:

- Para completar a solicitação do SW: É realizada quando o solicitante do serviço quer encontrar serviços com base em critérios contextuais, combinando o pedido com o contexto do solicitante, do serviço e do ambiente.
- Para completar a passagem de parâmetro de entrada: É realizada quando o serviço requer entradas que o solicitante não fornece, e essas entradas são, então, completadas através da informação contextual.

Vários modelos tem sido propostos para representar informação contextual, como por exemplo: modelo de atributo e valor, modelo de linguagem mark-up, modelo de lógica de primeira ordem, modelo orientado a objeto, ontologia (SURACI; MIGNANTI; AIUTO, 2007).

Entretanto, segundo Sucari et al. (2007), ontologia é o modelo mais adequado para representar e raciocinar sobre informação de contexto, pelas seguintes razões: (i) ontologias habilitam o compartilhamento de conhecimento em sistemas abertos e dinâmicos; (ii) ontologias bem definidas semântica declarativa permite o raciocínio eficiente sobre informação de contexto; e (iii) ontologias habilitam a interoperabilidade do serviço e fornece o meio para serviço colaborarem de uma maneira não ambígua.

2.2.4.1 Trabalhos Relacionados

Broens et al. (2004) propõem uma abordagem para descoberta de serviço que usa ontologia para capturar a semântica da consulta do usuário, do serviço e da informação contextual que é considerada relevante para o processo de *matching*. A abordagem consiste em, dada uma consulta do usuário, filtra-se os serviços que não são do tipo de serviço desejado, após filtra-se os serviços que não possuem o conceito de saída do serviço desejada, por fim verifica-se se a entrada do serviço pode ser fornecida pelo usuário ou pode ser fornecida pelo provedor de contexto (por exemplo, quando o serviço precisa saber a localização do usuário, e essa é fornecida pelo provedor de contexto da localização do usuário). Estas três fases fazem a filtragem através do *matching* de conceitos baseado em lógica. Com isso, esse trabalho utiliza o contexto de forma a suprir a entrada do serviço, quando essa não está disponível. Ainda, por fazer uma implementação própria de ontologia, desta forma, não utiliza regras ou motor de inferência para descobrir novo conhecimento.

Maamar et al. (2005) sugerem uma abordagem de personalização de composição de SW que é baseada no contexto do usuário (status atual e suas preferências em termos de localização de execução e tempo de execução do serviço), do SW (status atual e restrições de execução), e do recurso (status atual). Com isso, quando um usuário seleciona um SW, ele segue com sua personalização de acordo com seu tempo e localização. Dessa forma, o contexto do recurso para o contexto do SW tem uma relação de ajuste de execução, identificando as restrições de execução do SW (por ex., tempo de execução, localização de execução, dependência de fluxo) versus a capacidade de execução do recurso (por ex., o próximo período de disponibilidade, política de agendamento) que em o SW irá ser executado. O contexto do usuário para o SW fornece um relacionamento de personalização, identificando as preferências dos usuários (por ex., quando e onde o SW necessita ser executado, e quando e onde o resultado da execução necessita ser retornado) versus a capacidade do SW para fornecer essas preferências (por ex., pode o SW ser executado em um determinado tempo ou uma determinada localização). Entretanto, essa abordagem não utiliza outros tipos de *matching* de SW para a descoberta do SW, apenas utiliza o contexto para personalizar o resultado de acordo com as preferências do usuário. Ainda, o contexto não é modelado como uma ontologia.

Kuck e Gnasa (2007) apresentam uma abordagem de descoberta de SW sensível ao contexto, direcionada a usuários de dispositivos móveis. De forma que a informação contextual pode ser, por exemplo, lugar visitado atual, temperatura atual, iluminação recente, etc. Essa abordagem usa técnicas de recuperação de informação para descoberta de SW (*matching* de

palavras) entre a consulta do usuário e a descrição do serviço em WSDL. A informação contextual, presente no perfil do usuário, é utilizada para ordenação dos resultados através de um modelo probabilístico de recuperação de informação.

Suraci et al. (2007) estabelecem uma arquitetura para descoberta de serviços semânticos sensíveis ao contexto, que utiliza um processo de filtragem sensível ao contexto. De forma que, o contexto do usuário, do serviço e do ambiente são levados em consideração. A descoberta de SW é formada por 3 passos de filtragem: filtro básico, que envolve a consulta básica do usuário e a descrição básica dos serviços; filtro semântico, que filtra os serviços de acordo com as características que o usuário quer; e filtro contextual, que aplica as requisições quanto ao contexto.

Mokhtar et al. (2008) estabelecem a abordagem EASY (*Efficient semAntic Service discoverY*) para dar suporte eficiente, semântico, de sensibilidade ao contexto e de qualidade de serviço para a descoberta de serviços utilizando protocolos de descoberta de serviços existentes. Nessa abordagem, é proposta uma linguagem, a EASY-L, para descrever o serviço em termos de entradas, saídas, capacidade e propriedades. Com isso, entradas, saídas e capacidade de um serviço são descritas utilizando conceitos ontológicos, e propriedades são descritas como expressões booleanas, que representam a informação contextual e qualidade do serviço. O *matching* de SW é feito através do *matching* de conceitos das entradas, saídas e capacidades, e da verificação das propriedades solicitadas retorna um valor verdadeiro.

Sheng et al. (2009) descrevem uma plataforma para desenvolvimento rápido de SW com consciente de contexto, a *ContextServ*. Essa plataforma fornece ao modelador uma interface visual para especificar um SW sensível ao contexto, utilizando o *ContextUML* (SHENG; BENATALLAH, 2005). A partir do diagrama *ContextUML* é feito um mapeamento para BPEL (*Business Process Execution Language*), linguagem para especificar processos executáveis, através de um conjunto de regras de mapeamento. Entretanto, essa abordagem não trata do problema de descoberta de SW, apenas especifica qual serviço selecionar dependendo do contexto.

Sherchan et al. (2012) propõem uma abordagem para design de Interface de Usuário com um Serviço, para criar interfaces que automaticamente se adaptam as condições do usuários bem como as da aplicação. Essa abordagem aproveita as informações de contexto do usuário, do dispositivo e da localização geoespacial, para permitir fácil acesso aos serviços, adaptando a forma de apresentação (ou seja, a interface de usuário) para se adequar ao seu contexto. Dessa forma, informação contextual relacionada com tipo de interface do usuário é implicitamente misturada com a solicitação do serviço.

Han et al. (2014) apresentam um sistema de automação de construção adotando o paradigma SOA, onde informação contextual é coletada, processada, e enviada para um mecanismo de composição para coordenar serviços apropriadamente baseado no contexto, no plano de composição, e regras de política predefinidas. De forma que, um coletor de contexto monitora o ambiente, e envia dados a um processador, que verifica com as regras de política predefinidas se precisa fazer uma composição de serviços. Se precisar, o motor de inferência coleta as políticas de restrições para criar um serviço composto apropriado.

A seguir, no Quadro 2.2, é apresentado um visual geral dos trabalhos estudados e descritos nesta seção, onde: a primeira coluna indica o trabalho; a segunda, terceira e quarta colunas indicam o tipo de *matching* de SW utilizado para complementar a abordagem; a quinta coluna indica como foi modelado o contexto; a sexta coluna indica como o contexto foi implementado (“própria” indica que o trabalho implementou a sua própria maneira); a sétima coluna indica se usa motor de inferência (no caso em que o contexto foi modelado em uma linguagem ontológica); e a última coluna indica para que se utilizou contexto no trabalho.

Quadro 2.2 – Comparativo dos trabalhos relacionados de *matching* de SW baseado em pragmático

Trabalho	Matching de SW			Contexto		Motor de Inferência	Usa contexto para
	P	Si	Se	Modelagem	Implementação		
Broens et al. (2004)	-	-	x	ontologia	própria	-	parâmetros
MAAMAR et al. (2005)	-	-	-	-	própria	-	solicitação
Kuck e Gnasa (2007)	x	-	-	-	própria	-	solicitação
Suraci et al. (2007)	x	-	x	ontologia	própria	-	solicitação
Mokhtar et al (2008)	-	-	x	ontologia	OWL	-	solicitação
Sheng et al. (2009)	-	-	-	-	própria	-	solicitação
Sherchan et al (2012)	-	-	x	ontologia	própria	-	solicitação
Han et al. (2014)	-	-	-	ontologia	OWL	x	solicitação/ parâmetro

Fonte: quadro criado pela autora

2.3 Processo de Negócio

Um processo de negócio (PN), ou em inglês *Business Process* (BP), consiste em um conjunto de atividades que são executadas e coordenadas em um ambiente organizacional e técnico, que visa alcançar um objetivo de negócio (WESKE, 2012).

Um modelo de PN, ou em inglês *Business Process Model* (BPM), consiste em um conjunto de modelos de atividades e restrições de execução entre elas (WESKE, 2012).

Segundo Fabra et al. (2012), os modelos de PN podem ser modelados em três níveis de abstração: nível de Modelo Independente de Computação, ou em inglês *Computation Independent Model* (CIM); nível de Modelo Independente de Plataforma, ou em inglês *Platform Independent Model* (PIM); e nível de Modelo Especifico de Plataforma, ou em inglês *Platform Specific Model* (PSM).

Conforme esses níveis de abstração, as linguagens para modelar PN podem ser classificadas da seguinte maneira (FABRA et al., 2012):

- Nível CIM: *Business Process Model and Notation 2.0* (BPMN 2.0), *Architecture of Integrated Information Systems* (ARIS);
- Nível PIM: *Business Process Definition Metamodel* (BPDM), Java 2 Platform Enterprise Edition (J2EE SUN);
- Nível PSM: *Business Process Modelling Language* (BPML), *Electronic Business using eXtensible Markup Language* (comumente conhecido como e-business XML, or formalmente como ebXML), *Web Services Business Process Execution Language* (WS-BPEL ou BPEL), *Web Services Choreography Description Language* (WS-CDL).

Para o nível CIM, o BPMN 2.0 é padrão *de-facto* para representar em forma gráfica os processos que ocorrem em praticamente todos os tipos de organização (CHINOSI; TROMBETTA, 2012) (FABRA et al., 2012).

O BPMN fornece uma notação gráfica que é fácil de entendimento para todos os participantes em um PN, dos analistas de negócio aos desenvolvedores, especificando diagramas de PN por meio de fluxos, eventos, atividades e resultados (FABRA et al., 2012).

Contudo, os rótulos das atividades de um modelo de PN, normalmente são representadas por meio de texto livre, o que pode gerar problemas de compreensão. Para resolver isso, surgiram abordagens que utilizam ontologias nos modelos de PN, de forma a aumentar a compreensão para as partes interessadas e possibilitar a compreensão dos dados por parte de máquina (MANZORR; CEUSTERS; RUDNICKI, 2007).

Uma abordagem que utiliza ontologia no modelo de PN foi proposta por Gassen et al. (2012), onde cada rótulo de atividade é composto por uma tripla <Sujeito, Ação, Objeto>, de forma que: Sujeito é quem executa a atividade, Ação é o que é executado pela atividade, e o Objeto é quem ou o que sofre a ação. Ainda, Sujeito e Objeto são relacionados com conceitos ontológicos (conceito ou classe), enquanto Ações são relacionadas com relacionamentos (propriedade do objeto) entre os conceitos. Com isso, fornecendo mais semântica aos modelos do PN.

Passando para o nível PSM, o BPEL é o padrão de-fato para implementação do PN sobre a tecnologia de SW (OUYANG et al., 2006). Em BPEL, cada atividade do PN é implementada através de um SW (VERNER, 2004).

Entretanto, de acordo com Verner (2004), o BPEL e demais linguagens do nível PSM dificultam a análise de negócio nos estágios iniciais de desenvolvimento do processo. De forma que acarreta o aumento da lacuna entre o analista de negócio e o desenvolvedor de software, que é uma séria limitação no campo BPM (FABRA et al., 2012).

Desta forma, para tentar contorna esse problema, surgiram abordagens para transformar modelos descritos em BMPN para modelos em BPEL. Exemplos de trabalhos pode-se citar: Weidlich et al. (2008), Ouyang et al. (2009), Mazanek e Hanus (2011).

2.4 Smart Home e Home Care com foco em Coordenação de Serviços

Conforme Ariza et al. (2013), sistema de Inteligência do Ambiente, em inglês *Ambient Intelligence* (AmI), é aquele que pode aumentar a sensibilidade ao contexto através do uso de sensores embutidos em objetos do cotidiano. Sistemas AmI precisam de esperteza e conectividade, pois desta forma, decisões podem ser tomadas referente a toda a informação disponível.

Segundo Cook et al. (2009) as seguintes funções são básicas para AmI:

- **Sensoriamento:** O uso de sensores é vital para o AmI, uma vez que eles são a chave que relaciona o poder computacional com aplicações físicas. Os sensores são usados para medir a posição, para detecção químicas e humidade, e para determinar as leituras de luz, radiação, temperatura, som, tensão, pressão, posição, velocidade e direção, e para apoiar no sensoriamento fisiológico para a monitoramento da saúde.
- **Raciocínio:** Sensoriamento e atuação fornecem a ligação entre os algoritmos de inteligência e o mundo real onde eles operam. Para fazer com que eles algoritmos sejam

responsivos, adaptáveis, e benéficos ao usuário, um número de tipos de raciocínio deve ser feito. Raciocínio abrange modelagem de usuário, a previsão e reconhecimento da atividade, tomada de decisão, e raciocínio espaço-tempo.

- Atuação: Sistemas de AmI fornecem mecanismos pelo qual ações possam ser executadas.

Essas funções básicas de sistemas AmI demonstram a dependência dessa área sobre a sensibilidade ao contexto (Cook et al., 2009). A sensibilidade ao contexto pode ser definida como a habilidade da aplicação para se adaptar as mudanças e responder de acordo com o contexto de uso (KJELDSKOV; SKOV, 2004). De acordo com Paganelli e Giuli (2011), ontologias são uma tecnologia chave para modelagem de contexto.

Para Machado et al. (2014), *Smart Home* é um campo que está emergindo como AmI que foca nas características do usuário. *Smart Home*, ou Domótica, é usado geralmente para se referir a qualquer ambiente, de casa ou trabalho, que foi cuidadosamente construído para ajudar as pessoas na realização de atividades (CHAN et al., 2008).

Um dos objetivos de sistemas *Smart Home* é melhorar o acesso ao cuidado em casa para idosos e pessoas com limitações (Sistemas *Home Care*) (CHAN et al., 2008). Sistemas *Home Care* podem ser definidos como uma tecnologia usada para dar suporte a realização de tarefas de rede, fornecendo os meios para coletar, distribuir, analisar e gerenciar informações relacionadas aos cuidados (MCGEE-LENNON, 2008), visando capacitar pessoas que necessitam de assistência para que continuem a viver em suas próprias casas, enquanto que sua saúde e autonomia se deterioram (AUVINEN et al., 2011).

Para satisfazer os desejos do usuário e garantir o seu conforto e segurança, a casa tem que ser capaz de apresentar funcionalidades complexas mais que apenas acionar um único atuador ou uma sequência predeterminada de atuadores (KALDELI et al., 2013).

Dessa forma, a coordenação de sensores e atuadores é um dos desafios dessas áreas. Ainda, muitos trabalhos representam sensores e atuadores como serviços, e os implementam como SW. A seguir, alguns trabalhos relacionados quem tratam do problema de coordenação de serviços em AmI são apresentados.

2.4.1 Trabalhos Relacionados

Gu et al. (2005) desenvolveram a arquitetura SOCAM (Service Oriented Context-Aware Middleware) para construção e protótipo de serviços sensíveis ao contexto. Essa arquitetura

fornece um modelo ontológico para contexto descrevendo pessoa, localização, entidade computacional e atividade. As atividades são acionadas através de regras que descrevem qual o contexto para sua execução. E cada atividade está vinculada a um único SW.

Ardissono et al. (2006) apresentam um arquitetura de um framework para integrar o gerenciamento de PN e execução de ações sensíveis ao contexto para dar suporte à personalização de gerenciamento de guias médicos em serviços Home Care. Nessa arquitetura, os guias médicos são descritos como um PN executado através do roteiro das atividades. Cada atividade possui um objetivo, um conjunto de condições para serem aplicadas, e está vinculada a um SW, sendo executadas levando em consideração o contexto.

Cho et al. (2007) propõem um linguagem para PN e um sistema de PN sensível ao contexto. Esse sistema usa contexto como condição da execução da atividade.

Wang e Turner (2008) relatam um sistema Home Care para dar suporte à personalização, adaptação e de confiança, baseado em uma abordagem de regras de política. Nessas regras pode-se especificar a condição, a ação ser executada, o dispositivo ou o tipo do dispositivo a executar a ação. Permite instanciação dinâmica dos serviços, mas se baseia que todos tenham a mesma interface de comunicação, não precisando assim realizar algoritmos de matching de SW.

Xu et al. (2009) apresentam um framework baseado em ontologia para facilitar a composição automática de aplicações apropriadas. O contexto é modelado através de um modelo ontológico, e regras descrevem a política. Dispositivos, suas funcionalidades e perfil do ambiente são instanciados na ontologia, e quando uma regra é acionada, a regra tenta executar um tipo de funcionalidade, através da ontologia é verificado qual dispositivo possui essa determinada funcionalidade, e então o aciona.

Lamine et al. (2010) descrevem um arquitetura de sistema para dar suporte a coordenação de serviços Home Care. Nessa arquitetura, sensores monitoram o ambiente, e quando encontram uma situação relevante acionam um PN para tratar da situação. Entretanto, não é esclarecido como as atividades do PN são executadas.

Valiente-Rocha e Lozano-Tello (2010) propõem o IntelliDomo, um sistema AmI baseado em ontologia para controle de sistema doméstico. O contexto é modelado utilizando ontologia, e regras descrevem o comportamento do sistema. Contudo, como o sistema age no mundo real não fica claro.

Paganelli e Giuli (2011) apresentam um modelo contextual baseado em ontologia e um sistema de gerenciamento de contexto para fornecer um framework orientado a serviço configurável e extensível para fácil desenvolvimento de aplicações de monitoramento e

manipulação de condições crônicas de pacientes. O modelo contextual proposto representa as pessoas e os lugares. De acordo com as regras criadas, são inferidas situações relevantes, o nível do alarme, e para quem deve receber o alarme.

Preuveneers e Berbers (2012) propõem um framework de automação residencial baseado em dispositivos inteligentes. Esse framework utiliza elementos de contexto, modelados em ontologia, e regras, para inferir dinamicamente ajustes na composição de sensores e atuadores. Dessa forma, o usuário final pode criar atividades (numa interface), e onde as atividades são uma composição de blocos de grupos de dispositivos (sensores e atuadores) com funcionalidade relacionada. A configuração informada no bloco é executada para todos os dispositivos do grupo.

Kao e Yuan (2012) introduzem USHAS (User-configurable Semantic Home Automation System), um sistema de automação residencial semântico que adota WS e BPEL para executar os processos automaticamente. Nesse sistema, pessoa, dispositivo, tempo, ambiente, evento e localização, são modelados em uma ontologia; e uma linguagem de descrição de processo foi criada (SHPL) para especificar precondições, variáveis, tempo de execução e fluxo de invocação. Tal linguagem de descrição de processo é utilizada em conjunto com o BPEL na execução do processo, de maneira que é verificado se as precondições em cada processo semântico são satisfeitas, uma vez que a ontologia é atualizada.

Kaldeli et al. (2013) apresentam uma arquitetura para AmI, de forma a fornecer ao usuário final uma série de serviços. Os dispositivos possuem um identificador único e são agrupados em um tipo abstrato, podendo possuir variáveis, e suas funcionalidades são armazenadas como ações e possuem pré-condições e efeitos. Objetivos são previamente estabelecidos também em termos de pré-condições e efeitos. Há o componente de sensibilidade ao contexto, que monitora continuamente o estado dos dispositivos, localização do usuário, coleta informações agregadas e notifica os módulos interessados. Com isso, sempre que um objetivo é disparado, seja pelo próprio usuário ou pelo motor de regras, é verificado qual o melhor dispositivo ou conjunto de dispositivos que atendem ao objetivo.

Han et al. (2014) propõem um sistema de automação de construção, onde informação contextual é coletada, processada, e enviada para um mecanismo de composição para coordenar dispositivos/serviço apropriados baseado no contexto, plano de composição e regras de política predefinidas. De forma que um coletor de contexto monitora o ambiente e envia dados a um processador, que verifica com as regras de política predefinidas se precisa fazer uma composição de serviços. Se precisar, o motor de inferência coleta as políticas de restrições para criar um serviço composto apropriado.

A seguir, o Quadro 2.3 resume os trabalhos apresentados nessa seção, em termos de: como é feita a modelagem de contexto (segunda coluna); como o contexto foi implementado (terceira coluna); se o trabalho uso motor de inferência (quinta coluna); qual é a forma de coordenação dos serviços (sexta coluna); se o trabalho seleciona dinamicamente o SW (sétima coluna); e se utiliza algum algoritmo de matching de SW (última coluna). Ainda, “NI” indica não informado, e “própria” indica que o trabalho implementou a sua própria maneira.

Quadro 2.3 – Comparativo dos trabalhos relacionados de coordenação de serviços em *Smart Home* e *Home Care*

<i>Trabalho</i>	<i>Contexto</i>		<i>Usa Motor de Inferência</i>	<i>Forma de coord.</i>	<i>Sel. din. o SW</i>	<i>Utiliza alg. de matching de SW</i>
	<i>Mod.</i>	<i>Imp.</i>				
Gu et al. (2005)	ontologia	OWL	X	regras	-	-
Ardissono et al. (2006)	-	própria	-	Modelo de PN / regras	-	-
Cho et al. (2007)	NI	RDF	-	Modelo de PN / regras	NI	-
Wang e Turner (2008)	-	-	-	regras	X	-
Xu et al. (2009)	ontologia	NI	X	regras	X	-
Lamine et al. (2010)	-	-	-	Modelo de PN	NI	-
Valiente-Rocha e Lozano-Tello (2010)	ontologia	OWL	X	regras	NI	-
Paganelli e Giuli (2011)	ontologia	OWL	X	regras	NI	-
Preuveneers e Berbers (2012)	ontologia	OWL	X	regras	X	-
Kao e Yuan (2012)	ontologia	OWL	X	Modelo de PN/ regras	-	-
Kaldeli et al. (2013)	-	própria	-	regras	X	-
Han et al. (2014)	ontologia	OWL	X	regras	X	-

Fonte: quadro criado pela autora

2.5 Resumo do Capítulo 2

Neste capítulo foram apresentados os principais conceitos, tecnologias, técnicas e trabalhos relacionados a este trabalho.

A Seção 2.2 apresentou 4 formas de *matching* de SW: palavras, sintaxe, semântica, pragmático. Entretanto, este trabalho não irá utilizar o *matching* de SW baseado em palavras,

pois este tipo de *matching* geralmente só é utilizado quando não se tem a descrição semântica do serviço, e aqui teremos (ver Seção 3.2 e Seção 5.1.3). Ainda, para se ter uma granularidade mais fina do *ranking*, será utilizado uma abordagem não-lógica para o *matching* semântico de SW. E como mostrado no Quadro 2.1, as abordagens *matching* semântico de SW não baseadas em lógica utilizam técnica de *matching* de ontologia. Dessa forma, a técnica de *matching* semântico de SW baseada em não-lógica que este trabalho utilizará é a abordagem que melhor obteve resultado segundo a Tabela 2.1, a abordagem de Lin (1998) com IC computado como em Zhou et al. (2008).

Com os trabalhos relacionados à área de coordenação de serviços em *Smart Home* e *Home Care*, foi possível verificar as duas principais abordagens para coordenação de serviços: modelo de processo de negócios e regras. Entretanto, conforme Weske (2012), modelos de PN em notação gráfica facilitam a comunicação sobre o processo, possibilitando que diferentes pessoas envolvidas possam se comunicar de maneira eficiente, refinando e melhorando o modelo.

Ainda, a utilização de contexto é fundamental neste trabalho, tanto para o *matching* pragmático de SW como para a coordenação de serviços, e, segundo Sucari et al. (2007), ontologia é a forma mais adequada de representar o contexto.

Embora alguns dos trabalhos relacionados à área de coordenação de serviços em *Smart Home* e *Home Care* selecionem dinamicamente um serviço para executar uma ação designada, esses trabalhos ainda não se preocupam em como instanciar dinamicamente esses serviços. É esse ponto que este trabalho visa contribuir, pois desta forma, novos serviços podem ser incorporados de maneira transparente.

3 CENÁRIO MOTIVACIONAL E ABORDAGEM PROPOSTA

A seguir é apresentado o cenário motivacional deste trabalho a fim de melhorar o entendimento da problemática que o cerca, e, após, é explicada a abordagem proposta para resolver o problema.

3.1 Cenário Motivacional

Machado et al. (2013) descrevem um cenário em que aplicações de *Home Care* que usa modelo de PN para gerenciar seu comportamento. Esses PN descrevem orientações para gerenciamentos de situações de interesse, que são aquelas situações relevantes para o contexto de uma aplicação. Assim, quando uma situação de interesse é detectada, o sistema aciona a execução do PN que endereça tal situação.

Dessa forma, ambientes inteligentes equipados com sensores e atuadores, representados por serviços Web, poderiam ser gerenciados por um controlador, que é responsável pela execução do PN, possibilitando com isso uma vida mais adequada e tranquila para as pessoas do ambiente gerenciado pela aplicação de *Home Care*.

Nesse cenário, modelos de PN são especificados em um alto nível de abstração. Com isso, as atividades do modelo de PN devem ser instanciadas e adaptadas de acordo com o ambiente de cada casa, sendo gerenciadas por um controlador. Esse controlador precisa identificar a funcionalidade/operação do serviço Web mais adequada para executar cada atividade do PN. Por exemplo, o controlador poderia ter que executar uma atividade que pede para enviar um alerta ao paciente informando que está na hora dele ingerir sua medicação.

Uma vez que uma casa possui muitos dispositivos atuadores que poderiam executar tal atividade (por exemplo, rádio, televisão ou celular), e o paciente pode ter restrições específicas (por exemplo, surdez), televisão e celular poderiam ser a melhor opção para enviar o alerta para o paciente. Contudo, se o paciente não está no mesmo cômodo onde a televisão ou o celular estão, outro dispositivo mais próximo, deveria ser selecionado. O problema, neste caso, torna-se a instanciação de atividades de atuação em um modelo de PN de maneira sensível ao contexto através de funcionalidade de dispositivos domésticos, representados por serviços Web em ambiente de *Home Care*. E esse é o problema que este trabalho visa tratar.

3.2 Abordagem Proposta

A abordagem proposta neste trabalho para tratar o problema de instanciação de atividades de atuação em um modelo de PN, em ambiente de *Home Care*, através de funcionalidade de dispositivos domésticos, representados por serviços Web, de maneira sensível ao contexto, pressupõe os seguintes requisitos:

- Para cada situação de interesse, existe um modelo de PN, no nível CIM (Seção 2.3), que visa gerenciar tal situação;
- Para cada dispositivo doméstico automatizado, existe um serviço Web que o representa e suas funcionalidades juntamente com sua descrição semântica;
- Existe um controlador que gerencia a casa;
- Quando um novo dispositivo doméstico automatizado é adquirido, o controlador já reconhece o serviço Web que o representa e tem acesso à sua descrição semântica.

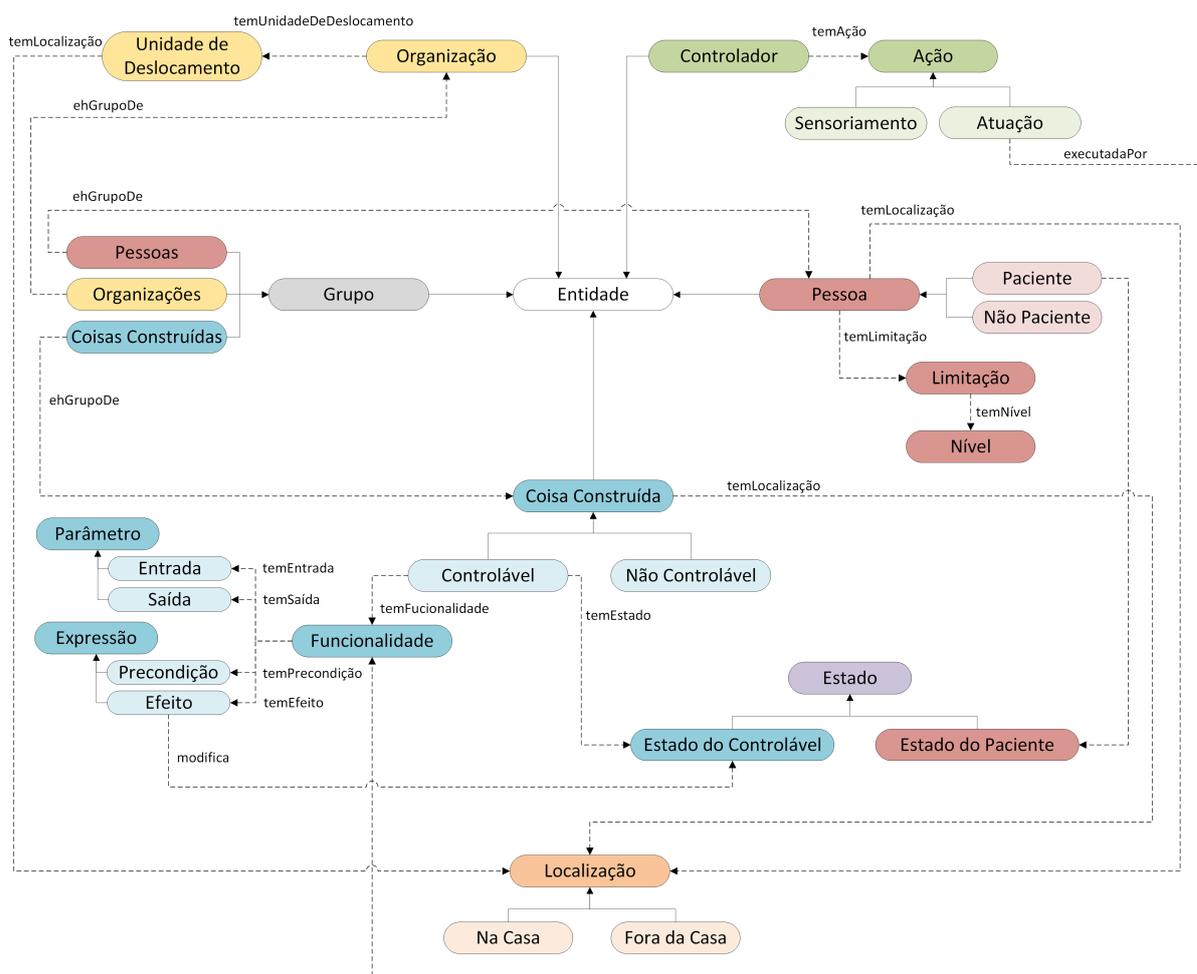
Com isso, a abordagem proposta é composta por:

- Um modelo ontológico para aplicações *Home Care*;
 - Estratégia de rotulação de atividade em modelo de PN;
 - Estratégia de instanciação (*framework*) de atividades de atuação do modelo de PN.
- Esses elementos são descritos nas subseções seguintes.

3.2.1 Modelo para aplicações *Home Care*

Visto que em um ambiente de *Home Care* é necessário manter informações sobre as pessoas que vivem no ambiente, incluindo suas características (incluindo habilidades e limitações), dispositivos disponíveis e organizações, o modelo observado para o desenvolvimento da abordagem proposta deve contemplar:

- Pessoa: caracteriza uma pessoa, suas limitações e sua localização;
- Organização: as organizações envolvidas (por exemplo, provedor de saúde);
- Localização: representa os cômodos dentro da casa, em um alto nível de abstração;
- Dispositivo: descreve os dispositivos, suas funcionalidades, parâmetros e restrições;
- Controlador: descreve as capacidades do controlador da casa.

Figura 2 – Visão geral do modelo para aplicações *Home Care*

Fonte: imagem criada pela autora

A Figura 2 mostra uma visão geral resumida do modelo proposto para aplicações *Home Care*, onde cada cor e suas variações de tonalidade representam um conjunto do modelo: vermelho representa o modelo de Pessoa; verde, o modelo de Controlador; azul, o modelo de Dispositivo; laranja, o modelo de Localização; amarelo, o modelo de Organização. A cor branca (Entidade), cinza (Grupo) e roxo (Estado) não representa um conjunto do modelo, são conceitos que são compartilhados por mais de um conjunto do modelo.

Esse modelo é uma base para um modelo ontológico, ou seja, não é uma ontologia de domínio completa e finalizada, mas é uma base para nova ontologia ou uma parte de uma ontologia. Visto que esse modelo tem como objetivo auxiliar na instanciação das atividades em um modelo de PN, desta forma o modelo apenas representa os conceitos e relacionamentos importantes para atingir tal objetivo.

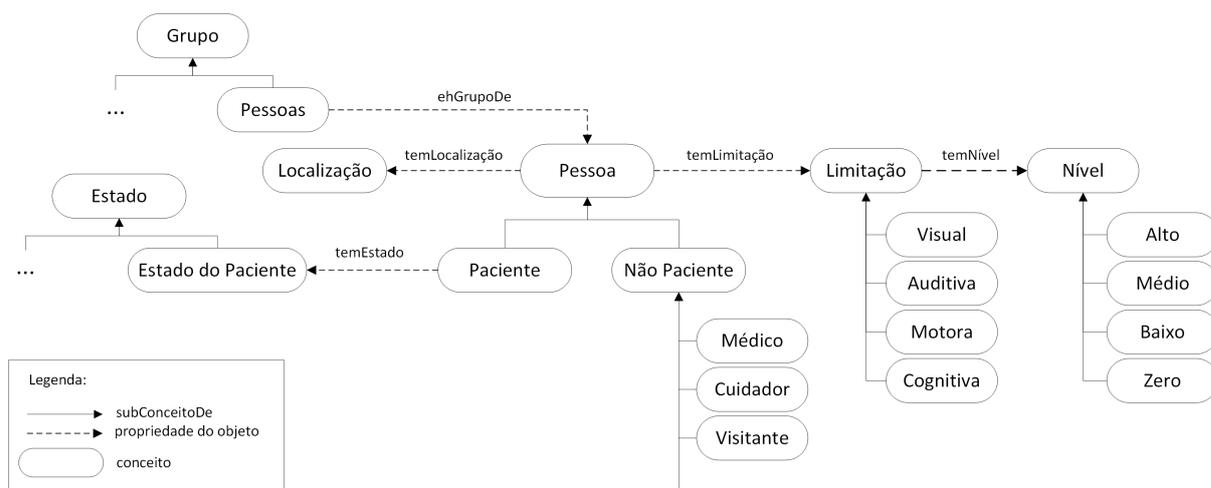
Tal modelo ontológico será detalhado nas próximas seções. A convenção adotada neste trabalho é que nomes de conceitos ontológicos terão a primeira letra em maiúsculo e estarão

entre aspas de forma a distingui-las de seus respectivos indivíduos do mundo real; e os nomes dos relacionamentos serão escritos sem espaço e estarão em *itálico>*.

3.2.1.1 Pessoa

As informações apresentadas no modelo de Pessoa são simples e contém somente o tipo da pessoa relativo ao domínio *Home Care*, suas limitações, o nível da limitação e sua localização dentro da casa.

Figura 3 – Modelo de Pessoa



Fonte: imagem criada pela autora

Como mostrado na Figura 3, o conceito “Pessoa” é categorizado em vários sub-conceitos (“Médico”, “Cuidador”, “Visitante”, ...) de modo a explicitar qual é o tipo da pessoa no domínio *Home Care*. Pessoa possui um único relacionamento com “Localização”, que representa o lugar dentro ou fora da casa (Seção 3.2.1.3), pois uma pessoa só pode estar em um local por vez. “Pessoa” ainda tem um relacionamento para cada tipo de “Limitação”, que representa uma incapacidade.

A “Limitação” (incapacidade) pode ser de quatro tipos, retirados de Crow (CROW, 2008):

- “Visual”: pessoas com limitação visual podem ter dificuldades em entender texto escrito e conteúdo gráfico, e ambos são os principais meios para apresentar informação;

- “Auditiva”: pessoas com este tipo de limitação sua habilidade auditiva reduzida para escutar determinadas frequências ou possuem dificuldades em todos os níveis de frequência, o que afeta a sua recepção de informação auditiva;
- “Cognitiva”: envolve memória, percepção, capacidade de resolver problema, conceptualização de mudanças, este tipo de limitação afeta amplamente a interação com a pessoa, pois informação poderia ter que ser repetida mais vezes;
- “Motora”: pessoas com limitação motora tem sua capacidade de coordenação motora reduzida, afetando o modo como a pessoa interage com dispositivos.

Embora o trabalho de Crow (2008) relate o impacto dessas limitações na aprendizagem online, entendeu-se, neste trabalho, que essas limitações também são importantes de ser levadas em consideração na vida cotidiana de uma pessoa.

Ressalta-se que não somente “Paciente” tem “Limitação”, mas “Pessoa”, não importando se é do tipo “Médico”, “Visitante”, etc. Uma vez que mesmo pessoas jovens podem ter limitações. Dessa forma, para se emitir uma mensagem sonora para o cuidador, deve-se primeiro verificar se ele não possui problemas auditivos.

Cada “Limitação” tem um único relacionamento com “Nível”, que descreve o nível da limitação, podendo ser: “Alto”, “Médio”, “Baixo” ou “Zero” (não possui a limitação).

Como descrito anteriormente, “Pessoa” tem um relacionamento para cada tipo de “Limitação”, mesmo que uma determinada pessoa não possua todas os tipos de limitações descritas no mundo real, na modelagem, “Pessoa” é representada como tendo todos os tipos de “Limitação” mas tendo um relacionamento com “Zero” como nível da limitação. Isso porque posteriormente é usado um motor de inferências para descobrir informação sobre o contexto modelado, e o motor de inferências adota a suposição de mundo aberto, ou seja, assume que o que é explícito é verdade mas não necessariamente toda a verdade (ZAMBORLINI; GUIZZARDI, 2010). Dessa forma, se um indivíduo “Pessoa” não tiver nenhum relacionamento com “Limitação” (pois no mundo real o indivíduo em questão não tem limitações), o motor de inferência não pode afirmar que o indivíduo não tenha nenhuma limitação, pois o motor só consegue fazer inferência sobre o que foi dito explicitamente. Por isso, optou-se por explicitar que uma determinada limitação é nula (através do relacionamento com o “Nível” do tipo “Zero”).

Ainda, “Pessoa” do tipo “Paciente” (a partir de agora para simplificar a escrita, o conceito e seus subconceitos serão escritos como “Conceito Subconceito”, neste caso por exemplo, “Pessoa Paciente”) tem vários relacionamentos com “Estado do Paciente”, ou seja, os estados em que o paciente se encontra, por exemplo: “Agitação”, “Choque”, “Estado agudo de

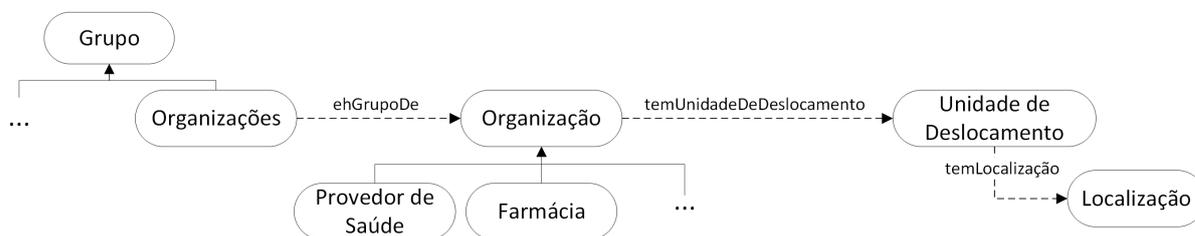
diabetes”, etc. Nada impede que o “Paciente” esteja com vários tipos de estados ao mesmo tempo.

Por fim, “Pessoas” é um subtipo do conceito “Grupo” (detalhes deste conceito na Seção 3.2.1.6) e possui um relacionamento único *ehGrupoDe* com “Pessoa” que indica o coletivo de qual conceito “Pessoas” representa.

3.2.1.2 Organização

O modelo de Organização modela entidades que não são parte da casa, mas de certa forma interagem com o controlador, por exemplo, o provedor de saúde, a farmácia, etc.

Figura 4 – Modelo de Organização



Fonte: imagem criada pela autora

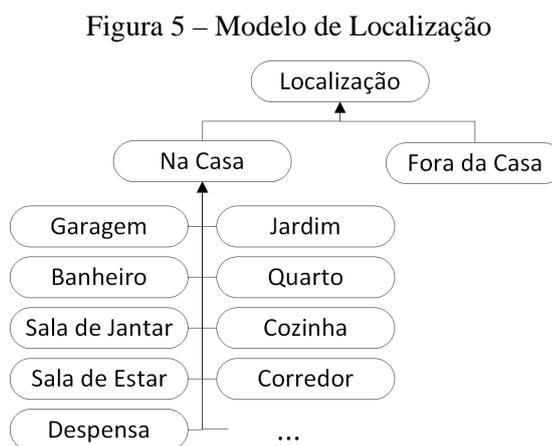
Como mostrado na Figura 4, o conceito “Organização” pode ser de vários tipos (“Provedor de Saúde”, “Farmácia”, etc.). Ainda, uma instância de “Organização” pode ter nenhuma ou várias “Unidade de Deslocamento”, que representa uma viatura enviada pela Organização.

Nesse modelo somente “Unidade de Deslocamento” possui o relacionamento único com “Localização” (Seção 3.2.1.3), uma vez que a localização de qualquer “Organização” pode ser num local diferente da casa do paciente (“Fora da Casa”), mas a “Unidade de Deslocamento”, que se move, poderia estar na casa do paciente. Por exemplo, se uma determinada organização necessita enviar uma unidade até a casa do paciente (por exemplo, uma ambulância), e se quer saber sua localização (ou seja, se a unidade já chegou na casa do paciente, “Na Casa”, ou não, “Fora da Casa”).

Similarmente ao modelo de Pessoa, o modelo de Organização também possui um tipo do conceito Grupo: “Organizações”, que indica o coletivo de “Organização” através do relacionamento único *ehGrupoDe* com “Organização”.

3.2.1.3 Localização

O modelo de Localização (Figura 5) especifica em alto nível de abstração se algo ou alguma coisa está localizada dentro da casa (“Na Casa”) ou não (“Fora da Casa”), e, se está dentro da casa, especifica onde se encontra (“Garagem”, “Jardim”, “Quarto”, “Cozinha”, etc.). A parte que caracteriza a localização dentro da casa foi retirada da ontologia DogOnt (BONINO; CORNO, 2008).



Fonte: imagem criada pela autora

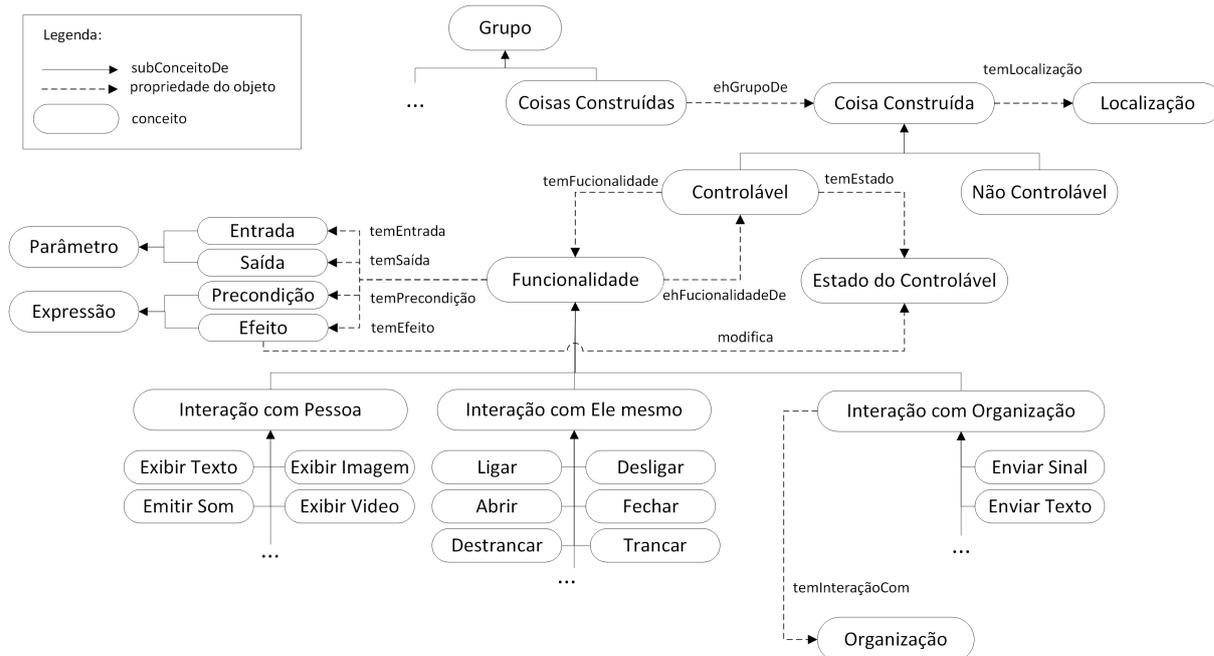
3.2.1.4 Dispositivo

O modelo de Dispositivo (Figura 6) é usado para categorizar as coisas construídas em controlável ou não, e para explicitar as funcionalidades das coisas controláveis.

Os conceitos de “Coisa Construída”, “Controlável” e “Não Controlável”, “Funcionalidade” e “Estado do Controlável” foram retirados da ontologia DogOnt (BONINO; CORNO, 2008). O conceito “Coisa Construída” refere-se a qualquer coisa disponível dentro da casa. O conceito “Não Controlável” refere-se a qualquer objeto que não possa ser gerenciado, e “Controlável” refere-se a qualquer objeto que possa ser controlado. Ainda, existe o conceito “Coisas Construídas”, que é um tipo de Grupo que se relaciona com “Coisa Construída” através do relacionamento único *ehGrupoDe*, que indica a qual conceito esse é o coletivo.

Para facilitar a leitura, a partir de agora, o conceito “Controlável” será referenciado simplesmente como “Dispositivo”.

Figura 6 – Modelo do Dispositivo



Fonte: imagem criada pela autora

Dispositivo relaciona-se pelo menos uma vez com “Funcionalidade” e “Estado do Controlável”, isto porque para que um objeto seja controlável ele deve possuir pelo menos uma funcionalidade. O conceito “Estado do Controlável” refere-se a configurações internas que o dispositivo pode assumir, e “Funcionalidade” refere-se a o que o dispositivo pode fazer para mudar seu valor de estado. Por exemplo, uma lâmpada pode ter num determinado momento o estado *Ligado = falso*, mas uma vez que sua funcionalidade *ligar* é acionada, seu estado irá mudar para *Ligado = verdadeiro*.

Esse relacionamento do “Dispositivo” para com “Funcionalidade” é feito através do relacionamento *temFuncionalidade*, que é o relacionamento inverso de *ehFuncionalidadeDe*, que relaciona “Funcionalidade” e “Dispositivo”.

Com base na OWL-S (MARTIN et al., 2004), “Funcionalidade” relaciona-se zero ou mais vezes com os conceitos “Entrada” e “Saída”, ambos tipos de “Parâmetros”, e zero ou mais vezes com os conceitos “Precondição” e “Efeito”, que são subtipos de “Expressão”. O conceito “Entrada” representa o conceito que é recebido como parâmetro de entrada para a execução da funcionalidade; o conceito “Saída” representa um conceito que é gerado após a execução da funcionalidade. O conceito “Precondição” representa a regra que precisa atendida para a execução da funcionalidade, e o conceito “Efeito” representa a regra que muda o estado do dispositivo após a execução da funcionalidade.

O conceito “Funcionalidade” pode ser classificado de acordo com o objeto alvo da interação:

- “Interação com Pessoa”: quando se quer interagir com uma pessoa através do dispositivo;
- “Interação com Ele mesmo”: quando o dispositivo executa uma ação nele mesmo;
- “Interação com Organização”: quando se quer comunicar ou solicitar algo a uma entidade que está fora da casa.

As funcionalidades do tipo “Interação com Organização” possuem ainda relacionamento único *temInteraçãoCom*, que vincula a funcionalidade com o seu alvo em “Organização”.

Neste modelo, “Interação com Pessoa” é um tipo de “Funcionalidade”, desta forma, esse conceito herda os relacionamentos do conceito pai (*temEntrada*, *temSaída*, *temPrecondição*, *temEfeito*), e algumas características específicas de dada funcionalidade podem ser previamente estabelecidas.

Figura 7 – Exemplo de funcionalidade com características pré-determinadas



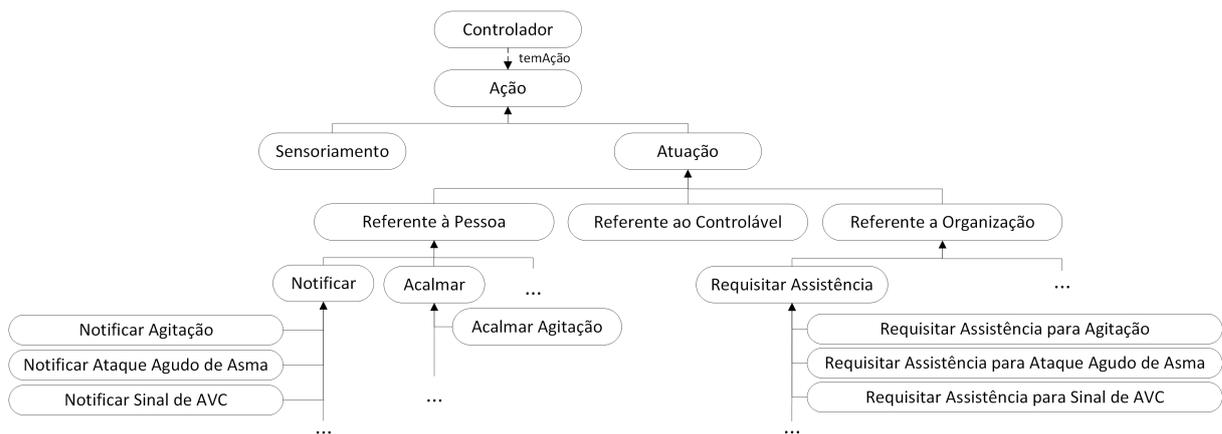
Fonte: imagem criada pela autora

Como mostrado na Figura 7, qualquer dispositivo que possua a funcionalidade “Exibir Texto” tem como “Precondição” a regra que informa que a pessoa alvo, aquela que o texto será mostrado (ou seja a entidade alvo), não pode ter limitação visual alta.

3.2.1.5 Controlador

O modelo de Controlador é usado para indicar as capacidades de um controlador e ajudar ao modelador a especificar o modelo de PN que irá executar as atividades apropriadas relacionadas a cada situação de interesse. Assim, como mostrado na Figura 8, o conceito “Controlador” possui um ou mais relacionamentos com o conceito Ação, que indica as ações que o controlador pode executar sobre o ambiente.

Figura 8 – Modelo de Controlador



Fonte: imagem criada pela autora.

Uma ação é entendida como algo que é feito, executado ou que acarreta em algo de maneira intencional, deliberada e com esforço (ZHU, 2004). No contexto deste trabalho, ação é definida como:

Definição 1: Ação é algo que uma entidade executa, faz, ou efetua, sendo ela manual ou automatizada.

Ainda, no caso específico do modelo de Controlador, as ações que ele pode executar podem ser categorizadas em ações de “Sensoriamento” e ações de “Atuação”, que correspondem respectivamente as ações de *Getting* e *Setting* (KALDELI et al., 2013).

Definição 2: Ação de Sensoriamento é aquela que devolve o estado de uma ou mais entidades entidade.

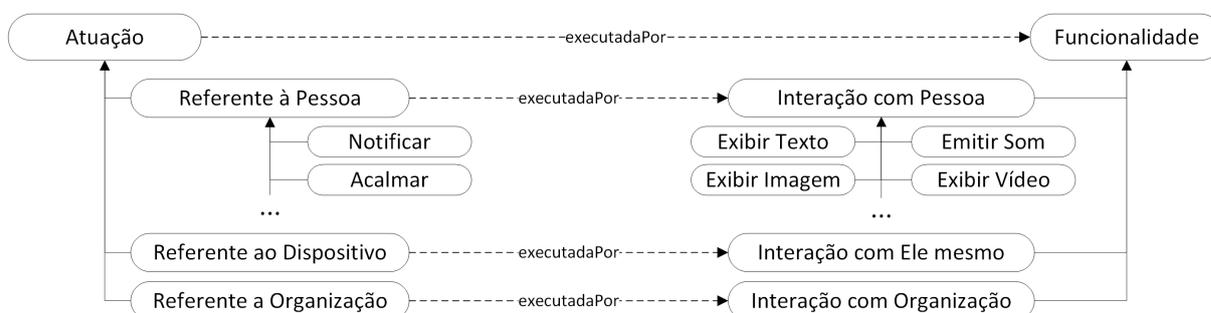
Definição 3: Ação de Atuação é aquela que muda o estado de uma ou mais entidade.

O conceito “Ação Atuação” pode ainda ser classificado considerando para quem ou em que ela é aplicada (Figura 8):

- “Referente a (uma) Pessoa”: refere-se a ação que é executada de forma a comunicar algo a uma pessoa;
- “Referente a (um) Dispositivo”: refere-se a ação que somente modifica um dispositivo;
- “Referente a (uma) Organização”: refere-se a ação que informa ou aciona uma organização.

Todas as ações de “Atuação” são executadas através de uma “Funcionalidade” de um “Dispositivo”. Dessa forma, quando ocorre uma ação de “Atuação Referente a (uma) Pessoa”, ela será executada através da “Funcionalidade Interação com Pessoa”; quando for uma “Atuação Referente a (um) Dispositivo”, ela será executada através da “Funcionalidade Interação com ele mesmo”; e quando for uma “Atuação Referente a (uma) Organização”, ela será executada através da “Funcionalidade Interação com Organização” (Figura 9).

Figura 9 – Ações de Atuação são executadas por algum tipo de funcionalidade do dispositivo

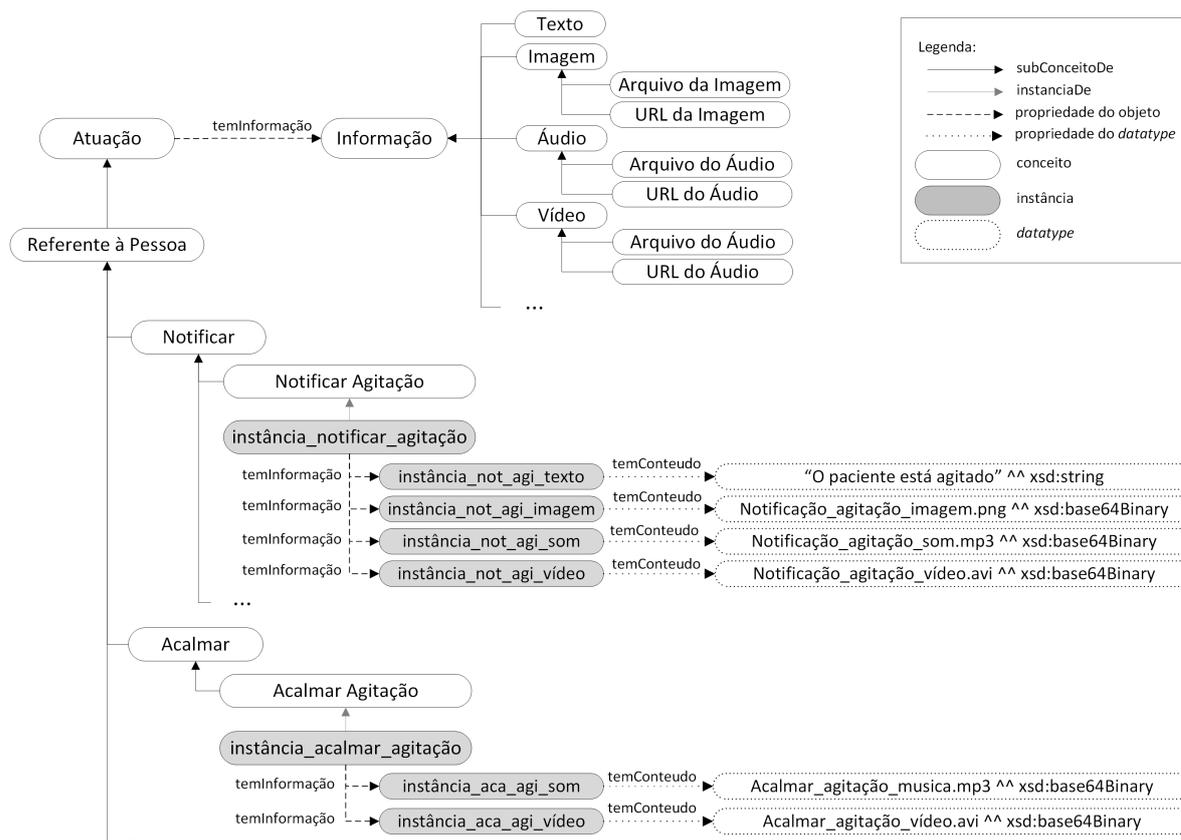


Fonte: imagem criada pela autora

Cada conceito folha do conceito ação de “Atuação” contem pelo menos um relacionamento com o conceito “Informação”, de modo que essa informação agregada a ação de “Atuação” possa permitir que se escolha mais de uma funcionalidade para executar tal ação. A figura seguinte ilustra esse aspecto: o conceito “Notificar Agitação” possui uma instância dos conceitos “Texto” (“instância_not_agi_texto”), “Imagem” (“instância_not_agi_imagem”), “Som” (“instância_not_agi_som”) e “Vídeo” (“instância_not_agi_vídeo”) (na Figura 10 o relacionamento *instanciaDe* foi omitido nessas instâncias para não prejudicar a visibilidade).

Por fim, toda instância de “Informação” possui um relacionamento único com um *datatype* que representa o conteúdo contido nessa instância (Figura 10).

Figura 10 – Conceitos folhas das Ações de Atuação possuem informação agregada



Fonte: imagem criada pela autora

Neste modelo é necessário pelo menos um tipo, mas não é necessário fornecer todos os possíveis tipos do conceito “Informação” para cada conceito folha de ação do tipo “Atuação”. A variedade de tipos do conceito “Informação” afeta na relação de quais funcionalidades podem executar tal ação. Por exemplo, a ação “Acalmar Agitação” (Figura 10) não possui todos os tipos do conceito “Informação” (“Texto”, “Imagem”, “Som”, “Vídeo”, ...), o que neste caso faz sentido porque é improvável que um paciente em estado de agitação se acalme com um texto ou com uma imagem. Dessa forma, somente dispositivos que possuem a funcionalidade “/emitir Som” ou “Exibir Vídeo” podem executar a ação “Acalmar Agitação”.

3.2.1.6 Conceitos Secundários

O conceito “Grupo” aparece nos modelos de Pessoa, Organização e Dispositivo, e tem como objetivo indicar que seus subconceitos são grupos de outros conceitos. Todo subconceito de “Grupo” (como “Pessoas”, “Organizações”, “Dispositivos”, “Portas”, etc.) tem um

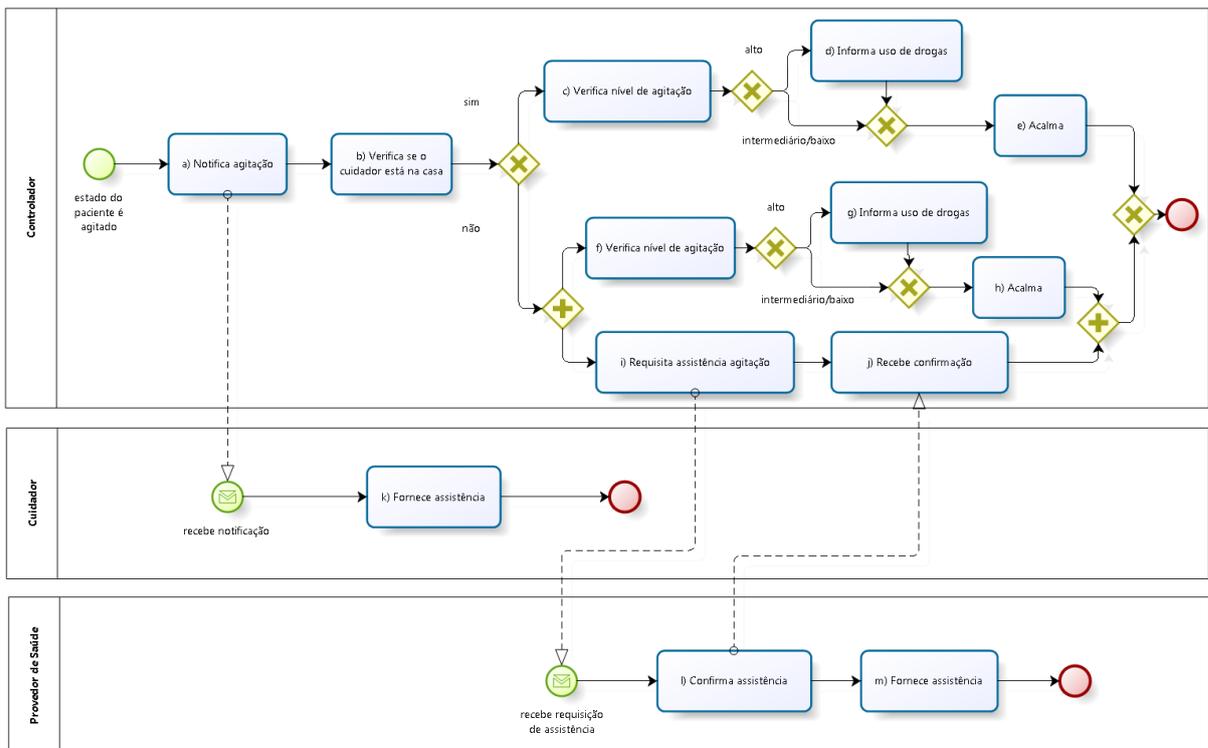
relacionamento único *ehGrupoDe* que o relaciona com o conceito ao qual ele é o coletivo. Por exemplo, “Pessoas” tem o relacionamento único *ehGrupoDe* com “Pessoa”.

Ainda, outro conceito secundário é “Entidade”. O conceito “Entidade” tem por objetivo indicar quais conceitos são entidades relevantes ao *framework*.

3.2.2 Rótulo de atividade de modelo de PN

Baseado na metodologia de Gassen et al.(2012), neste trabalho também se propõe que cada rótulo de atividade de um modelo de PN seja composto pela tripla <Sujeito, Ação, Objeto>. Contudo, neste trabalho, cada elemento da tripla está relacionado com um conceito ontológico, inclusive Ação. Dessa forma, a Ação do rótulo da atividade será relacionada com o conceito “Ação” do modelo de Controlador (Seção 3.2.1.5), que possui elementos relacionados associados a ela (instâncias do conceito “Informação”).

Figura 11 – Modelo de PN sem a utilização de tripla no *label* das atividades

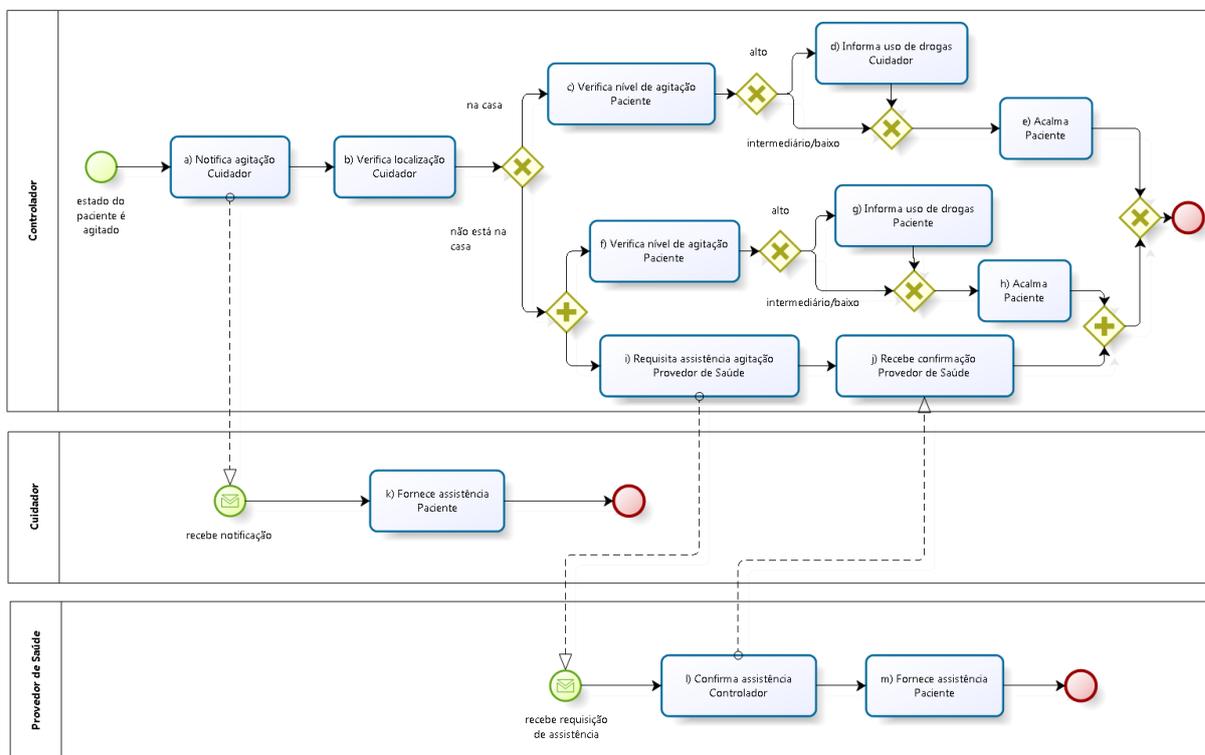


Fonte: imagem criada pela autora com base no trabalho de Gassen et al.(2012)

A Figura 11 mostra o modelo de PN para descrever o comportamento num cenário de *Home Care* para o estado de agitação do paciente, baseado no modelo de PN de Gassen et al.

(2012, p.5), e a Figura 12 mostra como fica esse modelo com a utilização da tripla no rótulo das atividades. Como em cada *pool* (divisão), em um modelo PN, as atividades descritas nele são realizadas e indicadas pelo nome do *pool*, o elemento Sujeito foi omitido da tripla, mas é indicado pelo nome do *pool*.

Figura 12 – Modelo de PN com a utilização de tripla no *label* das atividades

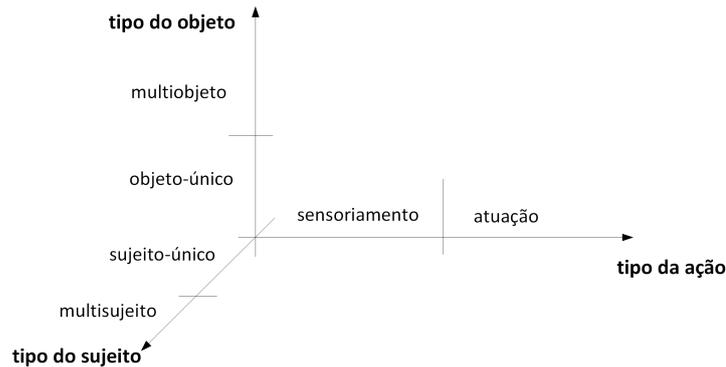


Fonte: imagem criada pela autora com base no trabalho de Gassen et al.(2012)

3.2.3 Instanciação das atividades de modelo de PN

Nesta seção explica-se em detalhes como são descritos os elementos das triplas de rótulo de atividades de modelo de PN. Também se descreve como, a partir deles e com o suporte do modelo para aplicações *Home Care* (Seção 3.2.1), se consegue instanciar atividades relacionadas. A figura seguinte apresenta a as possíveis classes de sujeito, ação e objeto e a relação entre elas.

Figura 13 – Classificação de sujeito, ação e objeto



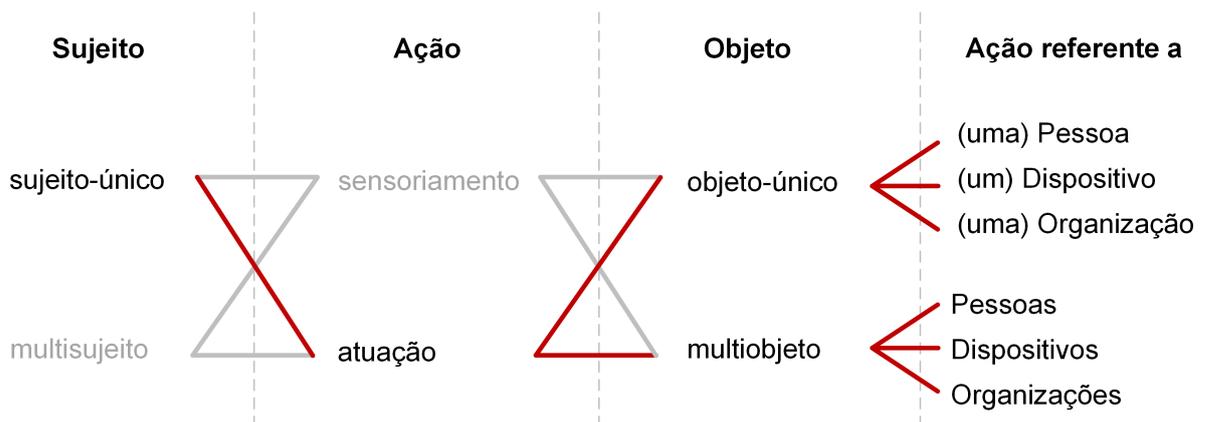
Fonte: imagem criada pela autora

Como demonstrado na Figura 13, o Sujeito pode ser classificado em sujeito-único ou multissujeito, onde sujeito-único indica que a ação é executada por apenas um indivíduo, e multissujeito indica que vários indivíduos executam a mesma ação. A Ação também possui duas classificações: sensoriamento e atuação (a descrição desses tipos foi feita na Seção 3.2.1.5). E, por fim, o objeto é classificado em objeto-único e multiobjeto, onde objeto-único indica que a ação é executada com um único objeto (ou indivíduo) como alvo, e multiobjeto indica que a ação executada tem vários objetos (ou indivíduos) como alvo.

A classificação de cada elemento da tripla do rótulo de atividade de modelo de PN é ortogonal, como indicado na Figura 13, ou seja, um não interferindo no tipo do outro. Por exemplo, o rótulo < “Controlador”, “Fecha”, “Porta”>, indica que o controlador (sujeito-único) deve fechar (ação) somente uma única porta (objeto-único).

Este trabalho foca apenas nas ações do tipo atuação em que o sujeito do rótulo é do tipo sujeito-único e o objeto pode ser tanto objeto-único como multiobjeto. O esquema do foco do trabalho é mostrado na Figura 14.

Figura 14 – Foco do trabalho



Além disso, o comportamento do controlador muda dependendo de quem ou do que a ação é referente. Por isso, nas próximas seções será explicado o funcionamento do *framework* para os seguintes casos de rótulo:

- <sujeito-único, ação-atuação, objeto-único> referente a (uma) Pessoa;
- <sujeito-único, ação-atuação, objeto-único> referente a (um) Dispositivo;
- <sujeito-único, ação-atuação, objeto-único> referente a (uma) Organização;
- <sujeito-único, ação-atuação, multiobjeto > referente a Pessoas;
- <sujeito-único, ação-atuação, multiobjeto > referente a Dispositivos;
- <sujeito-único, ação-atuação, multiobjeto > referente a Organizações.

Como listado acima, os casos só mudam em relação ao objeto e em quem ou o que a ação é referente. Desta forma, por simplificação, esses casos serão referidos apenas pelo tipo de objeto e a quem ou o que a ação é referente.

3.2.3.1 Objeto-único com ação referente a (uma) Pessoa

O *framework* proposto para a instanciação de atividades com rótulo do tipo objeto-único com ação referente a (uma) Pessoa é apresentado na Figura 15. Este *framework* considera a localização e as limitações da pessoa alvo da ação. Neste texto, utilizar-se-á a abreviação FuncDisp para indicar uma funcionalidade de um dispositivo.

De acordo com o *framework* (Figura 15), quando o controlador necessita executar uma determinada ação através de uma funcionalidade de um dispositivo doméstico disponível, precisa seguir os seguintes passos:

- a) Filtrar pela localização da pessoa: a partir do conjunto disponível de FuncDisp, o controlador seleciona apenas os que estão na mesma localização da pessoa-alvo ou em um ambiente próximo.
- b) Filtrar pelo tipo da funcionalidade: seleciona do conjunto dos FuncDisp retornado por (a) aqueles que possuem a funcionalidade do tipo que pode executar a ação (por exemplo, para a ação do tipo “Referente à Pessoa”, a funcionalidade que pode executá-la é a do tipo “Interação com Pessoa”).
- c) Filtrar de acordo com a limitação da pessoa: seleciona a partir do conjunto dos FuncDisp obtido em (b) considerando as limitações da pessoa alvo. Isso é feito verificando-se as precondições da funcionalidade (por exemplo, para usar a

funcionalidade “Exibir Imagem”, a pré-condição é a limitação visual da pessoa-alvo, que não pode ser alta).

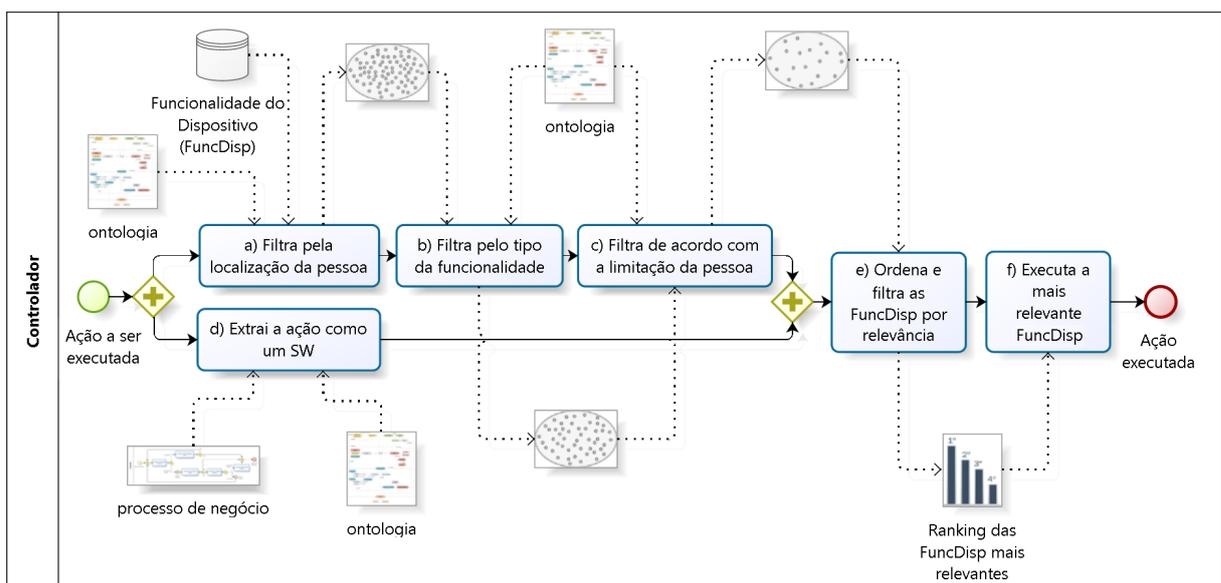
- d) Extrair a ação como um SW: um serviço Web abstrato *default* é definido a partir do PN e a ontologia envolvida. Neste passo, cria-se um ou mais serviço Web abstrato *default*, esse número corresponde ao número de instancias do conceito “Informação” contida na ação especificada (Seção 3.2.1.5). Por exemplo, a ação “Acalmar Agitação” possui duas instâncias de do conceito “Informação” (uma do subtipo “Som” e outra do subtipo “Vídeo”), assim, é definido dois serviços Web, uma para cada conceito de “Informação”, ou seja, cada serviço Web abstrato *default* usa a instância do conceito “Informação” como parâmetro de entrada. Dessa forma, todo o serviço Web abstrato *default* criado possui somente um elemento como parâmetro de entrada.
- e) Ordenar e filtrar os FuncDisp por relevância: utilizando o serviço Web abstrato *default* definido em (d), algoritmos de *matching* semânticos e sintáticos são executados no conjunto FuncDisp retornado em (c), após uma lista dos FuncDisp em ordem decrescente de relevância é criada, baseada no resultado retornado nos algoritmos de *matching*, e um limiar é aplicado para descartar os FuncDisp irrelevantes (aqueles que o controlador não consegue instanciar). Se mais de um serviço Web abstrato *default* foi criado em (d), os algoritmos de *matching* sintático e semântico de SW são executados para cada um, e o resultado de todos são ordenados na mesma lista. Detalhes de quais algoritmos de *matching* foram utilizados, ver na Seção 5.1.4.
- f) Executar o mais relevante FuncDisp: a FuncDisp mais relevante obtida na listagem de (e) é selecionada, verifica-se se as pré-condições da FuncDisp estão satisfeitas, se então, a FuncDisp executada, se não, a próxima FuncDisp é selecionada e verificada.

Resumidamente, o fluxo do *framework* começa quando a execução de uma dada ação é requisitada (“Ação a ser executada”, Figura 15). O controlador, então, verifica quais FuncDisp estão disponíveis (ou seja, possuem o estado ativo) e estão na mesma localização ou perto da pessoa alvo. Após esta seleção inicial, verifica-se quais dentre esse conjunto podem executar este tipo de ação em particular (como a ação é do tipo “Referente à Pessoa”, a funcionalidade que executá-la é a do tipo “Interação com Pessoa”), e quais podem ser utilizadas levando em consideração as limitações da pessoa alvo (analisando as pré-condições de cada funcionalidade). A partir desse conjunto de FuncDisp (que contem funcionalidades de dispositivos que estão na mesma localização ou próximos da pessoa-alvo, que estão ativos, que podem executar o tipo

da ação determinada, e que respeitam as limitações da pessoa alvo), é realizado os algoritmos de *matching* entre esse conjunto e o serviço Web abstrato *default* (obtido através do modelo de PN e a ontologia), pois não se sabe se o controlador consegue instanciar o serviço Web que representa cada FuncDisp, a partir da instância de “Informação” contida na ação especificada. Após, uma lista ordenada por relevância é criada com base no resultado desses algoritmos de *matching*. Por fim, o controlador seleciona o FuncDisp mais relevante, de acordo com os algoritmos de *matching*, verifica se as precondições desse FuncDisp estão satisfeitas, se estão executa-o, se não, seleciona o próximo FuncDisp da lista e verificado suas precondições.

Desta forma, através do modelo de PN e considerando as entidades envolvidas (“Controlador”, “Pessoa”, “Dispositivo”, “Organização”) e suas localizações, é possível instanciar a atividade do modelo de PN, selecionando a funcionalidade de um dispositivo que está na mesma localização ou próxima da pessoa alvo da ação da atividade, e levando em consideração as informações contextuais da pessoa (localização e suas limitações) e do dispositivo (localização e seus estados).

Figura 15 – Framework proposto para objeto-único com ação referente a (uma) Pessoa



Fonte: imagem criada pela autora

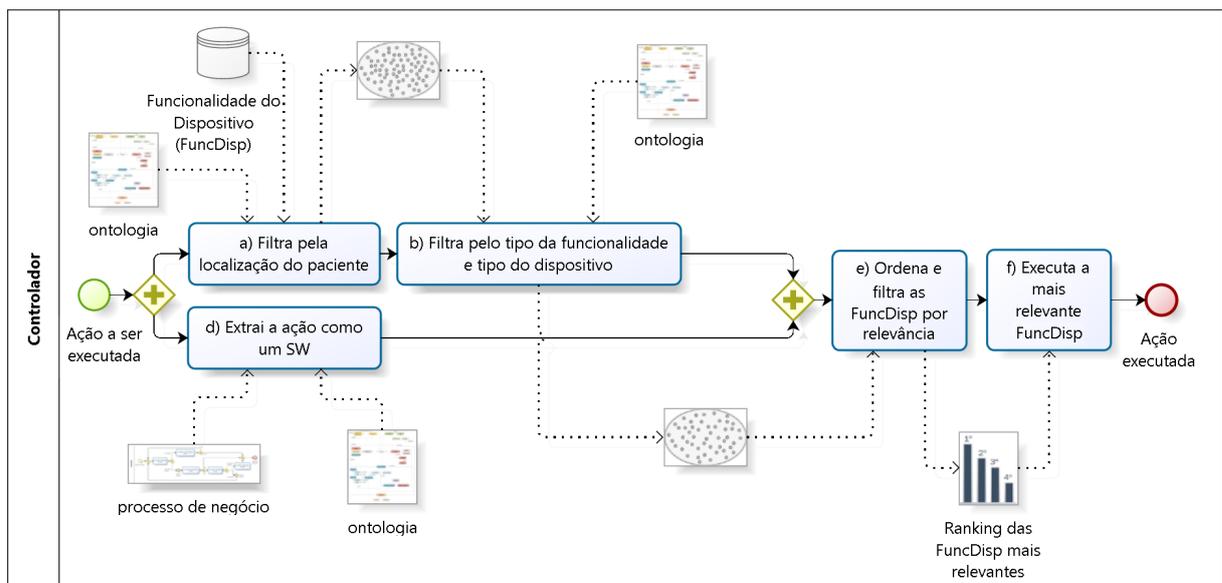
3.2.3.2 Objeto-único com ação referente a (um) Dispositivo

A Figura 16 mostra o *framework* proposto para a instanciação de atividades com rótulo do tipo objeto-único com ação referente a (um) Dispositivo. Este framework tem o objetivo de executar uma determinada ação a um tipo de dispositivo especificado no objeto-único, que

esteja perto do paciente, e esta ação não interage diretamente com uma pessoa. Por isso, esse *framework* apenas considera a localização do paciente. Por exemplo, <“Controlador”, “Abre”, “Janela”>, o que se quer é que o controlador (sujeito) abra (ação-atuação) a janela (objeto-único) que está no mesmo cômodo que o paciente, visto que este é um *framework* que visa dar apoio para a condição de vida do paciente.

Assim, a diferença desse *framework* ao que foi apresentado na Seção 3.2.3.1 é o passo (a), que agora vai filtrar pela localização do paciente; o passo (b), que filtra pela funcionalidade referenciada pela ação e pelo tipo de dispositivo indicado no objeto; e o passo (c) não é realizado, uma vez que não necessita comunicação com uma pessoa.

Figura 16 – Framework proposto para objeto-único com ação referente a (um) Dispositivo



Fonte: imagem criada pela autora

3.2.3.3 Objeto-único com ação referente a (uma) Organização

Este *framework* tem os passos de execução iguais ao do objeto-único com ação referente a Dispositivo (Seção 3.2.3.2). A única diferença é que o alvo da ação é uma organização.

3.2.3.4 Multiobjeto com ação referente a Pessoas

O *framework* para multiobjeto com ação referente a Pessoas tem por objetivo executar uma determinada ação para todas as pessoas que referencia o multiobjeto. Por exemplo:

<“Controlador”>, “Alerta horário medicação”, “Cuidadores”>, o que se quer nesta atividade é que o controlador (sujeito-único) alerte o horário de medicação (ação-atuação) para todos os cuidadores cadastrados no sistema (multiobjeto).

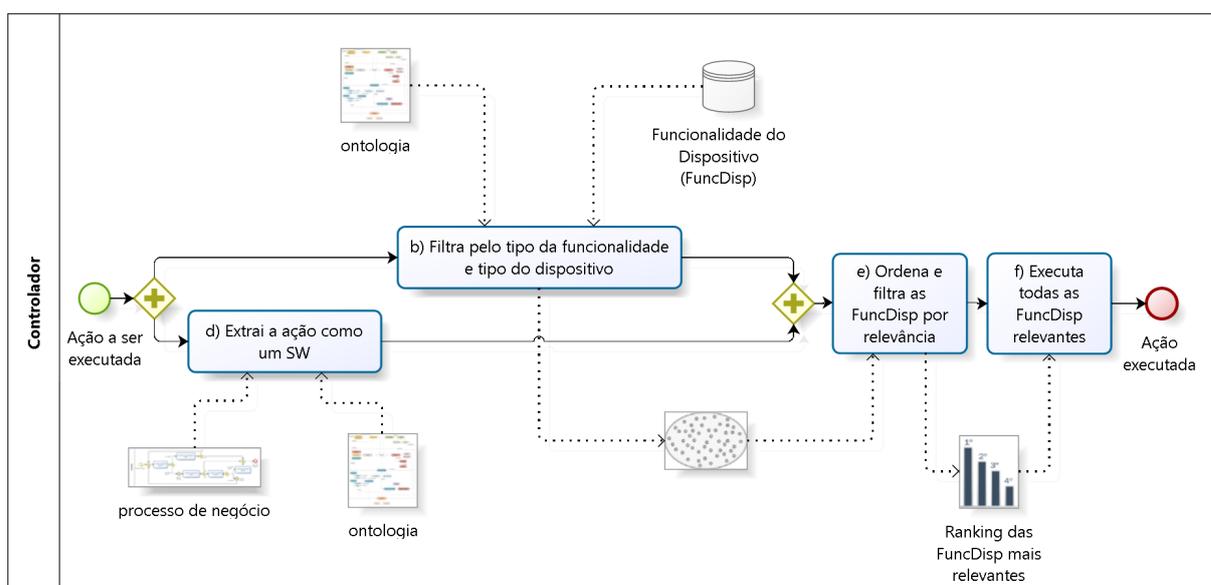
Com isso, o *framework* para multiobjeto executa o *framework* para objeto-único com ação referente a (uma) Pessoa (Seção 3.2.3.1), para cada pessoa pertencente ao tipo informado no multiobjeto.

3.2.3.5 Multiobjeto com ação referente a Dispositivos

O *framework* proposto para multiobjeto com ação referente a Dispositivos (Figura 17) tem por objetivo executar a ação sobre todos os dispositivos pertencentes ao tipo informado no multiobjeto. Por exemplo: <“Controlador”, “Fechar”, “Janelas”>, o que se quer nessa atividade é que o controlador (sujeito) feche (ação-atuação) todas as janelas da casa (multiobjeto).

Desta forma, esse *framework* é similar ao apresentado na Seção 3.2.3.2 (objeto-único com ação referente a (um) Dispositivo), com a diferença que: o passo (a) não é mais executado (pois se quer todos os dispositivos do tipo especificado no objeto, sem olhar a localização de qualquer pessoa); e o passo (f) passa a executar todos os FuncDisp retornados no passo (e).

Figura 17 – Framework proposto para multiobjeto com ação referente a Dispositivos



Fonte: imagem criada pela autora

3.2.3.6 Multiobjeto com ação referente a Organizações

Este *framework* é igual ao apresentado para multiobjeto com ação referente a Dispositivos (Seção 3.2.3.5), mas tem como alvo mais de uma organização.

3.3 Resumo do Capítulo 3

Neste capítulo foi apresentado um modelo para aplicações *Home Care*, onde foi descrito os principais conceitos para este trabalho: pessoa, organização, localização, dispositivo e controlador. Entretanto, esse não é um modelo definitivo, e deve ser visto como um modelo que serve de base para aplicações de *Home Care*, uma vez que tem como objetivo dar suporte na instanciação de atividades de atuação em um PN, o modelo apenas representa os conceitos e relacionamentos importantes para atingir tal objetivo.

Também foi mostrado uma abordagem para composição do rótulo de atividade em modelo de PN com suporte de ontologia, que visa dar mais semântica, diminuir a ambiguidade e tornar o rótulo compreensível por máquina.

Por fim, foi descrito como ocorre a instanciação de atividades de atuação em um PN, para quando o controlador é que executa a ação (sujeito), a ação é de atuação, e quem recebe a ação podendo ser uma única pessoa, dispositivo ou organização (objeto-único) ou um grupo (multiobjeto).

4 CENÁRIOS DE APLICAÇÃO

Para ilustrar a abordagem, são descritas aplicações para o *framework* em quatro cenários fictícios distintos relacionados com *Home Care*:

- Cenário 1: Tratar estado de agitação;
- Cenário 2: Tratar quando se esquece o fogão ligado;
- Cenário 3: Tratar para luzes;
- Cenário 4: Tratar alto nível de monóxido de carbono no ar.

O cenário 1 descreve o comportamento do controlador quando o paciente, que possui a doença de Alzheimer, está no estado de agitação. O cenário 2 descreve o comportamento do controlador quando o paciente, que sofre de demência senil, esquece o fogão ligado. O cenário 3 descreve o comportamento do controlador quando o paciente, que usa cadeira de rodas, entra em um ambiente escuro dentro da casa. E o cenário 4 descreve o comportamento do controlador quando é verificado que o ar dentro da casa possui alto nível de monóxido de carbono.

Esses cenários foram retirados da literatura e escolhidos para exemplificar a utilização do *framework* com triplas de atividades distintas, conforme apresentado no Quadro 4.1.

Quadro 4.1 – Tipos de triplas e cenários para sua exemplificação

<Sujeito, Ação, Objeto>	Cenário utilizado para exemplificação
<sujeito-único, ação-atuação, objeto-único> referente a (uma) Pessoa	Cenário 1
<sujeito-único, ação-atuação, objeto-único> referente a (um) Dispositivo	Cenário 3
<sujeito-único, ação-atuação, objeto-único> referente a (uma) Organização	Cenário 1
<sujeito-único, ação-atuação, multiobjectivo > referente a Pessoas	Cenário 2
<sujeito-único, ação-atuação, multiobjectivo > referente ao Dispositivos	Cenário 4
<sujeito-único, ação-atuação, multiobjectivo > referente a Organizações	—

Fonte: quadro criada pela autora.

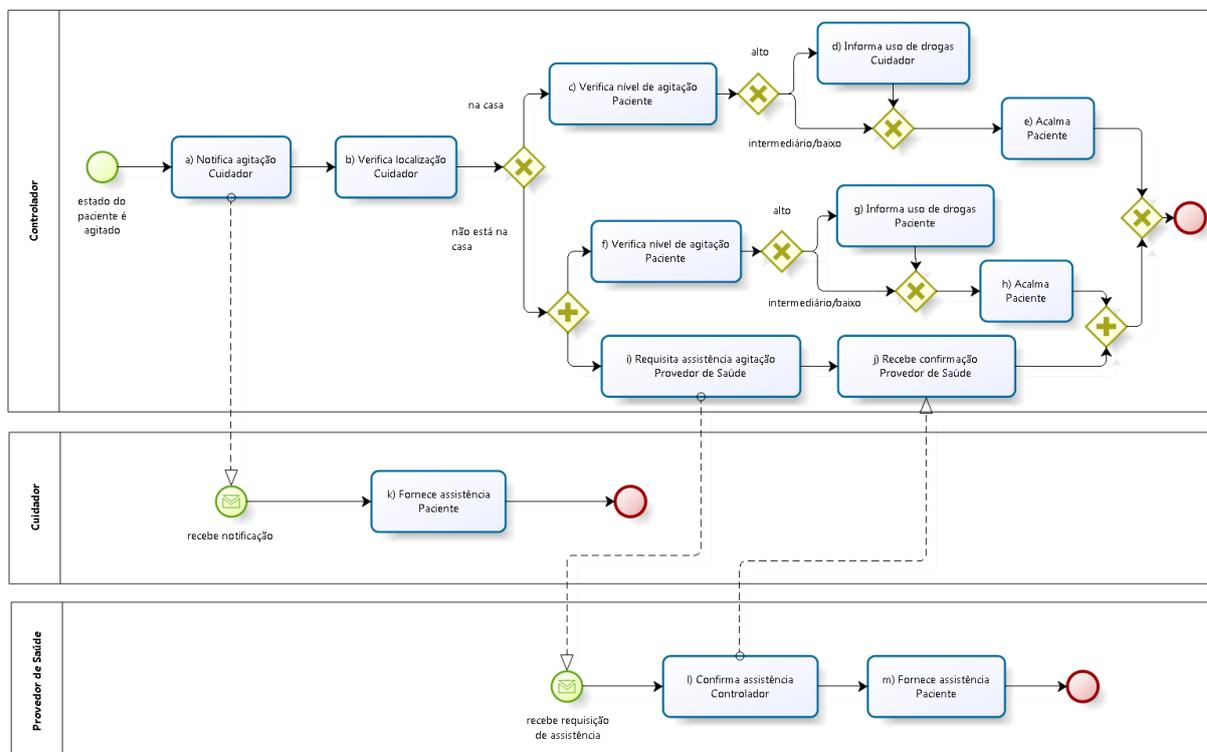
Entretanto, não conseguiu-se encontrar na literatura um cenário para exemplificar a tripla <sujeito-único, ação-atuação, multiobjectivo > referente a Organizações.

A seguir, cada cenário será descrito com maiores detalhes.

4.1 Cenário 1: Tratar estado de agitação

Na Figura 18, o modelo de PN para lidar com o estado de agitação do paciente é mostrado. Esse modelo foi modelado usando a abordagem proposta e foi adaptado de Gassen et al. (2012).

Figura 18 – Modelo do PN para tratar do estado de agitação do paciente (em BPMN 2.0)



Fonte: Imagem criada pela autora com base no trabalho de Gassen et al. (2012).

A Figura 18 descreve o modelo de PN que é acionado quando o controlador detecta o estado de agitação no paciente. Após, o controlador notifica esse estado ao cuidador do paciente (atividade “a”), para que o cuidador forneça alguma assistência (atividade “k”). Ainda, o controlador verifica a localização do cuidador (atividade “b”). Se o cuidador está presente na casa, é verificado o nível de agitação do paciente (atividade “c”). Se o nível está alto, é informado ao cuidador que o paciente precisa usar drogas (atividade “d”) e então o controlador tenta acalmar o paciente (atividade “e”); se o nível está intermediário ou baixo, o controlador apenas tenta acalmar o paciente (atividade “e”). Entretanto, se cuidador não está na casa, o controlador verifica o nível de agitação do paciente (atividade “f”), se está alto o nível, informa o uso de drogas ao paciente (atividade “g”) e tenta acalmar o paciente (atividade “h”), caso o

nível está intermediário ou baixo, apenas tenta acalmar o paciente (atividade “h”). Paralelamente, quando o cuidador não está em casa, o controlador requisita assistência para agitação ao provedor de saúde cadastrado no sistema (atividade “i”) e espera o recebimento de confirmação (atividade “j”).

Esse modelo de PN visa descrever o comportamento do controlador para tentar gerenciar a situação quando o paciente está em estado de agitação com e sem a presença do cuidador na casa.

Embora os PN usualmente sejam modelados com todos os participantes envolvidos no processo, este trabalho só está interessado no controlador. Nesse modelo de PN, as ações das atividades do controlador têm os seguintes tipos:

- Sensoriamento: (b), (c) e (f) e (j) são ações de sensoriamento, uma vez que elas apenas observam o valor do estado de uma entidade, nesse caso, a localização e do cuidador e do paciente e o nível de agitação do paciente, respectivamente;
- Atuação: (a), (d), (e), (g), (h) e (i) são todas ações de atuação porque elas irão mudar o valor do estado de uma entidade.

Esse cenário visa ilustrar a abordagem quando: (i) o objeto da tripla da atividade é único e referente à pessoa; e (ii) o objeto da tripla da atividade é único e referente à organização. As atividades que ilustram (i) são (a), (d), (e), (g) e (h); e para (ii) é a atividade (i).

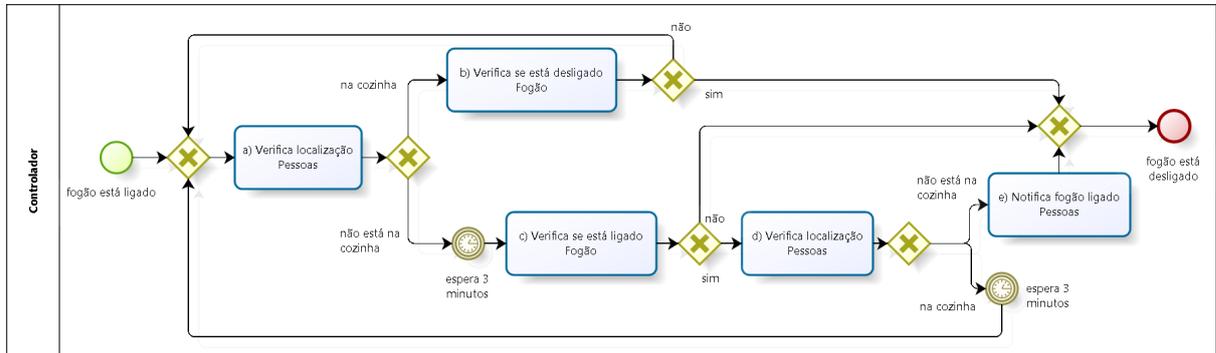
4.2 Cenário 2: Notificação do fogão ligado

O modelo de PN para tratar quando se esquece o fogão ligado, Figura 19, foi retirado do trabalho de Augusto et al. (2006).

O modelo da Figura 19 descreve o comportamento do controlador quando é detectado que o fogão foi ligado (evento inicial). Quando este evento ocorre, o controlador, então, verifica a localização de todas as pessoas cadastradas em seu sistema (atividade “a”). Se há pelo menos uma pessoa na cozinha, é verificado novamente o estado do fogão (atividade “b”), se agora ele está desligado, se encerra o PN, do contrário, o controlador continua monitorando. Se não há ninguém na cozinha, o controlador aguarda 3 minutos, e após verifica novamente o estado do fogão (atividade “c”). Se o fogão está desligado, encerra-se o PN, do contrário, verifica-se novamente a localização de todas as pessoas cadastradas no sistema (atividade “d”). Se não há ninguém na cozinha é então, notificado que o fogão está ligado para todas as pessoas

cadastradas no sistema (atividade “e”). Se há ninguém na cozinha, o controlador aguarda 3 minutos e volta a verificar a localização das pessoas cadastradas no sistema (atividade “a”).

Figura 19 – Modelo do PN para tratar quando se esquece o fogão ligado (em BPMN 2.0)



Fonte: Imagem criada pela autora com base no trabalho de Augusto et al. (2006).

Com isto, esse modelo de PN visa monitorar do acionamento do fogão ao seu desligamento, tentando evitar o fogão fique muito tempo sem alguém no mesmo cômodo.

Explicitando os tipos de ações das atividades do controlador:

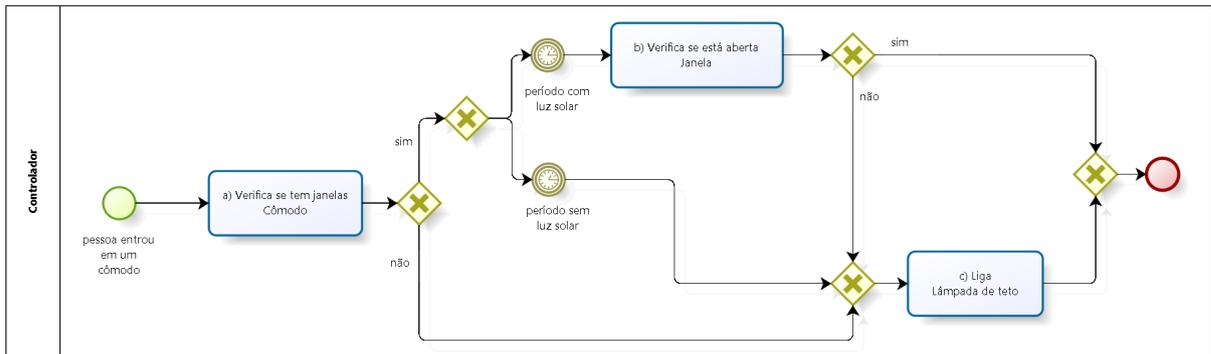
- Sensoriamento: (a), (b), (c) e (d);
- Atuação: (e).

Esse cenário foi escolhido para mostrar a abordagem quando o objeto da tripla da atividade é múltiplo e referente a pessoas. Assim, a ação a ser executada, na atividade (e), será executada várias vezes, de modo a alcançar todas as pessoas envolvidas.

4.3 Cenário 3: Gestão/controle de iluminação

A Figura 20 mostra o modelo do PN para tratar quando uma pessoa entra em um cômodo escuro. Esse cenário foi baseado no cenário proposto por Baladrón et al. (2010), onde se quer automatizar o sistema de iluminação para a pessoa que vive na casa e que é usuária de cadeira de rodas.

Figura 20 – Modelo do PN para tratar quando a pessoa entra em um cômodo (em BPMN 2.0)



Fonte: Imagem criada pela autora com base no trabalho de Baladrón et al. (2010).

Na Figura 20, o modelo de PN tem como seu evento inicial acionado quando uma pessoa entra em um cômodo da casa. Após, é verificado se o cômodo em questão possui janelas (atividade “a”). Se possui janelas, é analisado se o período atual possui luz solar. Se possui luz solar, é verificado se a janela está aberta (atividade “b”), se está aberta, encerra-se o PN, do contrário, é liga-se a lâmpada de teto do cômodo (atividade “c”). Se o período atual não possui luz solar ou se o cômodo não tem janela, liga-se a lâmpada do teto (atividade “c”).

O objetivo desse modelo de PN é evitar que a pessoa ao entrar em um cômodo sem iluminação tenha que procurar o interruptor de luz.

Tipos de ações de atividades do controlador presentes no modelo da Figura 20:

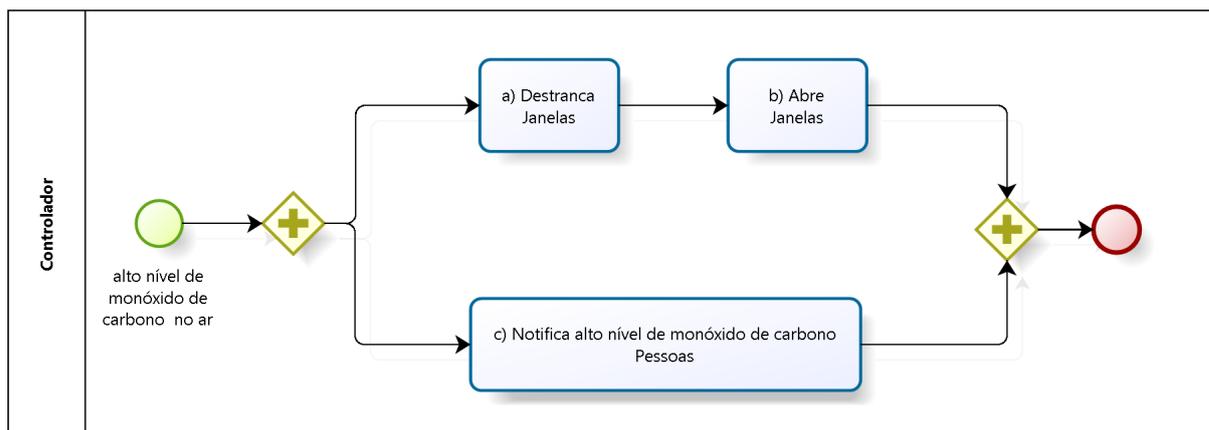
- Sensoriamento: (a) e (b);
- Atuação: (c).

Nesse cenário se quer mostrar a abordagem quando o objeto da tripla da atividade é único e referente a um dispositivo. A atividade que exemplifica este tipo de tripla é a atividade (c), ao qual se quer que um dispositivo (lâmpada) execute uma funcionalidade (ligar).

4.4 Cenário 4: Tratar alto nível de monóxido de carbono no ar

O modelo de PN mostrado na Figura 21 descreve como o controlador deve atuar quando o ar dentro da casa apresenta alto nível de monóxido de carbono. Este cenário foi extraído do trabalho de Wilson et al. (2008).

Figura 21 – Modelo do PN para tratar alto nível de monóxido de carbono no ar (em BPMN 2.0)



Fonte: Imagem criada pela autora com base no trabalho de Wilson et al. (2008)

O modelo de PN apresentado na Figura 21 descreve o comportamento do controlador quando se é detectado alto nível de monóxido de carbono no ar (evento inicial). Quando esse evento ocorre, o controlador destranca (atividade “a”) e abre (atividade “b”) todas as janelas da casa, e paralelamente, notifica a todas as pessoas cadastradas no sistema a presença de alto nível de monóxido de carbono (atividade “c”).

O objetivo desse modelo de PN é, quando detectado esta situação de interesse, arejar o ambiente e notificar as pessoas.

Este modelo de PN só apresenta ações de atividades do controlador do tipo atuação, e foi selecionado para exemplificar quando o objeto da tripla da atividade é múltiplo e referente a diversos dispositivos. As atividades que exemplificam isso são (a) e (b), onde se quer que as funcionalidades (destrancar e abrir) sejam executadas por todas as janelas presentes na casa.

4.5 Resumo do Capítulo 4

Neste capítulo foi apresentado quatro cenários de aplicação retirados da literatura, que visam exemplificar, no próximo capítulo, a abordagem proposta para as diferentes combinações de triplas de rótulo de atividade de PN. Contudo, não conseguiu-se encontrar na literatura um cenário para exemplificar a tripla <sujeito-único, ação-atuação, multiobjectivo > referente a Organizações.

Tais cenários apresentam ações de controlador do tipo sensoriamento e de atuação. Entretanto, apenas as ações de atuação são tratadas neste trabalho.

5 AVALIAÇÃO BASEADA EM ESTUDOS DE CASO

Para verificar se a abordagem proposta consegue selecionar e instanciar o serviço idealizado em cada cenário de aplicação (descrito na Seção 4), realizou-se duas etapas:

- Etapa 1: Criação e configuração do ambiente;
- Etapa 2: Execução dos modelos de processos de negócios.

Onde, na Etapa 1, criou-se e configurou-se um ambiente para a execução dos cenários de aplicação descritos em modelos de PN descritos na Seção 4, e, na Etapa 2, foi realizada a execução e análise desses cenários.

5.1 Etapa 1: Criação e Configuração do Ambiente;

Nesta etapa foram feitas as seguintes atividades:

- Implementação do modelo para aplicações *Home Care*;
- Especificação de um simulador de casa que possui dispositivos que recebem e enviam dados (ver Seção 5.1.2);
- Especificação de um conjunto de descrições semânticas para as funcionalidades dos dispositivos presentes na casa (ver Seção 5.1.3);
- Construção de um controlador para gerenciar o simulador de casa, levando em conta o modelo de Controlador (ver Seção 3.2.1.5) e como é feita a instanciação das atividades de modelo de PN (ver Seção 3.2.3).

Desta forma, o ambiente do experimento é composto por um controlador e um simulador de uma casa inteligente.

O controlador comunica-se com o simulador através de SW, pedindo dados de sensoriamento, e quando encontra uma situação de interesse, executa o modelo de PN que trata tal situação. Para cada atividade de atuação de modelo de PN é executado um conjunto de funcionalidades de dispositivos através de SW, utilizando as descrições semânticas dos dispositivos presentes na casa para verificar quais dispositivos possuem as funcionalidades mais adequadas e realizando o *matching* de SW (ver Seção 2.2).

5.1.1 Implementação do modelo para aplicações *Home Care*

A implementação do modelo proposto neste trabalho (Seção 3.2.1) foi codificada pela autora em OWL-DL utilizando o software *Protégé Desktop 4.3*¹, e as regras de inferência foram escritas em SWRL. A implementação se encontra disponível na página da autora².

Embora o modelo apresentado na Seção 3.2.1 esteja em português, e implementação do modelo foi feita em inglês, a fim de facilitar a disseminação do modelo e das regras associadas.

Para a avaliação dos cenários de aplicação foram criadas as seguintes regras de inferência:

- Regras para quando a ação é referente a (uma) Pessoa;
- Regras para quando a ação é referente a (um) Dispositivo;
- Regras para quando a ação é referente a (uma) Organização.

5.1.1.1 Regras para quando a ação é referente a (uma) Pessoa

Para a avaliação dos cenários de foram criadas quatro regras de inferência que fazem referência a apenas dois tipos de limitação (visual e auditiva), entretanto outras combinações poderiam ser realizadas:

- Regra 1: “Limitação Visual” quando o nível é “Alta” ou “Média”;
- Regra 2: “Limitação Auditiva” quando o nível é “Alta” ou “Média”;
- Regra 3: “Limitação Auditiva” e “Limitação Visual” quando os níveis são “Alta” ou “Média”;
- Regra 4: “Limitação Auditiva” e “Limitação Visual” quando os níveis são “Baixa” ou “Zero”.
- Regra 5: Quando o cuidador não está em casa

5.1.1.1.1 Regra 1: “Limitação Visual” quando o nível é “Alta” ou “Média”

Esta regra verifica se a pessoa alvo da atividade tem apenas a “Limitação Visual”, e essa limitação tem nível do tipo “Alto” ou “Médio”, então seleciona as funcionalidades “Emitir Som” ou “Exibir Vídeo”. A Figura 22 mostra essa regra, em um editor de texto com indentação

¹ Protégé Desktop 4.3, disponível em: <http://protege.stanford.edu/>.

² Implementação do modelo disponível em: <http://inf.ufrgs.br/~jkksilva/house.owl>.

para facilitar a leitura. Entretanto, como as regras são dependentes da nomenclatura utilizada no modelo implementado, essa regra e as demais estão na língua inglesa.

Figura 22 – Regra 1: “Limitação Visual” quando o nível é “Alta” ou “Média”

```

1 Activity(?activity),
2   hasAction(?activity, ?action),      RegardingAPerson(?action),
3   hasObject(?activity, ?person),
4   hasSubject(?activity, ?controller),  Controller(?controller),
5
6 Person(?person),
7   hasLocation(?person, ?location),
8   hasCognitiveDisability(?person, ?dcognitive),
9     Cognitive(?dcognitive),
10    hasDisabilityLevel(?dcognitive, ?lcognitive),
11    (HighDisabilityLevel or MediumDisabilityLevel or LowDisabilityLevel or
12     NoDisabilityLevel)(?lcognitive),
13   hasMotorDisability(?person, ?dmotor),
14     Motor(?dmotor),
15     hasDisabilityLevel(?dmotor, ?lmotor),
16     (HighDisabilityLevel or MediumDisabilityLevel or LowDisabilityLevel or
17      NoDisabilityLevel)(?lmotor),
18   hasHearingDisability(?person, ?dhearing),
19     Hearing(?dhearing),
20     hasDisabilityLevel(?dhearing, ?lhearing),
21     (LowDisabilityLevel or NoDisabilityLevel)(?lhearing),
22   hasVisualDisability(?person, ?dvisual),
23     Visual(?dvisual),
24     hasDisabilityLevel(?dvisual, ?lvisual),
25     (HighDisabilityLevel or MediumDisabilityLevel)(?lvisual),
26
27 AtHome(?location),
28
29 Controllable(?device),
30   hasFunctionality(?device, ?f),
31   (FuncPlayVideo or FuncPlaySound)(?f),
32   hasLocation(?device, ?location)
33
34 -> isBetterExecutedBy(?action, ?f)

```

Fonte: Imagem criada pela autora

Ressalta-se que para a execução desta e das demais regras, teve que criar mais um conceito no modelo ontológico, o “Activity”, e seus relacionamentos: “hasAction”, “hasObject” e “hasSubject”; e mais um relacionamento: o “isBetterExecutedBy”.

Nesta aplicação optou-se por apenas ter uma instância do conceito “Activity”, desta forma, essa instância e seus relacionamentos sempre representam a atividade de atual com ação de atuação. O relacionamento “isBetterExecutedBy” foi adicionado para que assim, possa-se diferenciar quais funcionalidades dos dispositivos são selecionadas pela regra de inferência.

Com isso, para a atividades com rótulo do tipo objeto-único com ação referente a Pessoas (3.2.3.1), essa regra de inferência e as demais implementam os passos (a), (b) e (c) da Figura 15.

5.1.1.1.2 Regra 2: “Limitação Auditiva” quando o nível é “Alta” ou “Média”

Verifica nessa regra se a pessoa alvo da atividade tem apenas a “Limitação Auditiva”, e essa limitação tem nível do tipo “Alto” ou “Médio”, então seleciona as funcionalidades “Exibir Texto” ou “Exibir Imagem” (Figura 23).

Figura 23 – Regra 2: “Limitação Auditiva” quando o nível é “Alta” ou “Média”

```

1 Activity(?activity),
2   hasAction(?activity, ?action),           RegardingAPerson(?action),
3   hasObject(?activity, ?person),
4   hasSubject(?activity, ?controller),     Controller(?controller),
5
6 Person(?person),
7   hasLocation(?person, ?location),
8   hasCognitiveDisability(?person, ?dcognitive),
9     Cognitive(?dcognitive),
10    hasDisabilityLevel(?dcognitive, ?lcognitive),
11    (HighDisabilityLevel or MediumDisabilityLevel or LowDisabilityLevel or
12     NoDisabilityLevel)(?lcognitive),
13  hasMotorDisability(?person, ?dmotor),
14    Motor(?dmotor),
15    hasDisabilityLevel(?dmotor, ?lmotor),
16    (HighDisabilityLevel or MediumDisabilityLevel or LowDisabilityLevel or
17     NoDisabilityLevel)(?lmotor),
18  hasHearingDisability(?person, ?dhearing),
19    Hearing(?dhearing),
20    hasDisabilityLevel(?dhearing, ?lhearing),
21    (HighDisabilityLevel or MediumDisabilityLevel)(?lhearing),
22  hasVisualDisability(?person, ?dvisual),
23    Visual(?dvisual),
24    hasDisabilityLevel(?dvisual, ?lvisual),
25    (LowDisabilityLevel or NoDisabilityLevel)(?lvisual),
26
27 AtHome(?location),
28
29 Controllable(?device),
30   hasFunctionality(?device, ?f),
31   (FuncDisplayText or FuncDisplayImage)(?f),
32   hasLocation(?device, ?location)
33
34 -> isBetterExecutedBy(?action, ?f)

```

Fonte: Imagem criada pela autora

5.1.1.1.3 Regra 3: “Limitação Auditiva” e “Limitação Visual” quando os níveis são “Alta” ou “Média”

Observa-se se a pessoa alvo da atividade tem a “Limitação Auditiva” e a “Limitação Visual” nos níveis do tipo “Alto” ou “Médio”, então seleciona as funcionalidades “Exibir Imagem” ou “Exibir Vídeo” (Figura 24).

Figura 24 – Regra 3: “Limitação Auditiva” e “Limitação Visual” quando os níveis são “Alta” ou “Média”

```

1 Activity(?activity),
2   hasAction(?activity, ?action),           RegardingAPerson(?action),
3   hasObjective(?activity, ?person),
4   hasSubject(?activity, ?controller),      Controller(?controller),
5
6 Person(?person),
7   hasLocation(?person, ?location),
8   hasCognitiveDisability(?person, ?dcognitive),
9     Cognitive(?dcognitive),
10      hasDisabilityLevel(?dcognitive, ?lcognitive),
11      (HighDisabilityLevel or MediumDisabilityLevel or LowDisabilityLevel or
12       NoDisabilityLevel)(?lcognitive),
13   hasMotorDisability(?person, ?dmotor),
14     Motor(?dmotor),
15     hasDisabilityLevel(?dmotor, ?lmotor),
16     (HighDisabilityLevel or MediumDisabilityLevel or LowDisabilityLevel or
17      NoDisabilityLevel)(?lmotor),
18   hasHearingDisability(?person, ?dhearing),
19     Hearing(?dhearing),
20     hasDisabilityLevel(?dhearing, ?lhearing),
21     (HighDisabilityLevel or MediumDisabilityLevel)(?lhearing),
22   hasVisualDisability(?person, ?dvisual),
23     Visual(?dvisual),
24     hasDisabilityLevel(?dvisual, ?lvisual),
25     (HighDisabilityLevel or MediumDisabilityLevel)(?lvisual),
26
27 AtHome(?location),
28
29 Controllable(?device),
30   hasFunctionality(?device, ?f),
31   (FuncPlayVideo or FuncDisplayImage)(?f),
32   hasLocation(?device, ?location)
33
34 -> isBetterExecutedBy(?action, ?f)

```

Fonte: Imagem criada pela autora

5.1.1.1.4 Regra 4: “Limitação Auditiva” e “Limitação Visual” quando os níveis são “Baixa” ou “Zero”

Nesta regra, se a pessoa alvo da atividade não possui “Limitação Auditiva” nem a “Limitação Visual”, seleciona-se as funcionalidades “Exibir Texto”, “Exibir Imagem”, “Emitir Som” e “Exibir Vídeo” (Figura 25).

Figura 25 – Regra 4: “Limitação Auditiva” e “Limitação Visual” quando os níveis são “Baixa” ou “Zero”

```

1 Activity(?activity),
2   hasAction(?activity, ?action),           RegardingAPerson(?action),
3   hasObject(?activity, ?person),
4   hasSubject(?activity, ?controller),      Controller(?controller),
5
6 Person(?person),
7   hasLocation(?person, ?location),
8   hasVisualDisability(?person, ?dvisual),
9     Visual(?dvisual),
10      hasDisabilityLevel(?dvisual, ?lvisual),
11      (LowDisabilityLevel or NoDisabilityLevel) (?lvisual),
12   hasHearingDisability(?person, ?dhearing),
13     Hearing(?dhearing),
14     hasDisabilityLevel(?dhearing, ?lhearing),
15     (LowDisabilityLevel or NoDisabilityLevel) (?lhearing),
16   hasCognitiveDisability(?person, ?dcognitive),
17     Cognitive(?dcognitive),
18     hasDisabilityLevel(?dcognitive, ?lcognitive),
19     (LowDisabilityLevel or NoDisabilityLevel) (?lcognitive),
20   hasMotorDisability(?person, ?dmotor),
21     Motor(?dmotor),
22     hasDisabilityLevel(?dmotor, ?lmotor),
23     (LowDisabilityLevel or NoDisabilityLevel) (?lmotor),
24
25 AtHome(?location),
26
27 Controllable(?device),
28   hasFunctionality(?device, ?f),
29   (FuncDisplayImage or FuncDisplayText or FuncPlayVideo or FuncPlaySound) (?f),
30   hasLocation(?device, ?location)
31
32 -> isBetterExecutedBy(?action, ?f)

```

Fonte: Imagem criada pela autora

5.1.1.1.5 Regra 5: Quando o cuidador não está na casa

Esta regra visa selecionar a funcionalidade do tipo “Interação com Pessoa” do dispositivo que foi cadastrado para emergência através do relacionamento *temComunicaçãoParaEmergencia*.

Figura 26 – Regra 5: Quando o cuidador não está na casa

```

1 Activity(?activity),
2   hasAction(?activity, ?action),           RegardingAPerson(?action),
3   hasObject(?activity, ?person),
4   hasSubject(?activity, ?controller),      Controller(?controller),
5
6 Location(?location),
7   NotAtHome(?location),
8
9 Person(?person),
10  Caregiver(?person),
11  hasLocation(?person, ?location),
12  hasCommunicationForEmergency(?person, ?device),
13
14 Controllable(?device),
15  hasFunctionality(?device, ?f),
16  ..... Functionality(?f),
17  ..... InteractionWithPerson(?f)
18
19 -> isBetterExecutedBy(?action, ?f)

```

Fonte: Imagem criada pela autora

5.1.1.2 Regras para quando a ação é referente a Dispositivo

Como nos cenários aparecem apenas 3 atividades com ação referente a Dispositivo (no cenário 3, atividade (c), e no cenário 4, atividade (a) e (b)), só são modeladas regras para demonstrar esses casos.

- Regra 6: “Ação Ligar” para Luz do Teto
- Regra 7: “Ação Destrancar” para Grupo de Janela Inteligente
- Regra 8: “Ação Abrir” para Grupo de Janela Inteligente

5.1.1.2.1 Regra 6: “Ação Ligar” para Luz do Teto

Nesta regra, para uma a atividade com objeto-único de ação referente a Dispositivo (representado no bloco de linhas 1-7 da Figura 27), tal que esta ação é do tipo “Ligar” (linha 22 da Figura 27) e o dispositivo é do tipo “Luz do Teto” (linha 13 da Figura 27), seleciona-se as funcionalidade do tipo “Ligar” (linha 21 da Figura 27) dos dispositivos do tipo “Luz do Teto” (linha 14 da Figura 27) com que estejam na mesma localização do paciente (bloco de linhas 9-11 e bloco de linhas 16-17 da Figura 27). Para a instância da “Ação Abrir” é então criado os relacionamentos *ehMelhorExecutadoPor* (linha 24 da Figura 27) com essas funcionalidades.

Figura 27 – Regra 6: “Ação Ligar” para Luz do Teto

```

1 Activity(?activity),
2   hasAction(?activity, ?action),
3     RegardingADevice(?action),
4   hasObject(?activity, ?device_1),
5     Controllable(?device_1),
6   hasSubject(?activity, ?controller),
7     Controller(?controller),
8
9 Patient(?pessoa),
10  hasLocation(?pessoa, ?location),
11    Location(?location),
12
13 Roof_light(?device_1),
14 Roof_light(?device_2),
15
16 Controllable(?device_2),
17   hasLocation(?device_2, ?location),
18   hasFunctionality(?device_2, ?f),
19     Functionality(?f),
20
21 FuncTurnOn(?f),
22 ActionTurnOn(?action)
23
24 -> isBetterExecutedBy(?action, ?f)

```

Fonte: Imagem criada pela autora

5.1.1.2.2 Regra 7: “Ação Destrancar” para Grupo de Janela Inteligente

Para uma a atividade com multiobjeto de ação referente a Dispositivo (representado no bloco de linhas 1-7 da Figura 28), tal que esta ação é do tipo “Destrancar” (linha 19 da Figura 28) e o multiobjeto é do tipo “Controláveis” (linha 5 da Figura 28), que são um grupo de dispositivos do tipo “Janela Inteligente” (linha 9-11 da Figura 28), seleciona-se as funcionalidade do tipo “Destrancar” (linha 18 da Figura 28), dos dispositivos do tipo “Janela Inteligente” (linha 12 da Figura 28). Para a instância da “Ação Destrancar” é então criado os relacionamentos *ehMelhorExecutadoPor* (linha 21 da Figura 28) com essas funcionalidades.

Figura 28 – Regra 7: “Ação Destrancar” para Grupo de Janela Inteligente

```

1 Activity(?activity),
2   hasAction(?activity, ?action),
3     RegardingADevice(?action),
4   hasObject(?activity, ?devices),
5     Controllables(?devices),
6   hasSubject(?activity, ?controller),
7     Controller(?controller),
8
9 isGroupOf(?devices, ?device_1),
10
11 Smart_window(?device_1),
12 Smart_window(?device_2),
13
14 Controllable(?device_2),
15   hasFunctionality(?device_2, ?f),
16     Functionality(?f),
17
18 FuncUnlock(?f),
19 ActionUnlock(?action)
20
21 -> isBetterExecutedBy(?action, ?f)

```

Fonte: Imagem criada pela autora

5.1.1.2.3 Regra 8: “Ação Abrir” para Grupo de Janela Inteligente

A mesma regra da seção anterior (Seção 5.1.1.2.2), só que muda o tipo da ação, de “Ação Destrancar” para “Ação Abrir” (linha 12 da Figura 29), e o tipo da funcionalidade, de “Funcionalidade Destrancar” para “Funcionalidade Abrir”.

Figura 29 – Regra 8: “Ação Abrir” para Grupo de Janela Inteligente

```

1 Activity(?activity),
2   hasAction(?activity, ?action),
3     RegardingADevice(?action),
4   hasObject(?activity, ?devices),
5     Controllables(?devices),
6   hasSubject(?activity, ?controller),
7     Controller(?controller),
8
9 isGroupOf(?devices, ?device_1),
10
11 Smart_window(?device_1),
12 Smart_window(?device_2),
13
14 Controllable(?device_2),
15   hasFunctionality(?device_2, ?f),
16     Functionality(?f),
17
18 FuncOpen(?f),
19 ActionOpen(?action)
20
21 -> isBetterExecutedBy(?action, ?f)

```

Fonte: Imagem criada pela autora

5.1.1.3 Regras para quando a ação é referente a Organização

Como nos cenários de uso só aparece uma atividade referente a Organização (cenário 1, atividade (i)), foi criado apenas uma regra para este caso.

Para uma a atividade com objeto-único de ação referente a Organização (representado no bloco de linhas 1-7 da Figura 30), tal que esta ação é do tipo “Referente a Organização” (linha 3 da Figura 30) e a organização é do tipo “Provedor de Saúde” (linha 9 da Figura 30), seleciona-se as funcionalidade do tipo “Interação com Organização” (linha 15 da Figura 30), dos dispositivos do tipo “Controlável” (linha 12 da Figura 30), cuja interação vise atingir a organização Provedor de Saúde (linhas 10 e 16 da Figura 30). Para a instância da “Ação Referente a Organização” é então criado os relacionamentos *ehMelhorExecutadoPor* (linha 21 da Figura 30) com essas funcionalidades.

Figura 30 – Regra 9: Interação com o Provedor de Saúde

```

1 Activity(?activity),
2   hasAction(?activity, ?action),
3     RegardingAnOrganization(?action),
4   hasObject(?activity, ?organization_1),
5     Organization(?organization_1),
6   hasSubject(?activity, ?controller),
7     Controller(?controller),
8
9 HealthcareProvider(?organization_1),
10 HealthcareProvider(?organization_2),
11
12 Controllable(?device),
13   hasFunctionality(?device, ?f),
14     Functionality(?f),
15     InteractionWithOrganization(?f),
16       hasInteractionWith(?f,?organization_2)
17
18 -> isBetterExecutedBy(?action, ?f)

```

Fonte: Imagem criada pela autora

5.1.2 Especificação de um Simulador de Casa

A especificação de simulador de casa englobou: os dispositivos e suas funcionalidades, com base no trabalho de Park et al. (2003), que a casa terá; os possíveis tipos de pessoas com suas limitações e estados; organização e o comportamento da unidade de deslocamento; e os objetos não são controláveis. A partir dessa especificação, Pötter (2013) desenvolveu o

*HouseSimulator*³ como uma aplicação Web para simular uma casa com dispositivos que recebem e enviam dados através de SW. Essa aplicação foi realizada dentro do projeto Ambientes Sociais Inteligentes Sensíveis ao Contexto⁴, ao qual este trabalho também faz parte, com o intuito de oferecer um ambiente onde a abordagem proposta neste trabalho possa ser analisada.

No *HouseSimulator* é possível:

- Configurar o simulador, através das seguintes atividades:
 - Adicionar, editar e remover locais dentro da casa;
 - Adicionar, editar e remover pessoas;
 - Adicionar, editar e remover organizações;
 - Adicionar, editar e remover objetos não controláveis;
 - Vincular, a cada local da casa criado, um conjunto de dispositivos já cadastrados;
 - Alterar o estado de um dispositivo (por exemplo, configurar para que o estado Ligado = Falso);
 - Especificar o tipo de uma pessoa (cuidador, médico, paciente, visitante), suas limitações (visual, motora, auditiva, cognitiva) e seus níveis (alto, médio, baixo);
 - Vincular uma pessoa a um local da casa;
 - Alterar o estado de uma pessoa ou dispositivo (por exemplo, configurar para que o estado Desidratação = Alta);
- Inicializar a simulação;
- Interagir durante a simulação;
 - Fazer requisições aos SW do simulador para obter dados de sensoriamento da casa;
 - Fazer requisições aos SW dos dispositivos para executar uma ação de atuação;
 - Visualizar os estados dos dispositivos;
 - Visualizar os estados do paciente;
 - Visualizar o *log* da simulação.
- Finalizar a simulação;
- Listar os dispositivos cadastrados;
 - Visualizar a lista de dispositivos cadastrados e seus estados (total de 32 dispositivos);

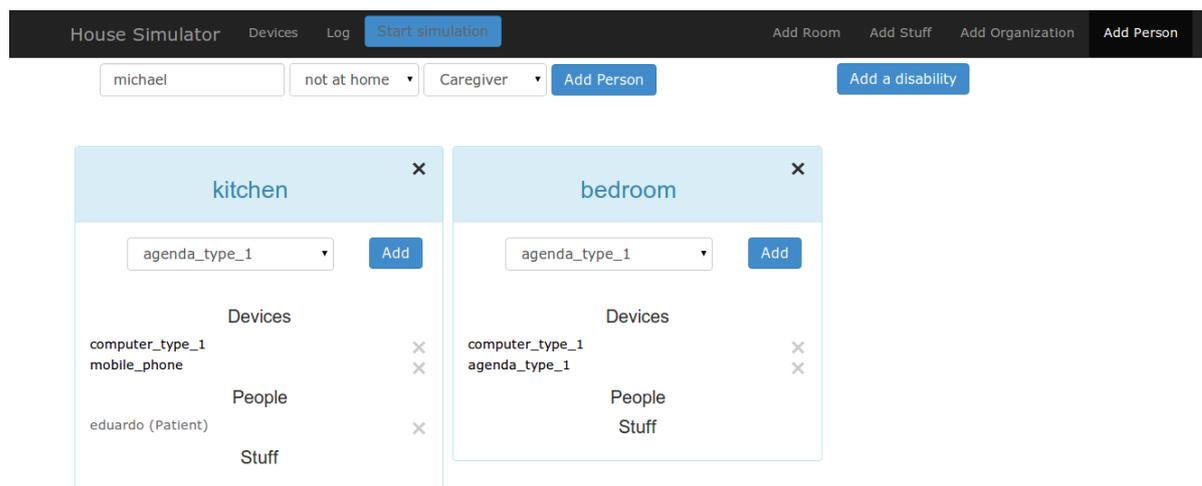
³ <http://housesimulator.herokuapp.com/>.

⁴ Projeto em desenvolvimento no Instituto de Informática da UFRGS, aprovado no Edital Universal 2012 do CNPq. O projeto é coordenado pelo prof. José Palazzo Moreira de Oliveira.

- Visualizar, para cada dispositivo, suas funcionalidades (total de 139 funcionalidades);
- Visualizar, para cada dispositivo, o WSDL do SW que o representa;
- Visualizar o histórico de *logs* de todas as simulações realizadas.

A Figura 31 exibe a tela de configuração inicial do *HouseSimulator*, onde foram criados dois locais, *kitchen* e *bedroom*. Neles foram vinculados dispositivos: em *kitchen* vinculou-se os dispositivos *computer_type_1* e *mobile_phone*, e em *bedroom* vinculou-se os dispositivos *computer_type_1* e *agenda_type_1*.

Figura 31 – Tela de configuração inicial do *HouseSimulator*



Fonte: Pötter (2013, p. 38)

Uma vez que é realizada a configuração do simulador, pode-se inicializar o *HouseSimulator*. Quando o *HouseSimulator* está executando, ele passa a receber e enviar dados através de requisições de SW.

A Figura 32 mostra como o simulador exibe o feedback quando o dispositivo *computer_type_1* atende a uma requisição sobre a sua funcionalidade para mostrar um texto. O simulador destaca em verde os estados que se alteram com a execução dessa funcionalidade (*busy* e *showing_text*), e mostra essa execução da funcionalidade do dispositivo no *log*.

Figura 32 – Tela de *feedback* do *HouseSimulator*

The screenshot displays two panels. The left panel, titled 'kitchen', lists 'Devices' including 'computer_type_1' and 'mobile_phone', each with status attributes like 'active', 'busy', 'on', 'playing_sound', 'playing_video', 'showing_image', and 'showing_text'. Below devices, it lists 'People' with 'eduardo (Patient)' and various health conditions such as 'agitation', 'asthma_attack', 'dehydration', 'diabetes_complication', 'glucose_level', 'shock', 'spinal_injury', 'stroke', and 'temperature'. The right panel, titled 'Organizations', lists 'hospital' and includes a log entry: '[09/12/2013 03:30] ID:93 NAME:computer_type_1 LOCATION:kitchen ACTION:show_text_computer_type_1 success'.

Fonte: *Screenshot* obtido pela autora com base no simulador.

5.1.3 Especificação do Conjunto de Descrições Semânticas para as Funcionalidades dos Dispositivos

Com base na especificação do conjunto de descrições semânticas para as funcionalidades dos dispositivos previstos na casa, foram feitas manualmente pelo bolsista de iniciação científica Paulo Renato Lanzarin, 139 descrições semânticas das funcionalidades. Ao todo foram criados 139 arquivos OWL-S com as descrições semânticas, um para cada funcionalidade de dispositivo (no Apêndice B é mostrado um exemplo de descrição semântica).

Essas descrições foram criadas em *OWL-S 1.2*⁵, possibilitando a explicitação dos conceitos de entrada e saída, e as pré-condições e efeitos para a execução de uma determinada funcionalidade. As pré-condições e efeitos foram descritas em SWRL.

Primeiramente, tentou-se utilizar o *OWL-S Editor*⁶, que é um *plugin* para o *Protégé Desktop 3.5*, para gerar as descrições semânticas OWL-S automaticamente. Entretanto, observou-se que a versão do OWL-S que o *OWL-S Editor* gera é incompatível com a versão que o *OWL-S API*, API para ler as descrições semânticas em OWL-S, compreende.

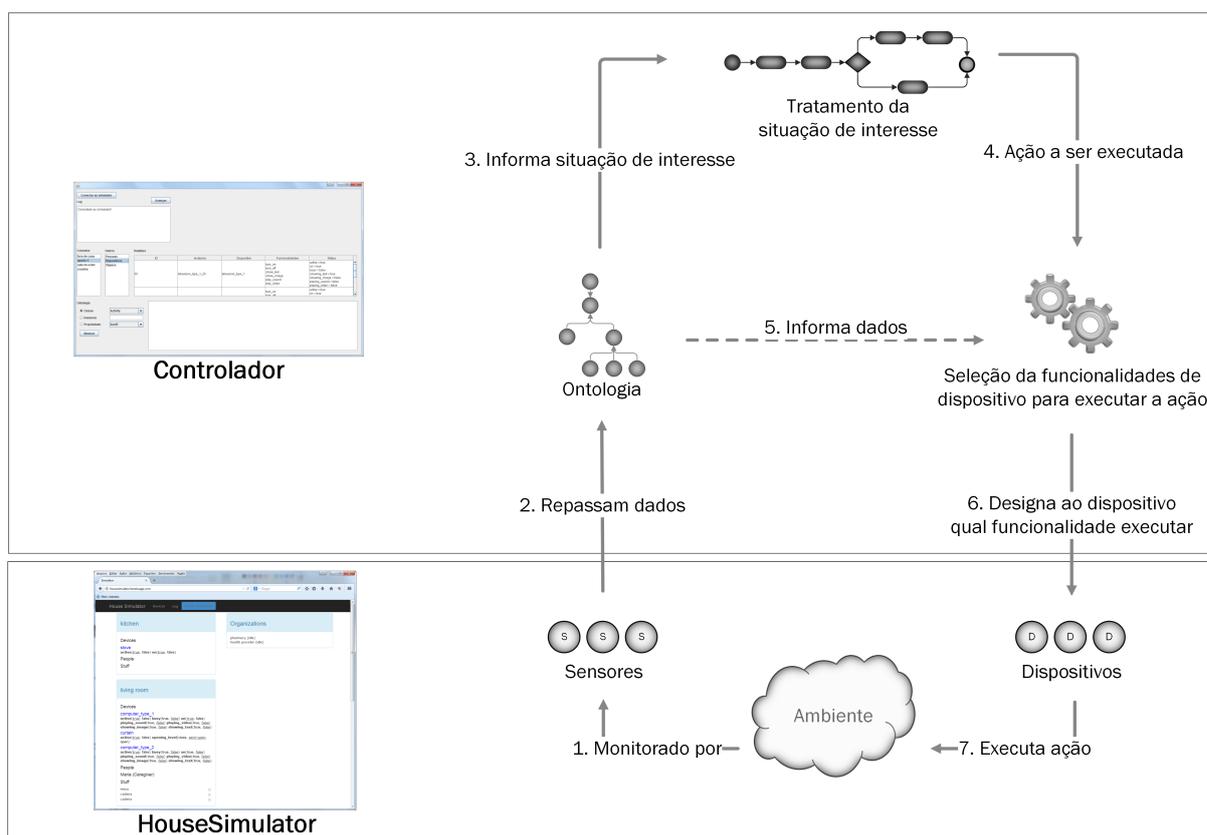
⁵ OWL-S 1.2: <http://www.ai.sri.com/daml/services/owl-s/1.2/>

⁶ OWL-S Editor: <http://owlseditor.semwebcentral.org/index.shtml>

5.1.4 Construção de um Controlador

A implementação do controlador foi desenvolvida pela autora, em Java, para verificar se a abordagem proposta consegue selecionar e instanciar um serviço a uma dada ação de atuação. Foram utilizadas as seguintes API: *OWL-S API 3.0*⁷ para manipular serviços descritos em OWL-S (integrada a essa API já está o *JENA API*⁸ que manipula ontologias descritas em OWL) e *SOA MODEL JAVA API*⁹ para manipular serviços descritos em WSDL. E o motor de inferência utilizado foi o Pellet (que está integrado ao *JENA API*).

Figura 33 – Visão geral da interação entre o controlador e o *HouseSimulator*



Fonte: Imagem elaborada pela autora

Toda a comunicação entre o controlador e o *HouseSimulator* é feita através de SW. A Figura 33 mostra uma visão geral da interação entre eles: o ambiente (simulado pelo *HouseSimulator*) é monitorado por sensores; o controlador recebe dados do ambiente por esses sensores; e a partir desses dados, o controlador atualiza a ontologia instanciada (Seção 5.1.1),

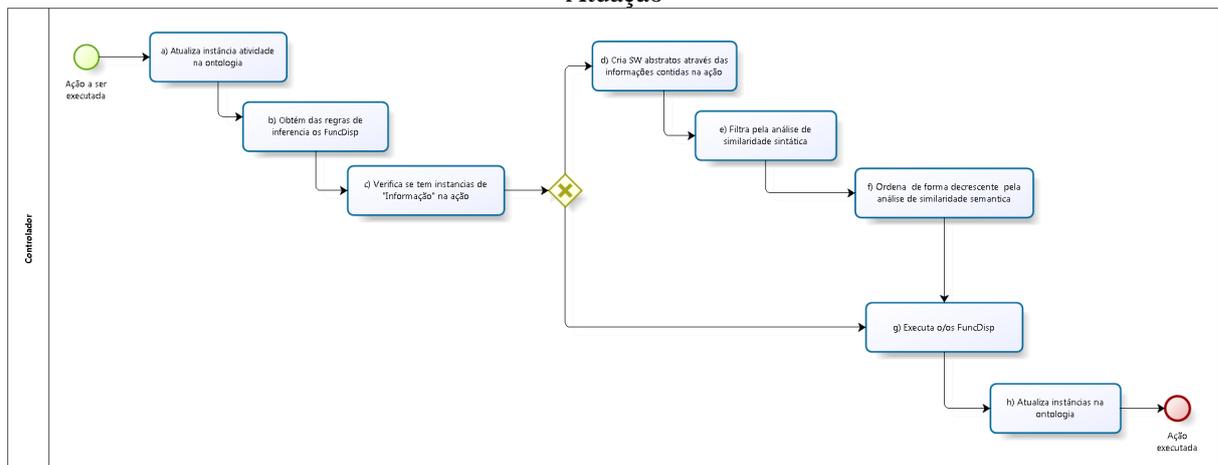
⁷ OWL-S API: <http://on.cs.unibas.ch/owl-s-api/>

⁸ JENA API: <https://jena.apache.org/>

⁹ SOA MODEL JAVA API: <http://www.membrane-soa.org/soa-model/>.

através da ontologia instanciada e dos situações cadastradas, o controlador detecta a situação de interesse; a situação de interesse é tratada pelo seu processo de negócio correspondente; para cada ação de atuação presente na atividade do processo de negócio, o controlador seleciona uma funcionalidade de dispositivo para executá-la; o controlador então faz uma requisição dessa funcionalidade de dispositivo ao simulador; e por fim, o simulador executa a ação.

Figura 34 – Implementação do framework proposto para instanciação de atividade com ação do tipo Atuação



Fonte: imagem criada pela autora

A implementação de como é feita a instanciação das atividades do PN (descrita na Seção 3.2.3), é da seguinte forma (Figura 34):

- a) Atualiza instância atividade na ontologia: A ontologia instanciada possui apenas uma única instância do tipo “Atividade”, assim, esta atividade é sempre a atividade atual, e por isso deve ser atualizada quando for executar a ação da atividade.
- b) Obtém das regras de inferência as FuncDisp: Uma vez atualizada a instância atividade, as regras de inferência (apresentadas na Seção 5.1.1) rodam automaticamente, e adicionam o relacionamento *isBetterExecutedBy* a ação com as FuncDisp que podem executá-la de acordo com as regras de inferência.
- c) Verifica se tem instâncias de “Informação” na ação: Se a instanciação da ação possui instâncias do tipo “Informação”, executa o passo (d), senão executa o passo (g).
- d) Cria SW abstratos através das informações contidas na ação: Para cada instância de informação é criado um serviço Web abstrato.
- e) Filtra pela análise de similaridade sintática: Cada serviço Web abstrato é combinado com cada FuncDisp obtida do passo (b), formando pares. Para cada par é então

analisado a similaridade sintática utilizando o algoritmo de Gao et al. (2002), (Seção 2.2.2), filtrando aqueles que não possuem similaridade sintática.

- f) Ordena de forma decrescente pela análise de similaridade semântica: Para cada par retornado do passo (e), é realizada a análise de similaridade semântica utilizando o algoritmo Lin (1998) com IC computado como em Zhou et al. (2008) (Seção 2.1), ordenando de forma decrescente com base neste valor de similaridade semântica, e aplicando um limiar de 0,7 para que um par seja considerado similar semanticamente.
- g) Executa o/os FuncDisp: De acordo com o tipo de atividade é, então, verificado as condições do primeiro FuncDisp do ordenamento ou todos os FuncDisp, para então se forem satisfeitas executá-lo.
- h) Atualiza instâncias na ontologia: Como estas ações do tipo “Atuação”, logo visam modificam o estado de uma ou mais entidades, as instâncias presentes na ontologia devem ser atualizadas para refletir este novo estado.

A Figura 35 exibe as áreas do controlador implementados, onde: a área 1, mostra o PN atual e suas atividades; a área 2, mostra as informações adquiridas do simulador da casa, quanto a pessoa, local e dispositivo; e a área 3, mostra as informações contida na ontologia instanciada, podendo, por exemplo, verificar as propriedades de uma determinada instancia da ontologia.

Figura 35 – Áreas do controlador

The screenshot displays a control interface with three distinct sections, labeled 1, 2, and 3 on the right side.

Area 1: Contains a 'Conectar ao simulador' button, a 'Log' section with an 'Avançar' button, and a text box showing 'Conectado ao simulador!'.

Area 2: Features a table with columns: ID, Instancia, Dispositivo, Funcionalidades, and Status. The table lists three instances: curtain_91, door_98, and stove_62.

ID	Instancia	Dispositivo	Funcionalidades	Status
91	curtain_91	curtain	open close open_halfway	active = true opening_level = close
98	door_98	door	open close lock unlock	active = true open = false locked = true
62	stove_62	stove	tturn_on turn_off	active = true on = false

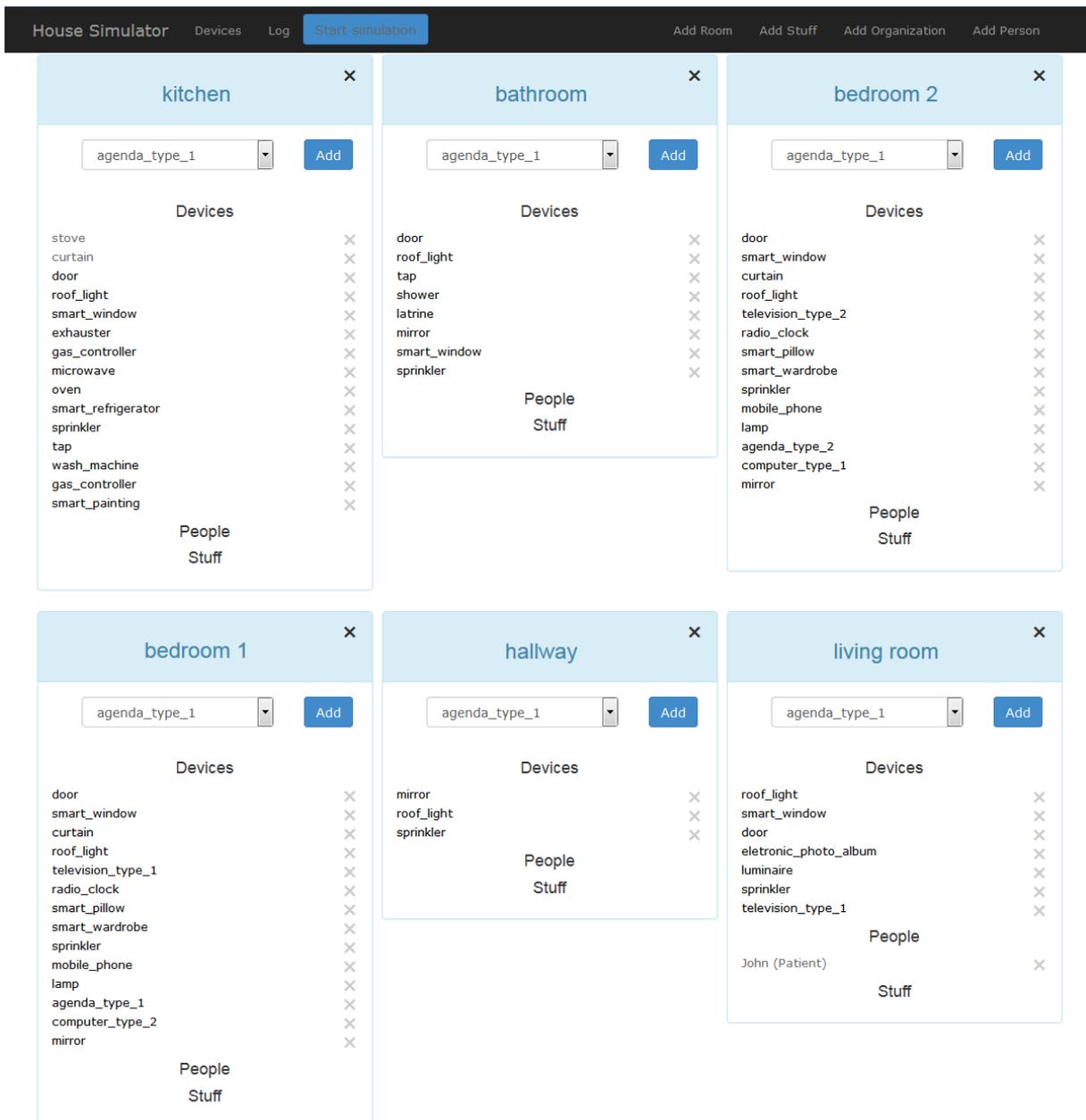
Area 3: Shows ontology details for 'person_patient_John_10'. It includes a 'Mostrar' button and a list of properties such as 'type' and 'sameAs' with their corresponding URIs.

Fonte: imagem criada pela autora

5.2 Etapa 2: Execução dos Processos de Negócios

Para executar os cenários descritos na Seção 4, primeiramente configurou-se o simulador com: 6 cômodos dentro da casa; os 60 dispositivos distribuídos pelos cômodos; e duas pessoas dentro da casa (John, que é o paciente com limitação visual média, e Eve, a cuidadora, que não possui limitações) (Figura 36).

Figura 36 – Configuração do simulador para a execução dos cenários



Fonte: Screenshot obtido pela autora com base no simulador

5.2.1 Execução do Cenário 1

No cenário 1 (tratar estado de agitação), o paciente John encontra-se no estado agitado (Figura 37) e a cuidadora Eve não está em casa.

Figura 37 – Paciente John no estado agitado

People

John (Patient)

agitation(*high*, intermediate, low, none) **asthma_attack**(high, intermediate, low, *none*) **dehydration**(high, intermediate, low, *none*) **diabetes_complication**(high, intermediate, low, *none*) **glucose_level**(high, *medium*, low) **shock**(high, intermediate, low, *none*) **spinal_injury**(true, *false*) **stroke**(high, intermediate, low, *none*) **temperature**(high, *medium*, low)

Fonte: *Screenshot* obtido pela autora com base no simulador

O controlador, que está monitorando o ambiente, detecta esta situação de interesse, e aciona o PN para tratar o estado de agitação do paciente (Figura 38).

Figura 38 – Controlador detecta estado de agitação no paciente

The screenshot shows a control interface with a log window, a flowchart, and a table of device details.

Log: Conectado ao simulador!
EVENTO -> Detectado estado de agitacao no paciente.

Flowchart: A flowchart starting with 'Estado do paciente é agitado'. It branches into 'na casa' and 'nao está na casa'. The 'na casa' path leads to 'a) Notifica agitação Cuidador', then 'b) Verifica localização Cuidador', and then a decision diamond. If 'intermediária/altas', it goes to 'c) Informa uso de drogas Cuidador', then 'd) Agiliza Paciente'. If 'baixa', it goes to 'e) Verifica nível de agitação Paciente', then 'f) Informa uso de drogas Paciente', then 'h) Agiliza Paciente'. The 'nao está na casa' path leads to 'e) Verifica nível de agitação Paciente', then 'g) Informa nível de agitação Paciente', then 'i) Requer assistência agitação Provedor de Saúde', and finally 'j) Notifica confirmação Provedor de Saúde'.

Table:

ID	Instancia	Dispositivo	Funcionalidades	Status
91	curtain_91	curtain	open close open_halfway	active = true opening_level = close
98	door_98	door	open close lock unlock	active = true open = false locked = true
62	stove_62	stove	turn_on turn_off	active = true on = false

Ontologia: Classe: Person; Instancia: ; Propriedade: hasLocation.

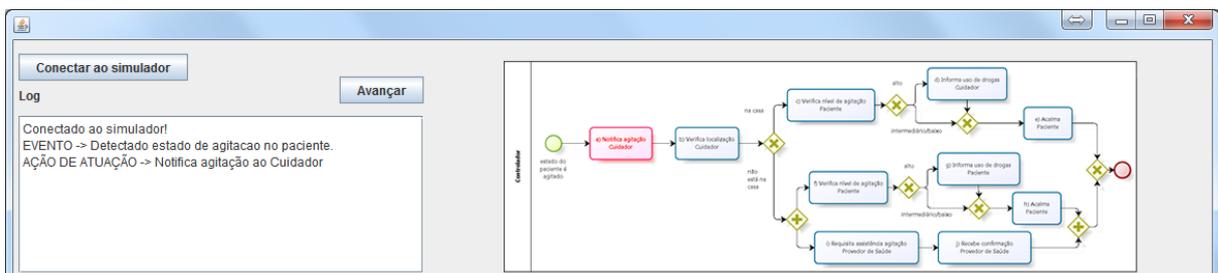
Fonte: *Screenshot* obtido pela autora com base no controlador

Após, a primeira atividade do PN acionada é a atividade (a): <<“Notificar Agitação”, “Cuidador”>> (Figura 39), onde o primeiro elemento é a ação, e o segundo, o objeto (o sujeito

da tripla da atividade será omitido por simplificação, uma vez que, sempre se trata do controlador). Essa atividade é do tipo “Atuação” com objeto-único referente a (uma) Pessoa, e como exemplificar a instanciação desse tipo de atividade e do tipo objeto-único referente a (uma) Organização são os objetivos desse cenário, suas instanciações serão detalhadas.

Antes, é importante salientar que a instância de “Ação Notificar Agitação” possui os quatro tipos de “Informação” (“Texto”, “Imagem”, “Áudio” e “Vídeo”) (Figura 40), onde a instância informação do tipo “Texto” tem conteúdo do tipo *string* (Figura 41), e as demais informações são do tipo *base64Binary* (Figura 42, Figura 43 e Figura 44). A partir dessas instâncias de “Informação” serão criados os serviços Web abstratos.

Figura 39 – Cenário 1, atividade (a) a ser executada



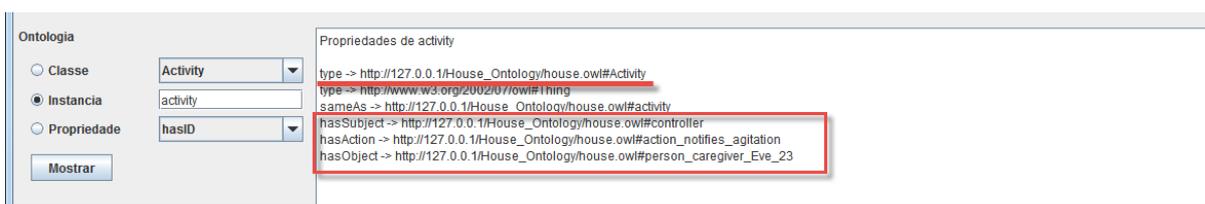
Fonte: Screenshot obtido pela autora com base no controlador

Figura 40 – Cenário 1, instâncias de “Informações” contidas na instância de “Ação Notificar Agitação”

Fonte: Screenshot obtido pela autora com base no controlador

Para executar a instância da “Ação Atuação”, o controlador atualiza a instância atividade (Figura 45) na ontologia. Desta forma, as regras de inferência rodam, e um conjunto de funcionalidades do dispositivo (FuncDisp) que são candidatas a executar a ação é obtido através do relacionamento *isBetterExecutedBy* que liga a ação com os FuncDisp (Figura 46). Neste caso, como a cuidadora Eve não está na casa, o conjunto de FuncDisp obtido foi meio da regra 5 apresentada na Seção 5.1.1.1.5, ao qual retorna as funcionalidades de um dispositivo cadastrado para emergências.

Figura 45 – Cenário 1, instancia de atividade atualizada para atividade (a)



Fonte: Screenshots obtido pela autora com base no controlador

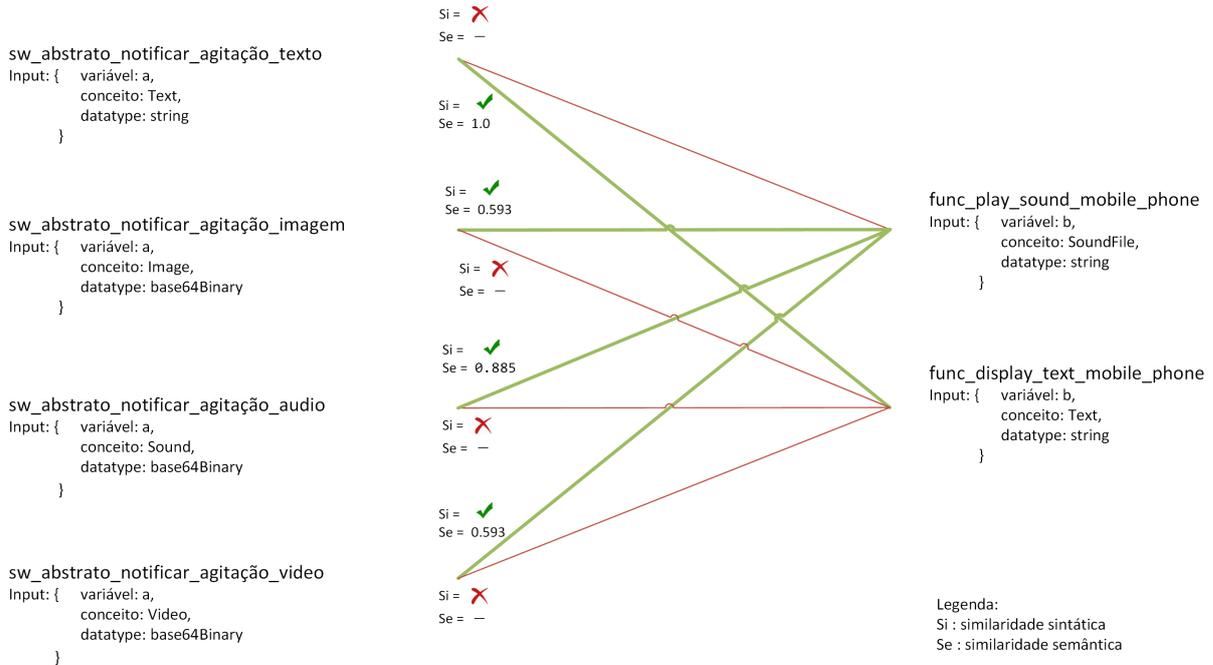
Figura 46 – Cenário 1, resultado das regras de inferência para atividade (g)



Fonte: Screenshots obtido pela autora com base no controlador

Com as 2 FuncDisp obtidas pelas regras de inferência e os 4 serviços Web abstratos criados a partir das instancias informação, formam-se 8 pares de combinações ao qual é verificado a similaridade sintática e semântica. Com a análise de similaridade sintática, consegue-se filtrar alguns pares, resultando em 4 pares similares sintaticamente (Figura 47). A partir desses pares, é feita a análise de similaridade semântica, e o valor de similaridade semântica é utilizado para ordenar os pares de forma decrescente, e após é aplicado o limiar 0,7, onde os pares com valor abaixo desse limiar não são considerados similares semanticamente.

Figura 47 – Cenário 1, resultado das análises de similaridade sintática e semântica para “Ação Notificar Agitação”



Fonte: imagem criada pela autora

Tabela 5.1 – Cenário 1, ordenação dos pares para a atividade (a) através da similaridade semântica

<i>SW abstrato</i>	<i>FuncDisp</i>	<i>Se</i>
sw_abstrato_notificar_agitação_texto	func_display_text_mobile_phone	1,000
sw_abstrato_notificar_agitação_audio	func_play_sound_mobile_phone	0,885
sw_abstrato_notificar_agitação_imagem	func_play_sound_mobile_phone	0,593
sw_abstrato_notificar_agitação_video	func_play_sound_mobile_phone	0,593

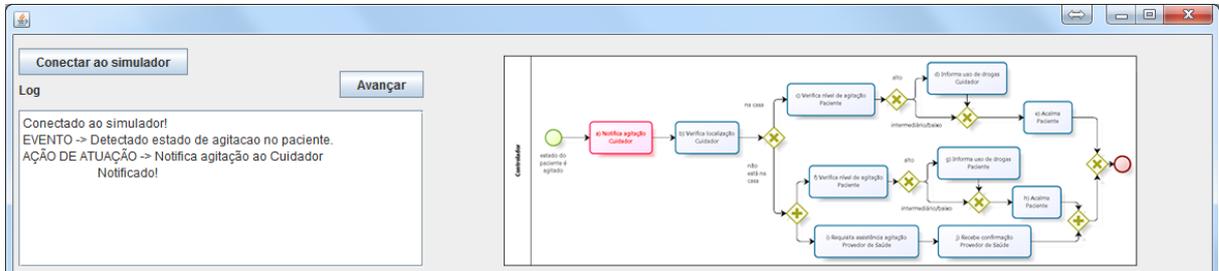
Fonte: tabela criada pela autora

A Tabela 5.1 mostra o par <SW abstrato, FuncDisp> e seu valor de similaridade semântica. Os resultados apresentados na tabela estão ordenados pelo valor de similaridade semântico, onde a área cinza mostra os pares que não alcançaram o limiar (0,7). Pares que não atingiram o limiar semântico, não poderiam ser utilizados, pois não são semanticamente similares, ou seja, são serviços que não desempenham a mesma função semântica. Logo, quanto maior o valor de similaridade semântica (mais próximo de 1), mais similar semanticamente um serviço é do seu par. Desta forma, o controlador que possui previamente o SW abstrato, que não existe no mundo real, quer encontrar uma FuncDisp (representada por SW), que existe no mundo real, que seja o mais similar a ela, para que possa assim, instanciá-la.

O valor 0,7 para o limiar semântico foi obtido através da observação do valor de limiar nos experimentos que demonstrava o melhor resultado.

Uma vez ordenado os pares e aplicado o limiar, o primeiro FuncDisp da ordenação (*func_play_sound_mobile_phone*) é acionado. A Figura 48 mostra o controlador após executar a atividade, e a Figura 49 mostra a tela do simulador, onde o log exibe o acionamento do SW.

Figura 48 – Cenário 1, atividade (a) executada pelo controlador



Fonte: *Screenshots* obtido pela autora com base no controlador

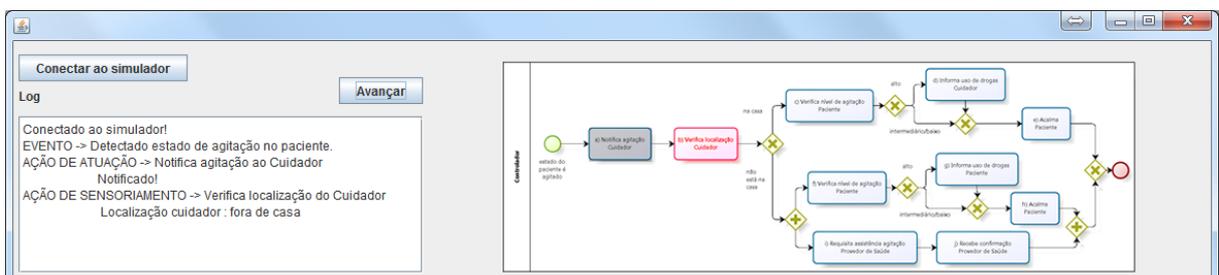
Figura 49 – Cenário 1, atividade (a) executada no simulador



Fonte: *Screenshots* obtido pela autora com base no simulador

Seguindo, a próxima atividade deste PN, a atividade (b): <“Verifica localização”, “Cuidador”>, é executada (Figura 50), ao qual se verifica a localização do cuidador, que ainda permanece fora da casa. Por se tratar de uma ação do tipo “Sensoriamento”, e esse não é o foco deste trabalho, as atividades deste tipo não serão detalhadas.

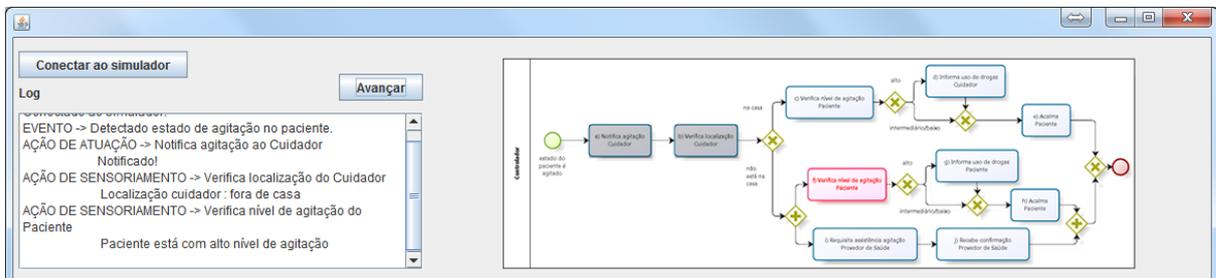
Figura 50 – Cenário 1, atividade (b) executada



Fonte: *Screenshots* obtido pela autora com base no controlador

A próxima atividade, a atividade (f): <“Verifica nível de agitação”, “Paciente”>, também é relativa a “Ação Sensoriamento”, onde se observa o nível da agitação do paciente (Figura 51), como ele está com o nível alto, a atividade (g) deve ser executada.

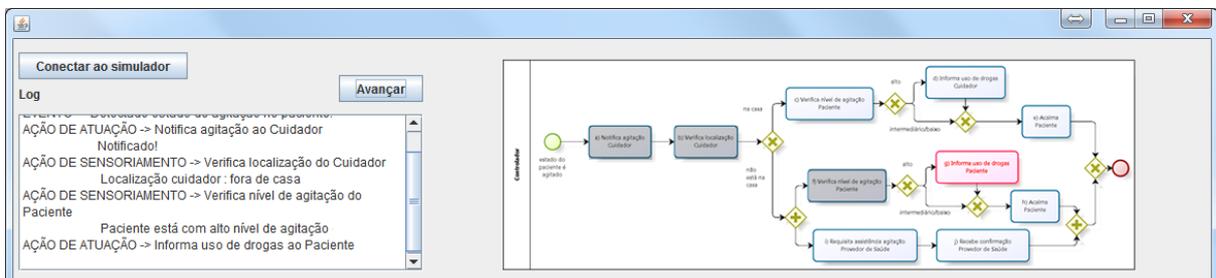
Figura 51 – Cenário 1, atividade (f) executada



Fonte: Screenshots obtido pela autora com base no controlador

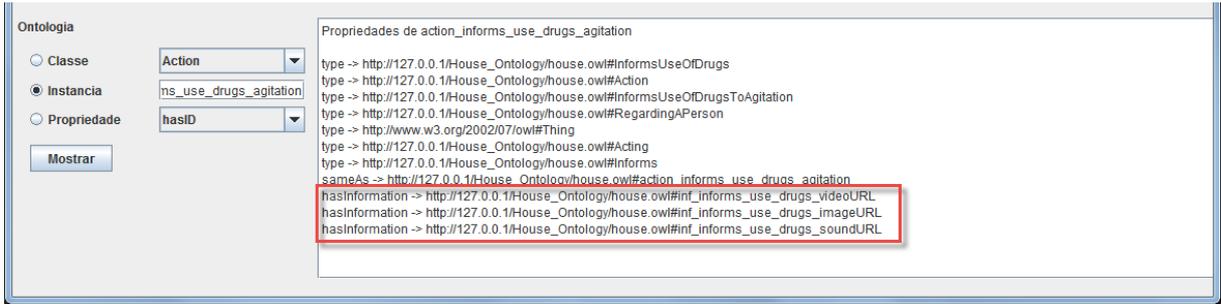
A atividade (g): <“Informar uso de drogas”, “Paciente”>, do tipo “Ação Atuação” é referente a (uma) Pessoa (Figura 52). Observando as informações contidas nessa ação, verifica-se que ela possui 3 instâncias desse conceito (URL do vídeo, URL da imagem e URL do som) (Figura 53). Aqui, ao invés de ter o arquivo em binário, o que se tem é um endereço para o arquivo. Desta forma, o conteúdo dessas informações é uma *string*.

Figura 52 – Cenário 1, atividade (g) a ser executada



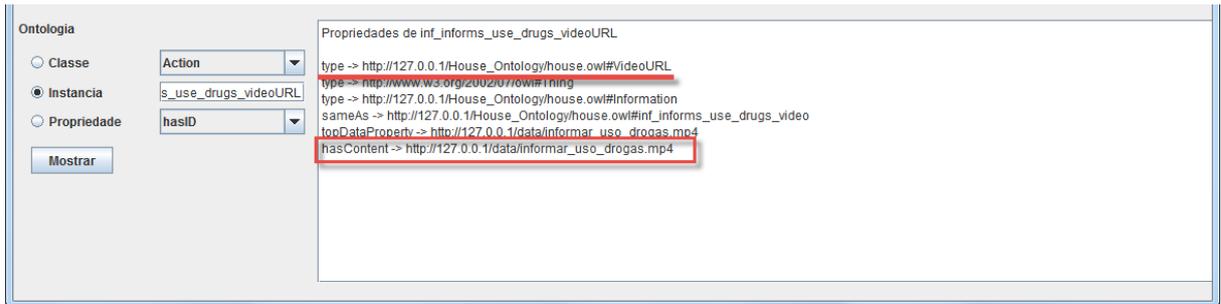
Fonte: Screenshots obtido pela autora com base no controlador

Figura 53 – Cenário 1, instâncias de “Informações” contidas na “Ação Informar uso de drogas”



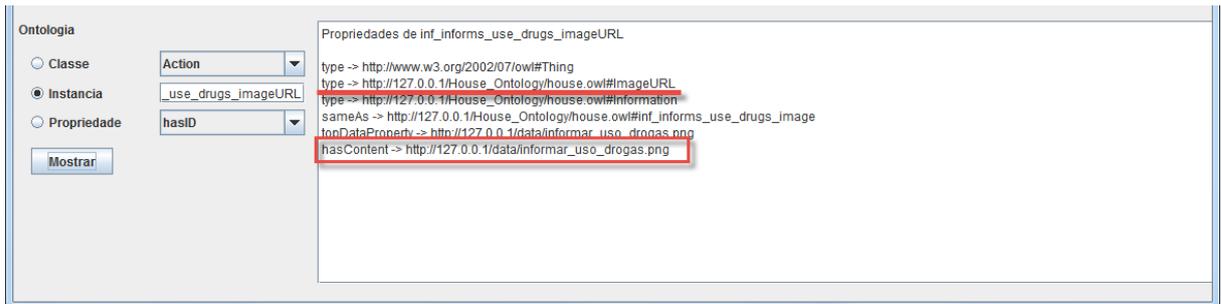
Fonte: Screenshots obtido pela autora com base no controlador

Figura 54 – Cenário 1, instância informação do tipo “URL do Vídeo” com conteúdo do tipo string



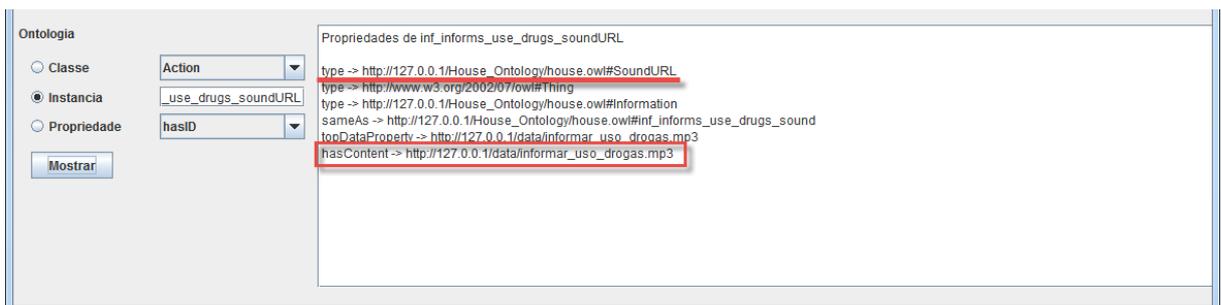
Fonte: Screenshots obtido pela autora com base no controlador

Figura 55 – Cenário 1, instância informação do tipo “URL da Imagem” com conteúdo do tipo string



Fonte: Screenshots obtido pela autora com base no controlador

Figura 56 – Cenário 1, instância informação do tipo “URL do Som” com conteúdo do tipo string



Fonte: Screenshots obtido pela autora com base no controlador

Lembrando que onde o paciente se encontra, na sala de estar, o cômodo possui os seguintes dispositivos que poderiam executar a uma ação referente a (uma) Pessoa (ou seja, funcionalidades do tipo “Interação com Pessoa”), de acordo com a configuração do simulador (Figura 36) e Apêndice A:

Tabela 5.2 – FuncDip localizados no mesmo cômodo do paciente e que possuem funcionalidades do tipo “Interação com Pessoa”

<i>Dispositivo</i>	<i>Funcionalidade</i>
smart_window	show_text(Text String text)
	show_image(ImageURL String imageURL)
eletronic_photo_album	show_text(Text String text)
	show_image(ImageURL String imageURL)
	play_sound(SoundURL String soundURL) (*)
television_type_2	show_text(Text String text)
	show_image(Image base64Binary image)
	play_sound(Sound base64Binary sound) (*)
	play_video(Video base64Binary video) (*)

Fonte: tabela criada pela autora

Desta forma, o controlador roda as regras de inferência (Seção 5.1.1.1), que levam em consideração as funcionalidades dos dispositivos, a localização da pessoa alvo e suas limitações, e obtém um conjunto de FuncDisp que podem executar a ação desejada, mas resta verificar se tais funcionalidades, representadas por serviço Web, podem receber como argumento a informação contida da ação. Na Figura 57 mostra na área em destaque, o resultado das regras de inferência para a ação da atividade (g), onde três FunDisp foram selecionadas (indicadas por (*) na Tabela 5.2), indicados através do relacionamento “*isBetterExecutedBy*”.

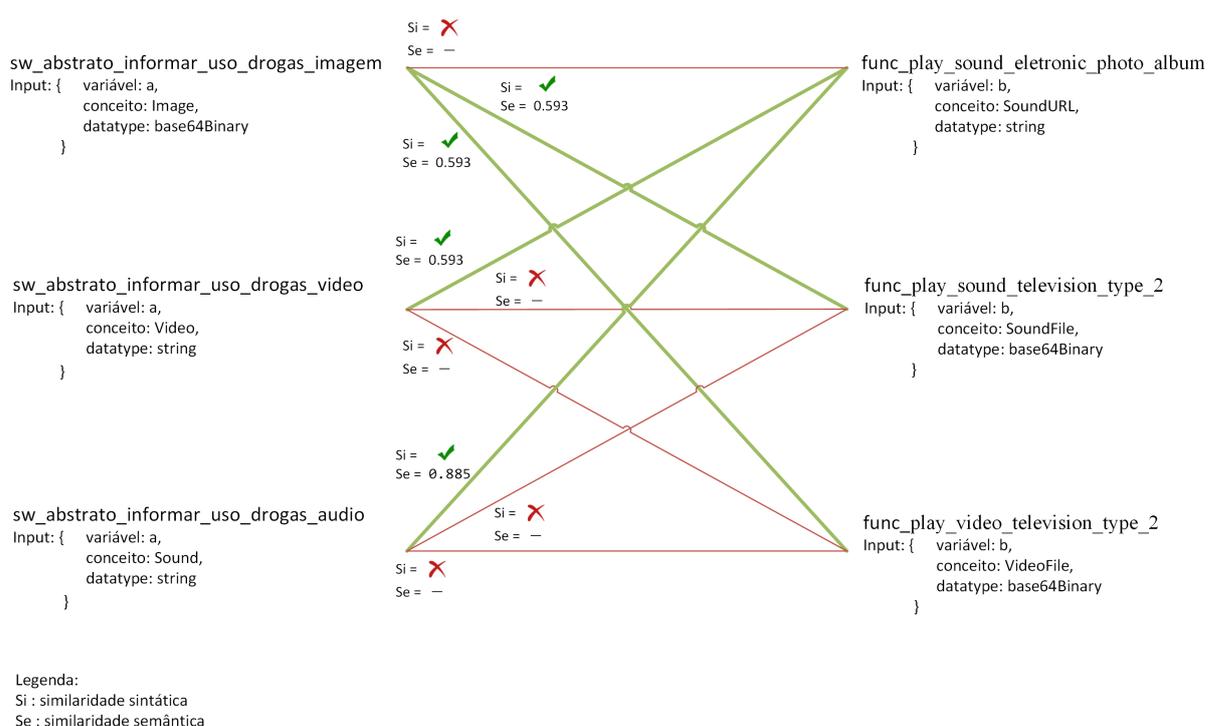
Figura 57 – Cenário 1, resultado das regras de inferência para atividade (g)

The screenshot shows an ontology editor interface. On the left, there are controls for 'Ontologia' with radio buttons for 'Classe', 'Instancia', and 'Propriedade', and a 'Mostrar' button. The main area displays the 'Propriedades de action_informs_use_drugs_agitation'. The 'isBetterExecutedBy' property is highlighted with a red box, showing three instances: 'http://127.0.0.1/House_Ontology/house.owl#func_play_video_television_type_2_157', 'http://127.0.0.1/House_Ontology/house.owl#func_play_sound_eletronic_photo_album_118', and 'http://127.0.0.1/House_Ontology/house.owl#func_play_sound_television_type_2_157'.

Fonte: Screenshots obtido pela autora com base no controlador

A partir do resultado das regras de inferência e dos serviços Web abstratos, criados a partir das instâncias de “Informação” contidas na instância de “Ação Informar Uso de Drogas”, é realizada a análise de similaridade sintática. Se a análise de similaridade sintática do FuncDisp é indicar que o FuncDisp é similar ao serviço Web abstrato, então executa-se a análise de similaridade semântica. A Figura 58 mostra o resultado das análises de similaridade sintática e semântica para cada par serviço Web abstrato e FuncDisp, onde as arestas em destaque (cor verde e de maior espessura) são aquelas cuja similaridade sintática é indicar que são similares.

Figura 58 – Cenário 1, resultado das análises de similaridade sintática e semântica para “Ação Informar Uso de Drogas”



Fonte: imagem criada pela autora

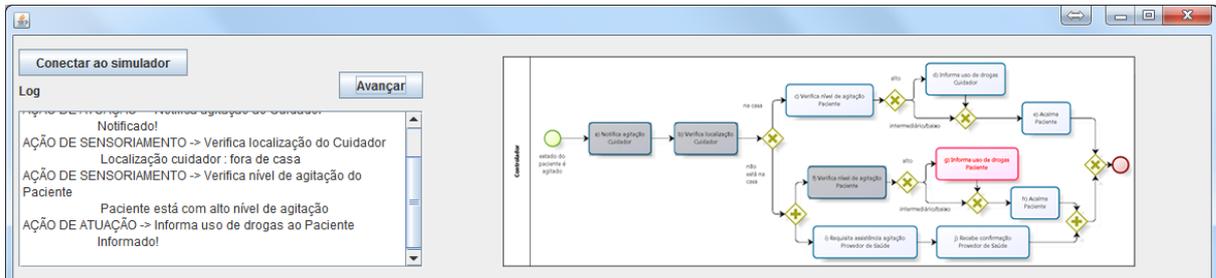
A partir dos pares que possuem similaridade sintática, é feita uma ordenação decrescente pelo valor de similaridade semântica e aplicado o limiar (Tabela 5.3). Como o limiar é 0,7, apenas o primeiro par poderia executar esta ação. O primeiro FuncDisp (*func_play_sound_eletronic_photo_album*) da lista dessa ordenação é então verificado se suas precondições estão satisfeitas (ter os estados ativo = *true*, ligado = *true* e ocupado = *false*). Como elas estão verdadeiras, é acionado para executar a ação. Essa execução da ação, pode ser visualizado no controlador (Figura 59) e no simulador (Figura 60), onde a FuncDisp acionada é realçada na cor verde e registrada no campo de *log*.

Tabela 5.3 – Cenário 1, ordenação dos pares para a atividade (g) através da similaridade semântica

<i>SW abstrato</i>	<i>FuncDisp</i>	<i>Se</i>
sw_abstrato_informar_uso_drogas_audio	func_play_sound_eletronic_photo_album	0.885
sw_abstrato_informar_uso_drogas_imagem	func_play_sound_television_type_2	0.593
sw_abstrato_informar_uso_drogas_imagem	func_play_video_television_type_2	0.593
sw_abstrato_informar_uso_drogas_video	func_play_sound_eletronic_photo_album	0.593

Fonte: tabela criada pela autora

Figura 59 – Cenário 1, atividade (g) executada pelo controlador



Fonte: Screenshots obtido pela autora com base no controlador

Figura 60 – Cenário 1, atividade (g) executada no simulador

House Simulator
Devices
Log
Finish simulation

living room

Devices

[roof_light](#)
active(true, false) light_type(intense, dimmed, pulsating) on(true, false)

[smart_window](#)
active(true, false) busy(true, false) on(true, false)
showing_image(true, false) showing_text(true, false)
locked(true, false) open(true, false)

[door](#)
active(true, false) locked(true, false) open(true, false)

[eletronic_photo_album](#)
active(true, false) busy(true, false) on(true, false)
playing_sound(true, false) showing_image(true, false)
showing_text(true, false)

[luminaire](#)
active(true, false) light_type(intense, dimmed, pulsating) on(true, false)

[sprinkler](#)
active(true, false) busy(true, false) on(true, false)

[television_type_2](#)
active(true, false) busy(true, false) on(true, false)
playing_sound(true, false) playing_video(true, false)
showing_image(true, false) showing_text(true, false)

People

John (Patient)
agitation(high, intermediate, low, none) asthma_attack(high, intermediate, low, none) dehydration(high, intermediate, low, none) diabetes_complication(high, intermediate, low, none) glucose_level(high, medium, low) shock(high, intermediate, low, none) spinal_injury(true, false) stroke(high, intermediate, low, none) temperature(high, medium, low)

Stuff

Organizations

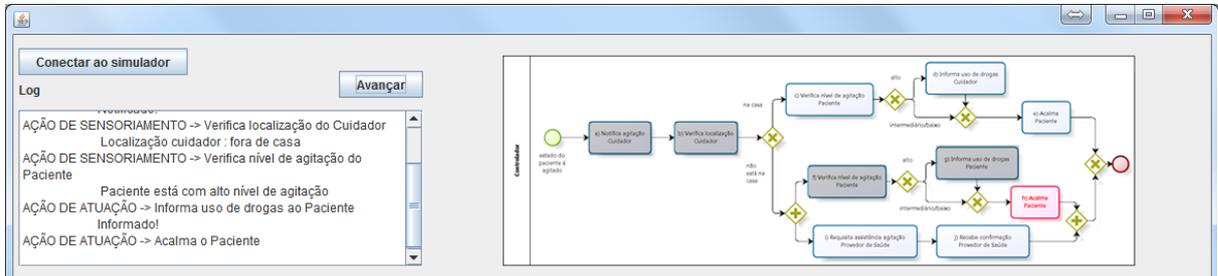
pharmacy (idle)
health provider (idle)

[23/01/2015 16:32] Caregiver Eve was notified that: O paciente está em estado de agitação
[23/01/2015 16:39] ID:118 NAME:eletronic_photo_album LOCATION:living room ACTION:play_sound_eletronic_photo_album success

Fonte: Screenshots obtido pela autora com base no simulador

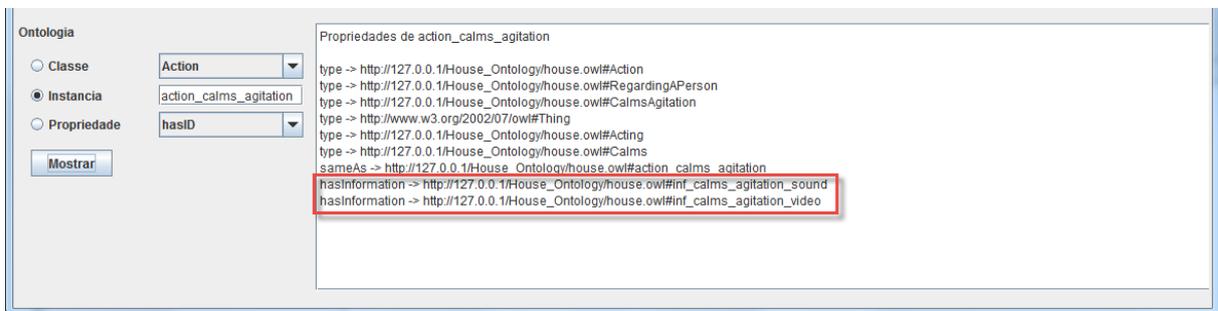
Continuando, a atividade (e): <“Ação Acalmar”, “Paciente”>, também tem a ação do tipo “Atuação Referente a (uma) Pessoa” (Figura 61), e possui 2 instâncias de “Informação” (Figura 62), uma do tipo “Áudio” () e outra do tipo “Vídeo” (), ambas com conteúdo do tipo *base64Binary*.

Figura 61 – Cenário 1, atividade (h) a ser executada



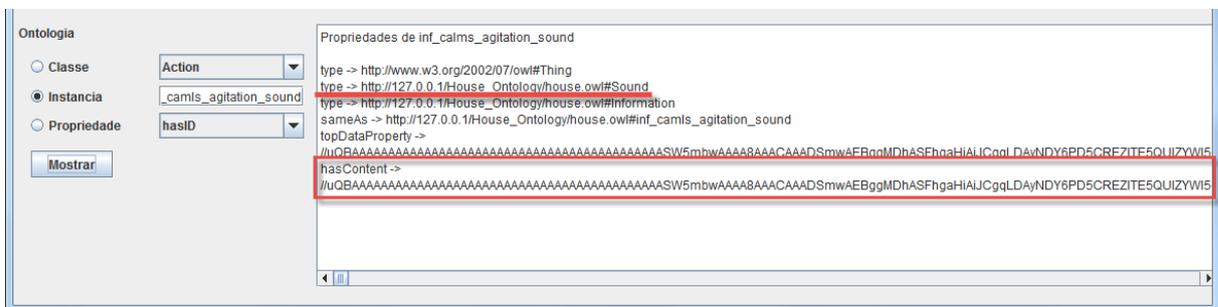
Fonte: Screenshots obtido pela autora com base no controlador

Figura 62 – Cenário 1, instâncias de “Informações” contidas na instância de “Ação Acalmar Agitação”



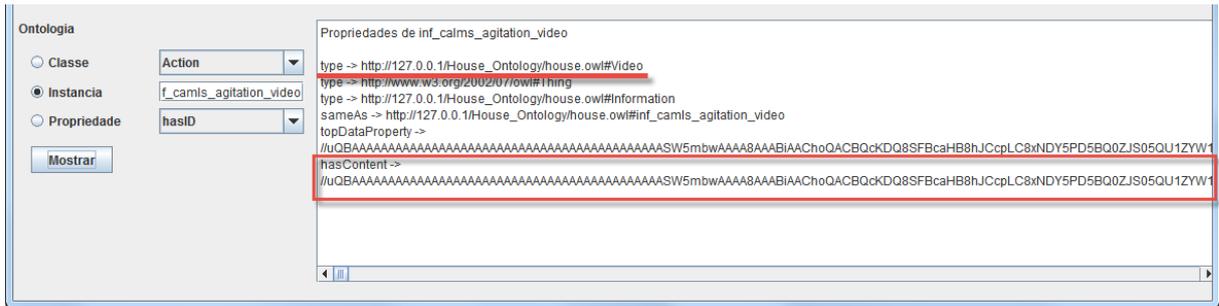
Fonte: Screenshots obtido pela autora com base no controlador

Figura 63 – Cenário 1, instância informação do tipo “Áudio” com conteúdo do tipo *base64Binary*



Fonte: Screenshots obtido pela autora com base no controlador

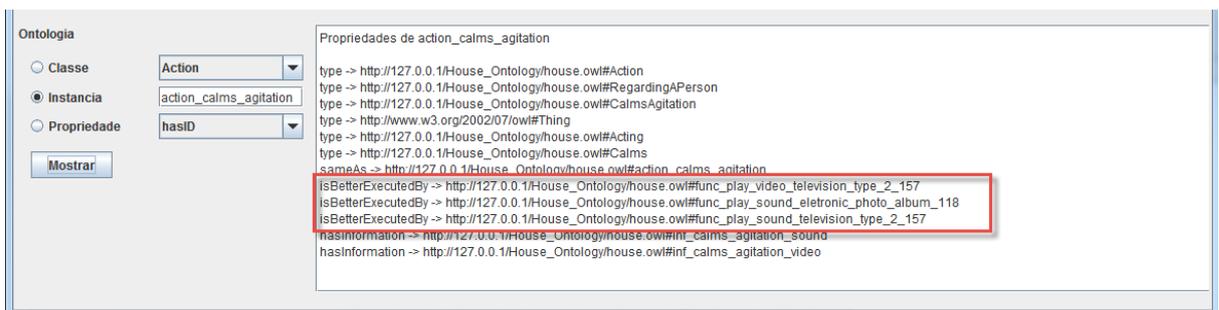
Figura 64 – Cenário 1, instância informação do tipo “Vídeo” com conteúdo do tipo *base64Binary*



Fonte: *Screenshots* obtido pela autora com base no controlador

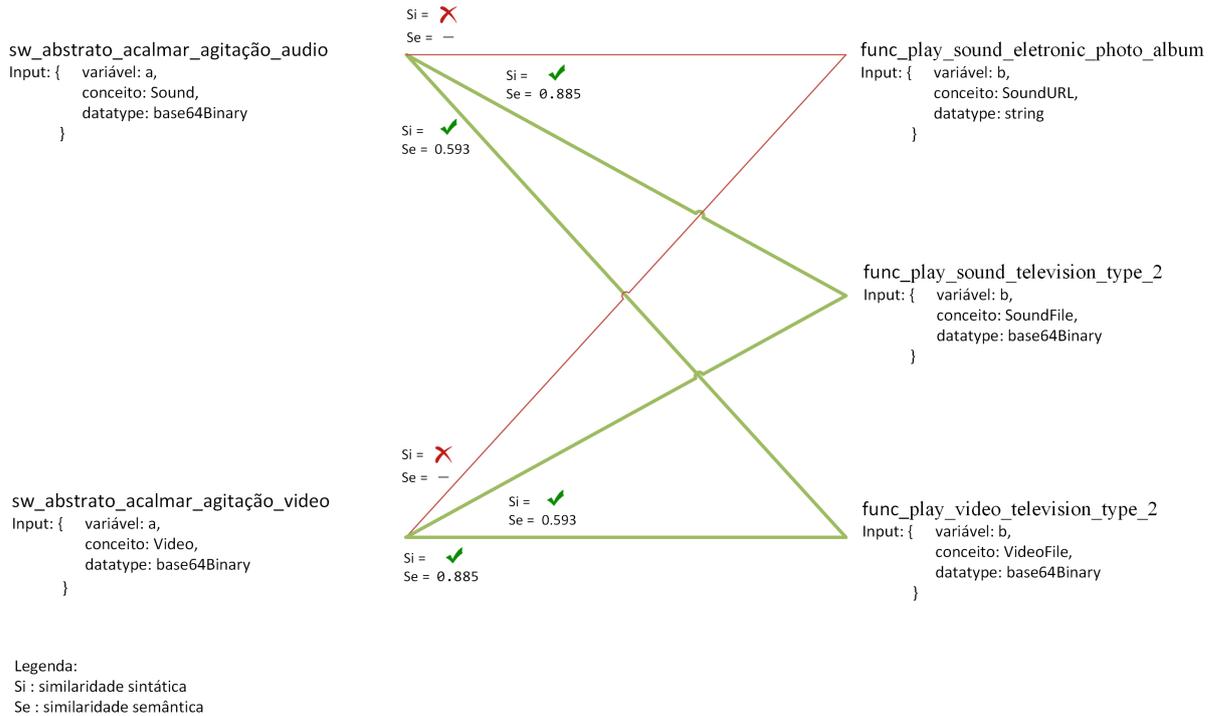
Da mesma forma que na ação anterior (da atividade (g)), a partir do resultado das regras de inferência (Figura 65), é executado a análise de similaridade semântica e sintática (Figura 66), a ordenação dos pares e aplicado o limiar (Tabela 5.4). O primeiro FuncDisp da ordenação é, então, verificado suas precondições (ter o estado ativo = *true*, ligado = *true* e ocupado = *false*). Estando elas satisfeitas, é acionado (Figura 67.b) e a ação é executada (Figura 67.a).

Figura 65 – Cenário 1, resultado das regras de inferência para atividade (h)



Fonte: *Screenshots* obtido pela autora com base no controlador

Figura 66 – Cenário 1, resultado das análises de similaridade sintática e semântica para “Ação Acalmar Agitação”



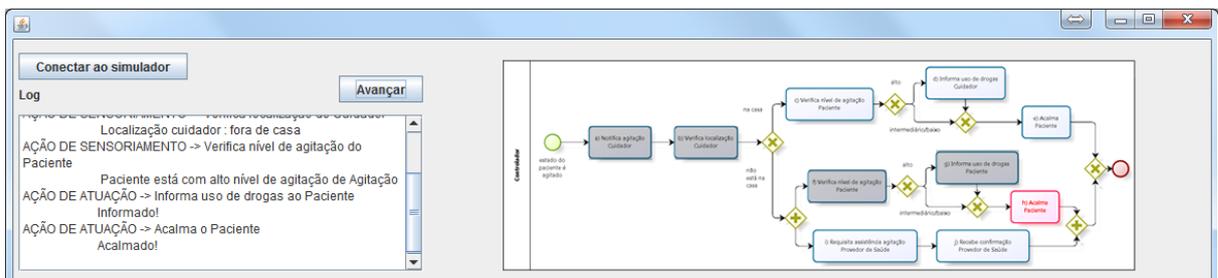
Fonte: imagem criada pela autora

Tabela 5.4 – Cenário 1, ordenação dos pares para a atividade (h) através da similaridade semântica

<i>SW abstrato</i>	<i>FuncDisp</i>	<i>Se</i>
sw_abstrato_informar_uso_drogas_audio	func_play_video_television_type_2	0.885
sw_abstrato_acalmar_agitação_video	func_play_sound_television_type_2	0.885
sw_abstrato_acalmar_agitação_audio	func_play_sound_television_type_2	0.593
sw_abstrato_informar_uso_drogas_video	func_play_video_television_type_2	0.593

Fonte: tabela criada pela autora

Figura 67 – Cenário 1, atividade (h) executada pelo controlador



Fonte: Screenshots obtido pela autora com base no controlador

Figura 68 – Cenário 1, atividade (h) executada no simulador

The screenshot displays the 'House Simulator' interface. On the left, the 'living room' panel lists various devices and their states:

- Devices:**
 - roof_light:** active(true, false) light_type(intense, dimmed, pulsating) on(true, false)
 - smart_window:** active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) locked(true, false) open(true, false)
 - door:** active(true, false) locked(true, false) open(true, false)
 - eletronic_photo_album:** active(true, false) busy(true, false) on(true, false) playing_sound(true, false) showing_image(true, false) showing_text(true, false)
 - luminaire:** active(true, false) light_type(intense, dimmed, pulsating) on(true, false)
 - sprinkler:** active(true, false) busy(true, false) on(true, false)
 - television_type_2:** active(true, false) busy(true, false) on(true, false) playing_sound(true, false) playing_video(true, false) showing_image(true, false) showing_text(true, false)
- People:**
 - John (Patient):** agitation(high, intermediate, low, none) asthma_attack(high, intermediate, low, none) dehydration(high, intermediate, low, none) diabetes_complication(high, intermediate, low, none) glucose_level(high, medium, low) shock(high, intermediate, low, none) spinal_injury(true, false) stroke(high, intermediate, low, none) temperature(high, medium, low)
- Stuff:**

On the right, the 'Organizations' panel lists:

- pharmacy (idle)
- health provider (idle)

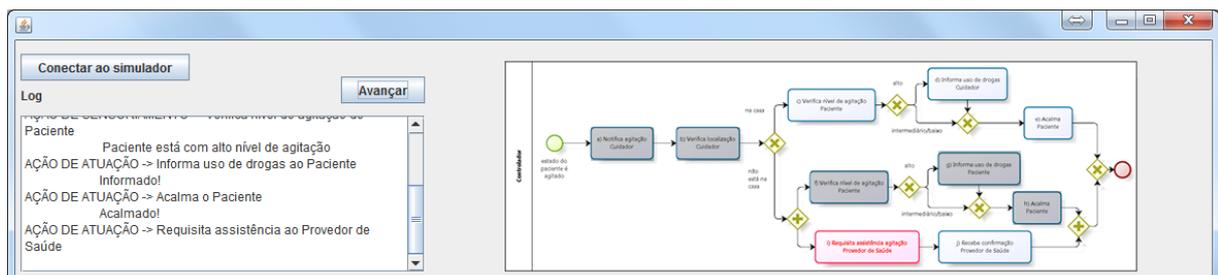
Below the organizations, a log of events is shown:

```
[23/01/2015 16:32] Caregiver Eve was notified that: O paciente está em estado de agitação
[23/01/2015 16:39] ID:118 NAME:eletronic_photo_album LOCATION:living room ACTION:play_sound_eletronic_photo_album success
[23/01/2015 16:43] ID:157 NAME:television_type_2 LOCATION:living room ACTION:play_video_television_type_2 success
```

Fonte: Screenshots obtido pela autora com base no simulador

A próxima atividade, atividade (i): <<“Requisita Assistência Agitação”, “Provedor de Saúde”>>, também é do tipo “Atuação”. Entretanto, neste caso, o objeto da atividade é único e a ação é referente a (uma) Organização (Provedor de Saúde).

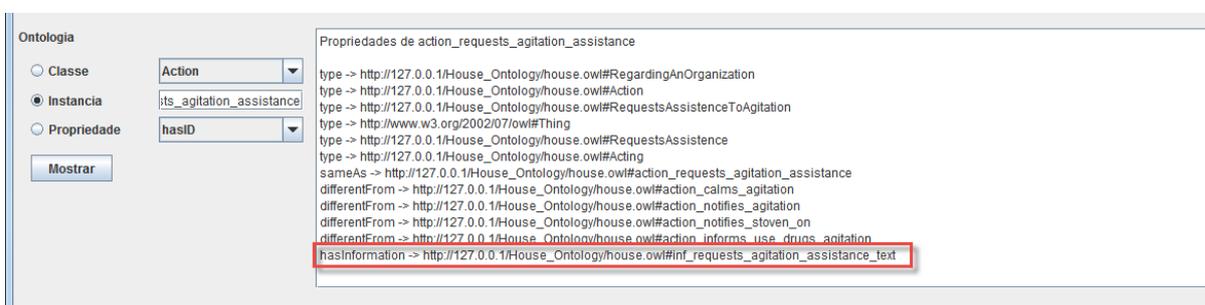
Figura 69 – Cenário 1, atividade (i) a ser executada



Fonte: Screenshots obtido pela autora com base no controlador

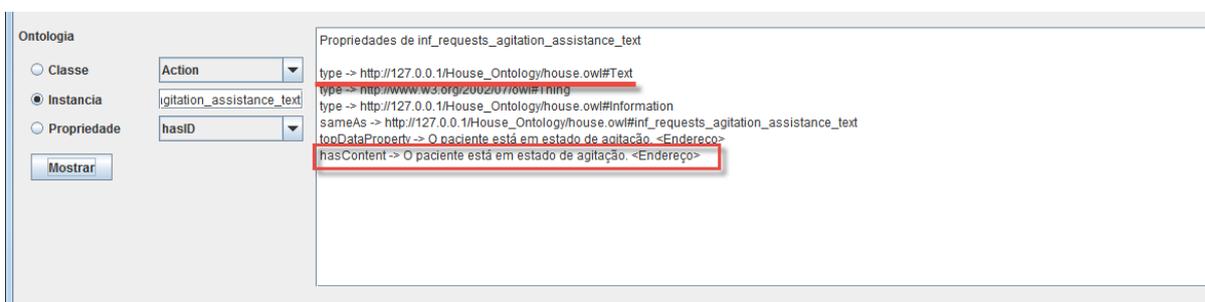
Da mesma maneira que as demais ações, esta ação também contém instância do tipo “Informações” (Figura 70), no caso, só possui uma única instancia do tipo “Texto” com conteúdo do tipo *string* (Figura 71).

Figura 70 – Cenário 1, instância de “Informação” contidas na instância de “Ação Requisita Assistência Agitação”



Fonte: Screenshots obtido pela autora com base no controlador

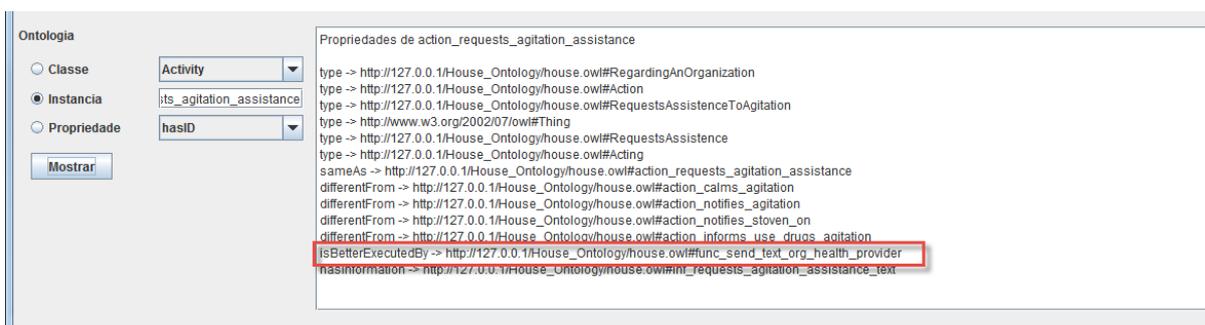
Figura 71 – Cenário 1, instância informação do tipo “Texto” com conteúdo do tipo *string*



Fonte: Screenshots obtido pela autora com base no controlador

Após executar as regras de inferência, o controlador descobre uma possível FuncDisp para executar essa ação (Figura 72). Neste cenário, apenas foi criado um FuncDisp para fazer a comunicação com o provedor de saúde, por isso, a regra só encontrou um FuncDisp.

Figura 72 – Cenário 1, resultado das regras de inferência para atividade (i)



Fonte: Screenshots obtido pela autora com base no controlador

Continuando, é feita a análise de similaridade sintática e semântica entre o serviço Web abstrato e o FuncDisp retornado das regras de inferência (Figura 73). Como ambos tem o mesmo tipo sintático e semântico, são similares. A ordenação e a aplicação do limiar é, então, realizada em cima desse único FuncDisp. Como esse FuncDisp não possui precondições, não precisa verificá-las e, o FuncDisp é acionado (Figura 75) e executado (Figura 74).

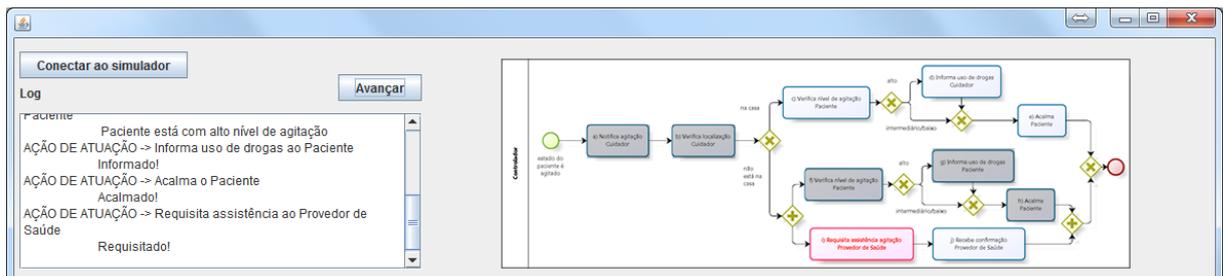
Figura 73 – Cenário 1, resultado das análises de similaridade sintática e semântica para “Ação Requisita Assistência Agitação”

<p>sw_abstrato_requisitar_assistencia_texto</p> <pre>Input: { variável: a, conceito: Text, datatype: string }</pre>	<p>Si = ✓ Se = 1.0</p>	<p>func_send_text_org_health_provider</p> <pre>Input: { variável: b, conceito: Text, datatype: string }</pre>
---	----------------------------	---

Legenda:
Si : similaridade sintática
Se : similaridade semântica

Fonte: imagem criada pela autora

Figura 74 – Cenário 1, atividade (i) executada pelo controlador



Fonte: Screenshots obtido pela autora com base no controlador

Figura 75 – Cenário 1, atividade (i) executada no simulador

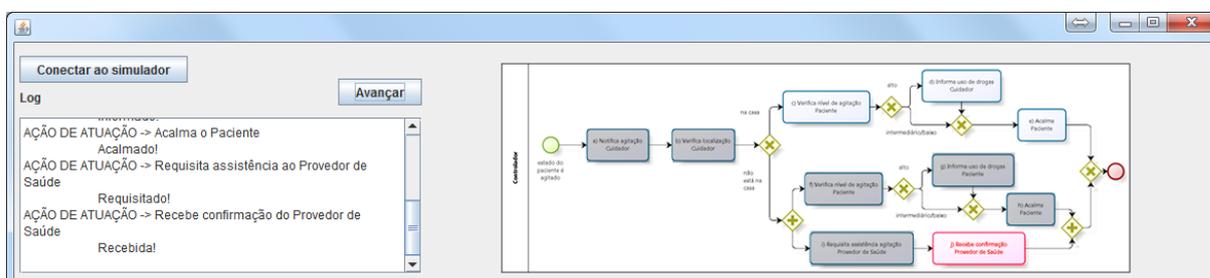
House Simulator Devices Log Finish simulation

<div style="background-color: #e9f5ff; padding: 5px; margin-bottom: 10px;">kitchen</div> <p>Devices</p> <p>stove active(true, false) on(true, false)</p> <p>curtain active(true, false) opening_level(close, semi-open, open)</p> <p>door active(true, false) locked(true, false) open(true, false)</p> <p>roof_light active(true, false) light_type(intense, dimmed, pulsating) on(true, false)</p> <p>smart_window</p>	<div style="background-color: #e9f5ff; padding: 5px; margin-bottom: 10px;">Organizations</div> <p>pharmacy (idle) health provider (idle)</p> <p>[23/01/2015 16:32] Caregiver Eve was notified that: O paciente está em estado de agitação</p> <p>[23/01/2015 16:39] ID:118 NAME:eletronic_photo_album LOCATION:living room ACTION:play_sound_eletronic_photo_album success</p> <p>[23/01/2015 16:43] ID:157 NAME:television_type_2 LOCATION:living room ACTION:play_video_television_type_2 success</p> <p>[23/01/2015 16:47] Organization health provider was called!</p>
--	--

Fonte: Screenshots obtido pela autora com base no simulador

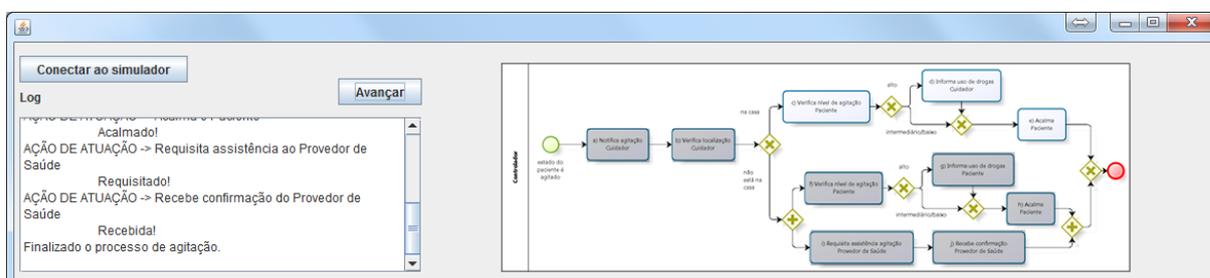
A última ação desse PN para tratar agitação é do tipo “Sensoriamento”, atividade (j): < “Recebe confirmação”, “Provedor Saúde”>, onde se recebe a confirmação da requisição ao provedor de saúde (Figura 76). E por fim, finaliza-se esse PN (Figura 77).

Figura 76 – Cenário 1, atividade (j) executada



Fonte: Screenshots obtido pela autora com base no controlador

Figura 77 – Cenário 1, fim do PN



Fonte: Screenshots obtido pela autora com base no controlador

5.2.2 Execução do Cenário 2

O cenário 2 aciona o PN para tratar quando se esquece o fogão ligado. Neste cenário, o paciente John ligou o fogão e se locomoveu para o cômodo quarto 1 e a cuidadora Eve está no quarto 2. Este cenário visa ilustrar a atividade do tipo multiobjetivo com ação referente a Pessoas.

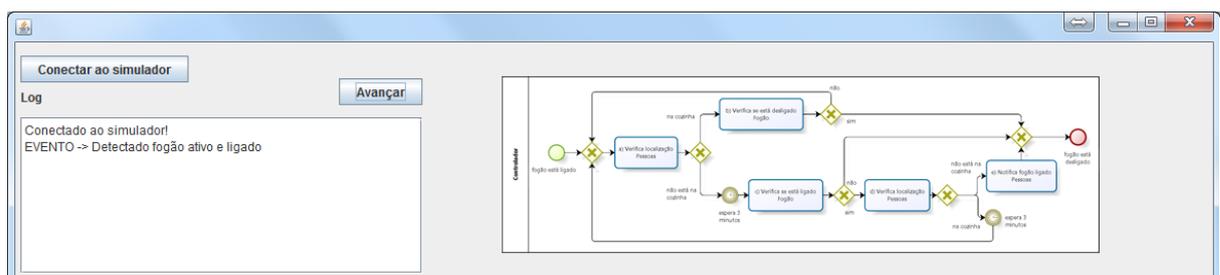
Figura 78 – Fogão no estado ligado



Fonte: *Screenshot* obtido pela autora com base no simulador

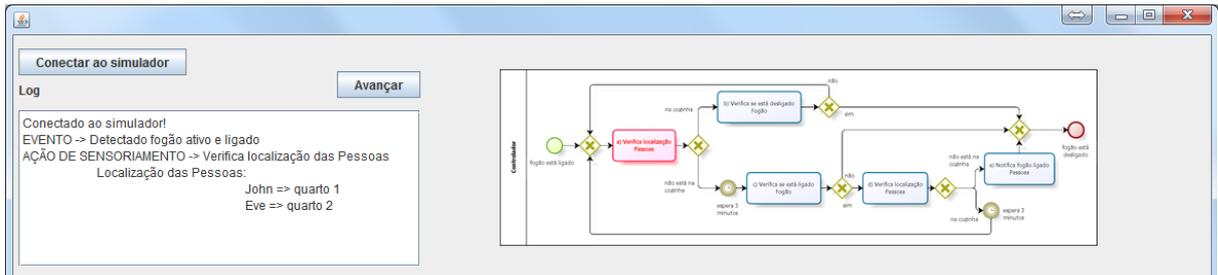
O controlador, que está sempre observando o ambiente, percebe que alguém ligou o fogão (evento inicial) (Figura 79). Após, verifica a localização das pessoas cadastradas no sistema (atividade (a): <“Verifica localização”, “Pessoas”>) (ação do tipo “Sensoriamento”) (Figura 80). Como não há ninguém na cozinha, o controlador espera um determinado tempo antes de iniciar a próxima atividade (evento de tempo) (Figura 82). Passado este tempo, o controlador verifica novamente se o fogão ainda está ligado (atividade (c): <“Verifica se está ligado”, “Fogão”>) (ação do tipo “Sensoriamento”) (Figura 82). Dado que o fogão ainda está ligado, o controlador verifica de novo se há alguém na cozinha (atividade (d): <“Verifica localização”, “Pessoas”>) (ação do tipo “Sensoriamento”) (Figura 83). Visto que nem o paciente John e a cuidadora Eve estão na cozinha, o controlador então executa a atividade (e): <“Notifica fogão ligado”, “Pessoas”> (ação do tipo “Atuação”) para as pessoas cadastradas (Figura 84). Essa é uma atividade do tipo multiobjeto com ação referente a Pessoas (Seção 3.2.3.4).

Figura 79 – Cenário 2, controlador detecta estado ligado do fogão



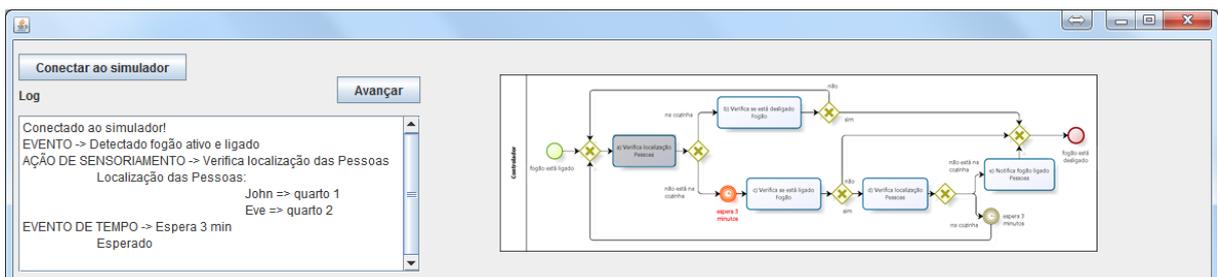
Fonte: *Screenshots* obtido pela autora com base no controlador

Figura 80 – Cenário 2, atividade (a) executada



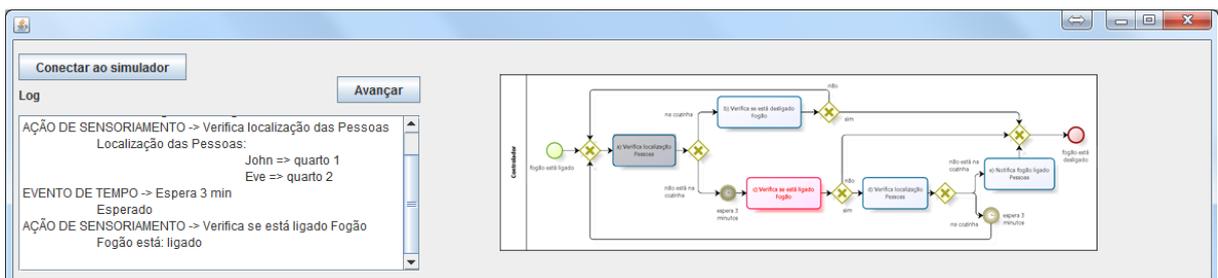
Fonte: Screenshots obtido pela autora com base no controlador

Figura 81 – Cenário 2, evento de tempo executado



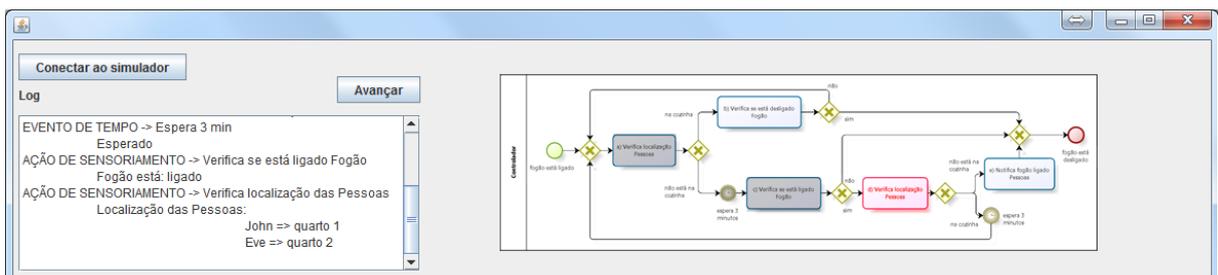
Fonte: Screenshots obtido pela autora com base no controlador

Figura 82 – Cenário 2, atividade (c) executada



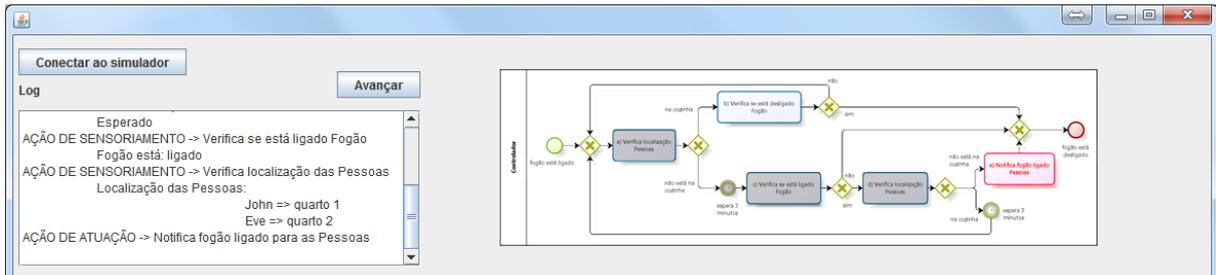
Fonte: Screenshots obtido pela autora com base no controlador

Figura 83 – Cenário 2, atividade (d) executada



Fonte: Screenshots obtido pela autora com base no controlador

Figura 84 – Cenário 2, atividade (e) executada



Fonte: Screenshots obtido pela autora com base no controlador

Para cada pessoa cadastrada no sistema, o controlador executa a atividade como se fosse do tipo objeto-único com ação referente a (uma) Pessoa. Desta forma, com alvo, primeiramente, no paciente John, atualiza a instância atividade e com as regras de inferência obteve-se 10 FuncDisp (Figura 85).

Figura 85 – Cenário 2, resultado das regras de inferência para atividade (e) foco no paciente

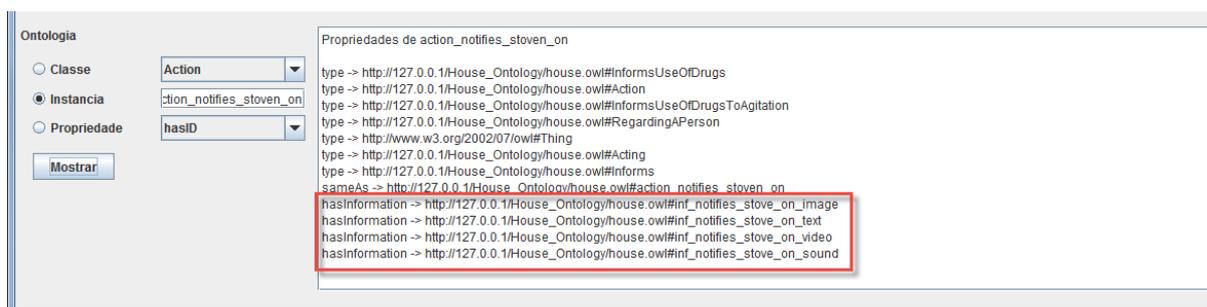
```

Propriedades de action_notifies_stoven_on
type -> http://127.0.0.1/House_Ontology/house.owl#InformUseOfDrugs
type -> http://127.0.0.1/House_Ontology/house.owl#Action
type -> http://127.0.0.1/House_Ontology/house.owl#InformUseOfDrugsToAgitation
type -> http://127.0.0.1/House_Ontology/house.owl#RegardingAPerson
type -> http://www.w3.org/2002/07/owl#Thing
type -> http://127.0.0.1/House_Ontology/house.owl#Acting
type -> http://127.0.0.1/House_Ontology/house.owl#Inform
sameAs -> http://127.0.0.1/House_Ontology/house.owl#action_notifies_stoven_on
differentFrom -> http://127.0.0.1/House_Ontology/house.owl#action_requests_agitation_assistance
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_video_television_type_1_128
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_video_computer_type_2_137
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_video_smart_pillow_130
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_video_mobile_phone_133
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_radio_clock_129
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_agenda_type_1_135
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_smart_pillow_130
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_mobile_phone_133
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_computer_type_2_137
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_television_type_1_128
hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_stove_on_image
hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_stove_on_text
hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_stove_on_video
hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_stove_on_sound
  
```

Fonte: Screenshots obtido pela autora com base no controlador

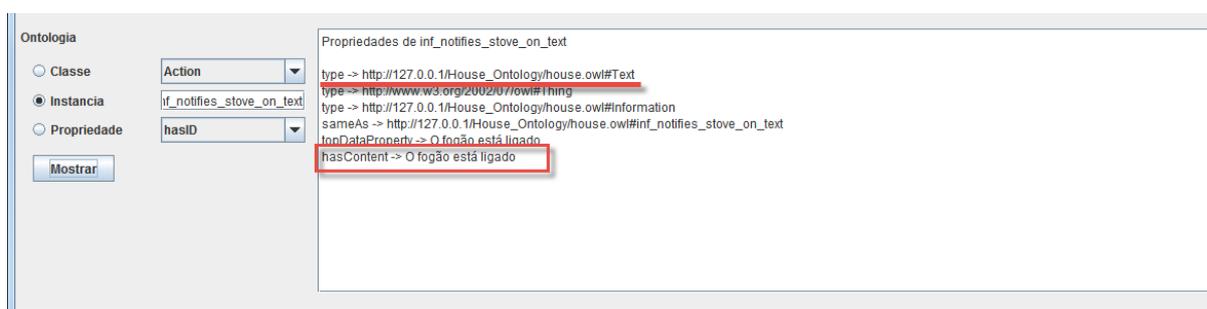
A instância de “Ação Notificar Fogão Ligado” possui 4 instâncias de “Informação” (Figura 86): “Texto” com conteúdo do tipo *string* (Figura 87), “Imagem” com conteúdo do tipo *base64Binary* (Figura 88), “Áudio” com conteúdo do tipo *base64Binary* (Figura 89), e “Vídeo” com conteúdo do tipo *base64Binary* (Figura 90). Com isso, fazendo uma análise de similaridade entre os 4 serviços Web abstratos gerados com as 10 FuncDisp, temos 40 combinações de pares. Por motivos de visibilidade, não serão mostrados os resultados da análise de similaridade sintática. Da análise de similaridade sintática, são retornados 18 pares, e com esses é, então, realizado a análise semântica. O resultado ordenado da análise semântica é mostrado na Tabela 5.5, onde os quatro primeiros pares possuem um valor semântico acima do que o limiar.

Figura 86 – Cenário 2, instâncias de “Informação” contidas na instância de “Ação Notificar Fogão Ligado”



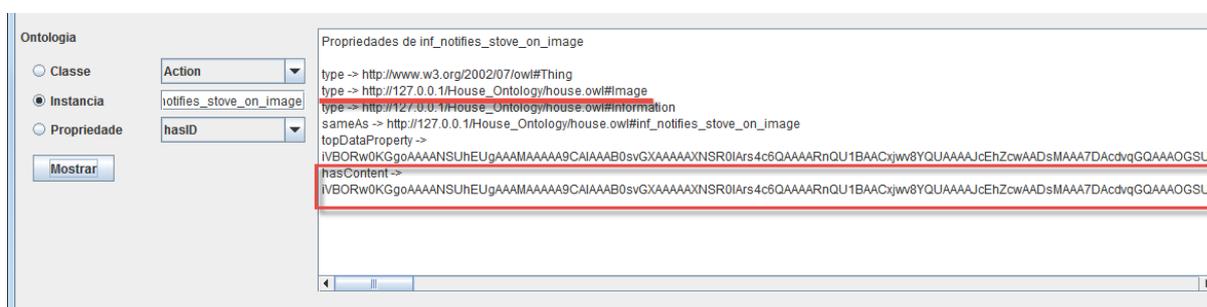
Fonte: *Screenshots* obtido pela autora com base no controlador

Figura 87 – Cenário 2, instância de “Informação” do tipo “Texto” com conteúdo do tipo *string*



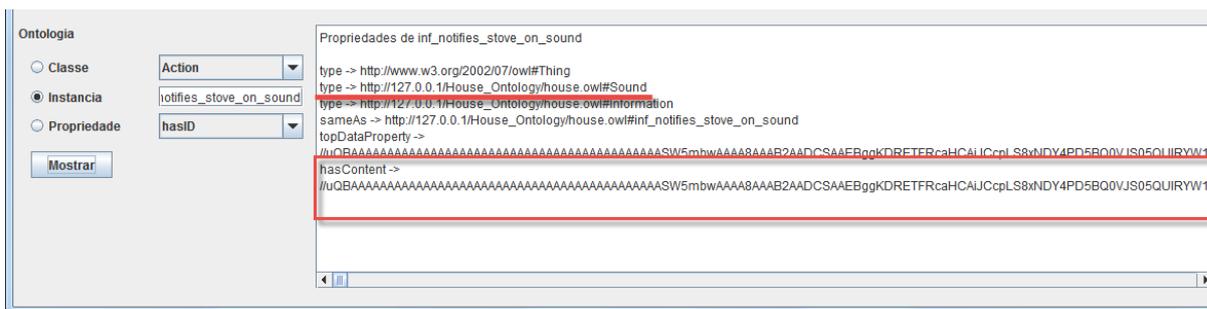
Fonte: *Screenshots* obtido pela autora com base no controlador

Figura 88 – Cenário 2, instância de “Informação” do tipo “Imagem” com conteúdo do tipo *base64Binary*



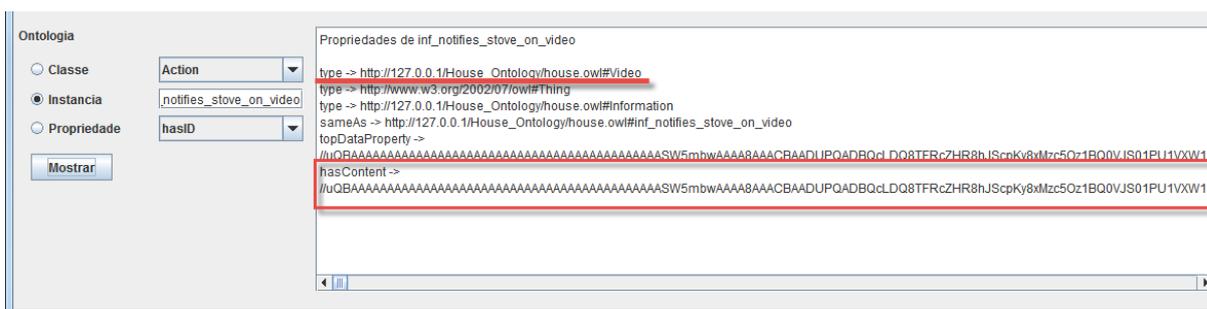
Fonte: *Screenshots* obtido pela autora com base no controlador

Figura 89 – Cenário 2, instância de “Informação” do tipo “Áudio” com conteúdo do tipo *base64Binary*



Fonte: Screenshots obtido pela autora com base no controlador

Figura 90 – Cenário 2, instância de “Informação” do tipo “Vídeo” com conteúdo do tipo *base64Binary*



Fonte: Screenshots obtido pela autora com base no controlador

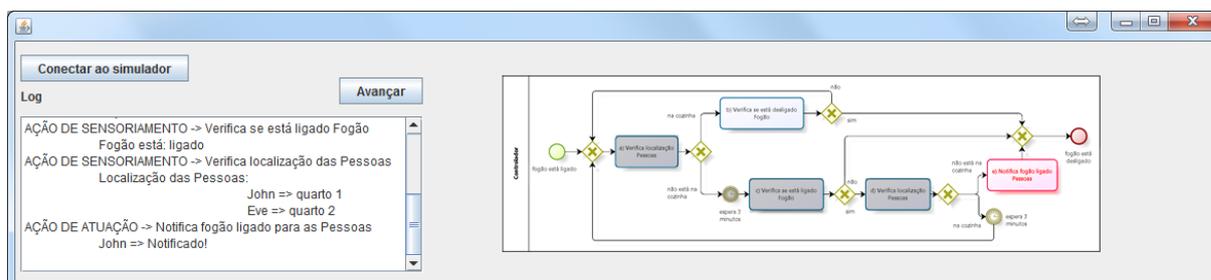
Tabela 5.5 – Cenário 2, ordenação dos pares para a atividade (e) com foco no paciente através da similaridade semântica

<i>SW abstrato</i>	<i>FuncDisp</i>	<i>Se</i>
sw_abstrato_notificar_fogão_ligado_audio	play_sound_computer_type_2_137	0,886
sw_abstrato_notificar_fogão_ligado_audio	play_sound_mobile_phone_133	0,886
sw_abstrato_notificar_fogão_ligado_video	play_video_mobile_phone_133	0,886
sw_abstrato_notificar_fogão_ligado_video	play_video_computer_type_2_137	0,886
sw_abstrato_notificar_fogão_ligado_audio	play_video_mobile_phone_133	0,594
sw_abstrato_notificar_fogão_ligado_audio	play_video_computer_type_2_137	0,594
sw_abstrato_notificar_fogão_ligado_video	play_sound_computer_type_2_137	0,594
sw_abstrato_notificar_fogão_ligado_video	play_sound_mobile_phone_133	0,594
sw_abstrato_notificar_fogão_ligado_imagem	play_sound_computer_type_2_137	0,594
sw_abstrato_notificar_fogão_ligado_imagem	play_sound_mobile_phone_133	0,594
sw_abstrato_notificar_fogão_ligado_imagem	play_video_mobile_phone_133	0,594
sw_abstrato_notificar_fogão_ligado_imagem	play_video_computer_type_2_137	0,594
sw_abstrato_notificar_fogão_ligado_texto	play_sound_television_type_1_128	0,557
sw_abstrato_notificar_fogão_ligado_texto	play_sound_smart_pillow_130	0,557
sw_abstrato_notificar_fogão_ligado_texto	play_sound_agenda_type_1_135	0,557
sw_abstrato_notificar_fogão_ligado_texto	play_sound_radio_clock_129	0,557
sw_abstrato_notificar_fogão_ligado_texto	play_video_smart_pillow_130	0,557
sw_abstrato_notificar_fogão_ligado_texto	play_video_television_type_1_128	0,557

Fonte: tabela criada pela autora

É então verificado as precondições da primeira FuncDisp da lista ordenada pelo valor semântico (*play_sound_computer_type_2_137*) (ter estado ativo = *true*, ligado = *true* e ocupado = *false*) e estando elas satisfeitas é acionado o FuncDisp (Figura 92) e a atividade (e) é parcialmente executada pelo controlador (Figura 91), pois ainda falta executar a atividade com foco na cuidadora Eve.

Figura 91 – Cenário 2, atividade (e) com foco no paciente executada pelo controlador



Fonte: Screenshots obtido pela autora com base no controlador

Figura 92 – Cenário 2, atividade (e) com foco no paciente executada no simulador

The screenshot displays the 'House Simulator' interface. At the top, there are navigation buttons: 'House Simulator', 'Devices', 'Log', and 'Finish simulation'. The main content is divided into two panels. The left panel, titled 'bedroom 1', lists various devices and their states. The right panel, titled 'Organizations', shows a list of organizations and a log entry.

bedroom 1

Devices

door
 active(true, false) locked(true, false) open(true, false)

smart_window
 active(true, false) busy(true, false) on(true, false)
 showing_image(true, false) showing_text(true, false)
 locked(true, false) open(true, false)

curtain
 active(true, false) opening_level(close, semi-open, open)

roof_light
 active(true, false) light_type(intense, dimmed, pulsating) on(true, false)

television_type_1
 active(true, false) busy(true, false) on(true, false)
 playing_sound(true, false) playing_video(true, false)
 showing_image(true, false) showing_text(true, false)

radio_clock
 active(true, false) busy(true, false) on(true, false)
 playing_sound(true, false) showing_text(true, false)

smart_pillow
 active(true, false) busy(true, false) on(true, false)
 playing_sound(true, false) playing_video(true, false)
 showing_image(true, false) showing_text(true, false)

smart_wardrobe
 active(true, false) busy(true, false) on(true, false)
 showing_image(true, false) showing_text(true, false)

sprinkler
 active(true, false) busy(true, false) on(true, false)

mobile_phone
 active(true, false) busy(true, false) on(true, false)
 playing_sound(true, false) playing_video(true, false)
 showing_image(true, false) showing_text(true, false)

lamp
 active(true, false) light_type(intense, dimmed, pulsating) on(true, false)

agenda_type_1
 active(true, false) busy(true, false) on(true, false)
 playing_sound(true, false) showing_image(true, false)
 showing_text(true, false)

computer_type_2
 active(true, false) busy(true, false) on(true, false)
 playing_sound(true, false) playing_video(true, false)
 showing_image(true, false) showing_text(true, false)

mirror
 active(true, false) busy(true, false) on(true, false)
 showing_image(true, false) showing_text(true, false)
 zoom_type(normal, in, out)

People

John (Patient)
 agitation(high, intermediate, low, none) asthma_attack(high, intermediate, low, none)
 dehydration(high, intermediate, low, none)
 diabetes_complication(high, intermediate, low, none)
 glucose_level(high, medium, low) shock(high, intermediate, low, none)
 spinal_injury(true, false) stroke(high, intermediate, low, none)
 temperature(high, medium, low)

Stuff

Organizations

pharmacy (idle)
 health provider (idle)

[05/01/2015 13:23] ID:137 NAME:computer_type_2 LOCATION:bedroom 1
 ACTION:play_sound_computer_type_2 success

Fonte: Screenshots obtido pela autora com base no simulador

Focando agora a atividade a cuidadora Eve, atualiza-se a instancia atividade, e obtém 27 FuncDisp que podem executar a ação (Figura 93). Com esses FuncDisp e dos 4 serviços Web abstratos gerados, formam-se 108 pares de combinações (não serão mostrados por motivo de falta de visibilidade). Aplicando a análise de similaridade sintática, obtém-se 43 pares similares sintaticamente, que são utilizados para fazer a análise de similaridade semântica, e o

resultado desta análise é ordenado decrescentemente e aplicado o limiar (Tabela 5.6, onde as linhas acinzentadas representam os pares que não alcançaram o limiar).

Figura 93 – Cenário 2, resultado das regras de inferência para atividade (e) com foco na cuidadora

The screenshot shows an ontology editor interface. On the left, there's a sidebar with 'Ontologia' and options for 'Classe', 'Instancia', and 'Propriedade'. The main area displays 'Propriedades de action_notifies_stoven_on'. A list of rules is shown, with a red box highlighting the 'isBetterExecutedBy' rules. The rules are as follows:

```

type -> http://127.0.0.1/House_Ontology/house.owl#InformsUseOfDrugs
type -> http://127.0.0.1/House_Ontology/house.owl#Action
type -> http://127.0.0.1/House_Ontology/house.owl#InformsUseOfDrugsToAgitation
type -> http://127.0.0.1/House_Ontology/house.owl#RegardingAPerson
type -> http://www.w3.org/2002/07/owl#Thing
type -> http://127.0.0.1/House_Ontology/house.owl#Acting
type -> http://127.0.0.1/House_Ontology/house.owl#Informs
sameAs -> http://127.0.0.1/House_Ontology/house.owl#action_notifies_stoven_on
differentFrom -> http://127.0.0.1/House_Ontology/house.owl#action_requests_antiplata_assistance
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_radio_clock_143
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_image_computer_type_1_150
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_text_television_type_2_142
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_text_smart_wardrobe_145
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_text_radio_clock_143
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_image_smart_window_139
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_image_agenda_type_2_149
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_image_mobile_phone_147
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_text_agenda_type_2_149
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_agenda_type_2_149
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_television_type_2_142
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_image_smart_wardrobe_145
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_video_smart_pillow_144
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_image_mirror_152
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_smart_pillow_144
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_text_computer_type_1_150
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_video_computer_type_1_150
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_text_mobile_phone_147
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_image_television_type_2_142
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_mobile_phone_147
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_image_smart_pillow_144
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_video_television_type_2_142
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_text_smart_pillow_144
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_text_smart_window_139
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_computer_type_1_150
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_video_mobile_phone_147
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_display_text_mirror_152
hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_stove_on_image
hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_stove_on_text
hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_stove_on_video
hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_stove_on_sound

```

Fonte: Screenshots obtido pela autora com base no controlador

Tabela 5.6 – Cenário 2, ordenação dos pares para a atividade (e) com foco na cuidadora através da similaridade semântica

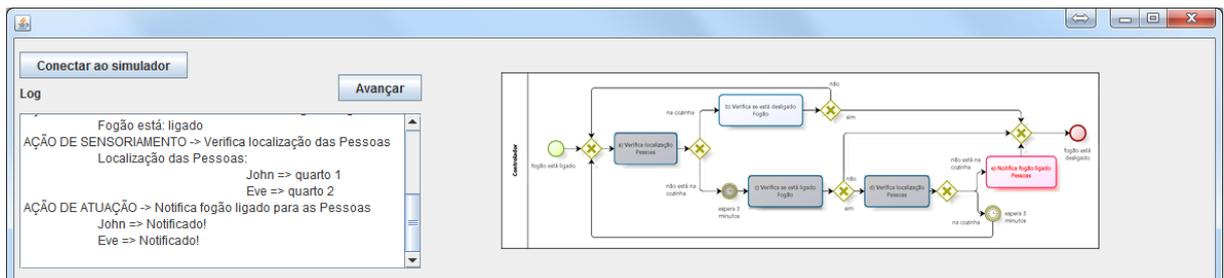
<i>SW abstrato</i>	<i>FuncDisp</i>	<i>Se</i>
sw_abstrato_notificar_fogão_ligado_texto	show_text_mirror_152	1,000
sw_abstrato_notificar_fogão_ligado_texto	show_text_smart_window_139	1,000
sw_abstrato_notificar_fogão_ligado_texto	show_text_smart_pillow_144	1,000
sw_abstrato_notificar_fogão_ligado_texto	show_text_mobile_phone_147	1,000
sw_abstrato_notificar_fogão_ligado_texto	show_text_computer_type_1_150	1,000
sw_abstrato_notificar_fogão_ligado_texto	show_text_agenda_type_2_149	1,000
sw_abstrato_notificar_fogão_ligado_texto	show_text_radio_clock_143	1,000
sw_abstrato_notificar_fogão_ligado_texto	show_text_smart_wardrobe_145	1,000
sw_abstrato_notificar_fogão_ligado_texto	show_text_television_type_2_142	1,000
sw_abstrato_notificar_fogão_ligado_audio	play_sound_mobile_phone_147	0,886
sw_abstrato_notificar_fogão_ligado_audio	play_sound_television_type_2_142	0,886
sw_abstrato_notificar_fogão_ligado_audio	play_sound_agenda_type_2_149	0,886
sw_abstrato_notificar_fogão_ligado_video	play_video_mobile_phone_147	0,886
sw_abstrato_notificar_fogão_ligado_video	play_video_television_type_2_142	0,886
sw_abstrato_notificar_fogão_ligado_imagem	show_image_television_type_2_142	0,886
sw_abstrato_notificar_fogão_ligado_imagem	show_image_mobile_phone_147	0,886
sw_abstrato_notificar_fogão_ligado_imagem	show_image_agenda_type_2_149	0,886
sw_abstrato_notificar_fogão_ligado_audio	play_video_mobile_phone_147	0,594

sw_abstrato_notificar_fogão_ligado_audio	play_video_television_type_2_142	0,594
sw_abstrato_notificar_fogão_ligado_audio	show_image_television_type_2_142	0,594
sw_abstrato_notificar_fogão_ligado_audio	show_image_mobile_phone_147	0,594
sw_abstrato_notificar_fogão_ligado_audio	show_image_agenda_type_2_149	0,594
sw_abstrato_notificar_fogão_ligado_video	play_sound_mobile_phone_147	0,594
sw_abstrato_notificar_fogão_ligado_video	show_image_television_type_2_142	0,594
sw_abstrato_notificar_fogão_ligado_video	play_sound_television_type_2_142	0,594
sw_abstrato_notificar_fogão_ligado_video	play_sound_agenda_type_2_149	0,594
sw_abstrato_notificar_fogão_ligado_video	show_image_mobile_phone_147	0,594
sw_abstrato_notificar_fogão_ligado_video	show_image_agenda_type_2_149	0,594
sw_abstrato_notificar_fogão_ligado_imagem	play_video_mobile_phone_147	0,594
sw_abstrato_notificar_fogão_ligado_imagem	play_video_television_type_2_142	0,594
sw_abstrato_notificar_fogão_ligado_imagem	play_sound_mobile_phone_147	0,594
sw_abstrato_notificar_fogão_ligado_imagem	play_sound_television_type_2_142	0,594
sw_abstrato_notificar_fogão_ligado_imagem	play_sound_agenda_type_2_149	0,594
sw_abstrato_notificar_fogão_ligado_texto	play_sound_computer_type_1_150	0,557
sw_abstrato_notificar_fogão_ligado_texto	show_image_smart_pillow_144	0,557
sw_abstrato_notificar_fogão_ligado_texto	play_video_computer_type_1_150	0,557
sw_abstrato_notificar_fogão_ligado_texto	play_sound_smart_pillow_144	0,557
sw_abstrato_notificar_fogão_ligado_texto	show_image_mirror_152	0,557
sw_abstrato_notificar_fogão_ligado_texto	play_video_smart_pillow_144	0,557
sw_abstrato_notificar_fogão_ligado_texto	show_image_smart_wardrobe_145	0,557
sw_abstrato_notificar_fogão_ligado_texto	show_image_smart_window_139	0,557
sw_abstrato_notificar_fogão_ligado_texto	show_image_computer_type_1_150	0,557
sw_abstrato_notificar_fogão_ligado_texto	play_sound_radio_clock_143	0,557

Fonte: tabela criada pela autora

O controlador então verifica se as precondições da primeira FuncDisp da ordenação (*show_text_mirror_152*) estão satisfeitas (ter o estado ativo = *true* e ocupado = *false*). E estando satisfeita as precondições, o controlador executa totalmente a atividade (e) (Figura 94), e acionando o FuncDisp (Figura 95). E por fim termina esse PN (Figura 96).

Figura 94 – Cenário 2, atividade (e) com foco na cuidadora executada pelo controlador



Fonte: Screenshots obtido pela autora com base no controlador

Figura 95 – Cenário 2, atividade (e) com foco na cuidadora executada no simulador

The screenshot shows the 'House Simulator' interface. The top bar includes 'House Simulator', 'Devices', 'Log', and a 'Finish simulation' button. The main content is divided into two panels:

- bedroom 2:** A list of devices and their states.
 - door:** active(true, false) locked(true, false) open(true, false)
 - smart_window:** active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) locked(true, false) open(true, false)
 - curtain:** active(true, false) opening_level(close, semi-open, open)
 - roof_light:** active(true, false) light_type(intense, dimmed, pulsating) on(true, false)
 - television_type_2:** active(true, false) busy(true, false) on(true, false) playing_sound(true, false) playing_video(true, false) showing_image(true, false) showing_text(true, false)
 - radio_clock:** active(true, false) busy(true, false) on(true, false) playing_sound(true, false) showing_text(true, false)
 - smart_pillow:** active(true, false) busy(true, false) on(true, false) playing_sound(true, false) playing_video(true, false) showing_image(true, false) showing_text(true, false)
 - smart_wardrobe:** active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false)
 - sprinkler:** active(true, false) busy(true, false) on(true, false)
 - mobile_phone:** active(true, false) busy(true, false) on(true, false) playing_sound(true, false) playing_video(true, false) showing_image(true, false) showing_text(true, false)
 - lamp:** active(true, false) light_type(intense, dimmed, pulsating) on(true, false)
 - agenda_type_2:** active(true, false) busy(true, false) on(true, false) playing_sound(true, false) showing_image(true, false) showing_text(true, false)
 - computer_type_1:** active(true, false) busy(true, false) on(true, false) playing_sound(true, false) playing_video(true, false) showing_image(true, false) showing_text(true, false)
 - mirror:** active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) zoom_type(normal, in, out)
- People:** Eve (Caregiver)
- Stuff:**

Organizations:

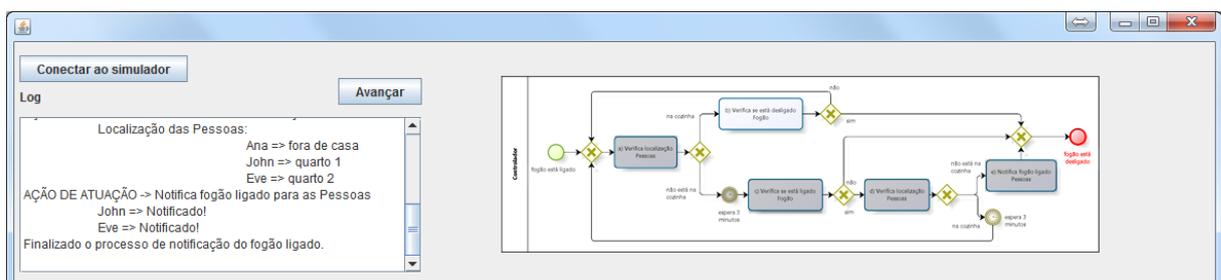
- pharmacy (idle)
- health provider (idle)

Log:

```
[05/01/2015 13:23] ID:137 NAME:computer_type_2 LOCATION:bedroom 1
ACTION:play_sound_computer_type_2 success
[05/01/2015 13:27] ID:152 NAME:mirror LOCATION:bedroom 2
ACTION:show_text_mirror success
```

Fonte: Screenshots obtido pela autora com base no simulador

Figura 96 – Cenário 2, fim do PN



Fonte: Screenshots obtido pela autora com base no controlador

5.2.3 Execução do Cenário 3

Neste cenário espera-se demonstrar a abordagem para o caso de atividade com objeto-único com ação referente a Dispositivo (Seção 3.2.3.5). Com isso, inicialmente, o paciente John entra no quarto de dia, onde as janelas estão fechadas e a luz do teto está apagada. A Figura 97 mostra os estados dos dispositivos janela inteligente (estado aberto = *false*) e luz do teto (estado ligado = *false*).

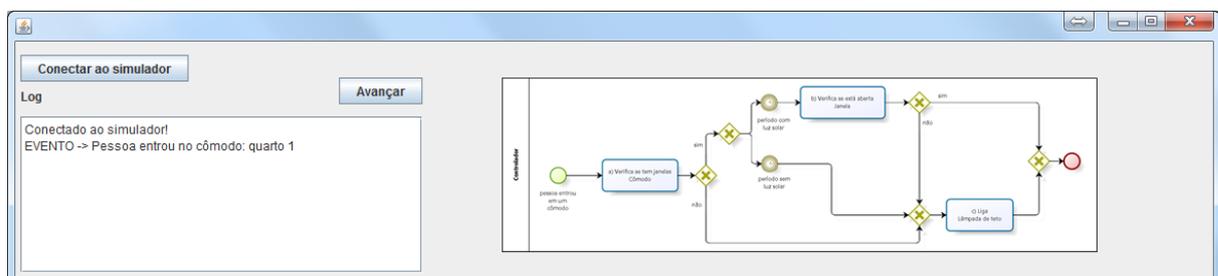
Figura 97 – Janela inteligente no estado fechado e a luz do teto no estado desligado



Fonte: Screenshot obtido pela autora com base no simulador

Desta forma, o controlador ao detectar que o paciente John entra no cômodo, ativa o PN de controle de iluminação (Seção 4.3) (Figura 98).

Figura 98 – Cenário 3, detectado quando o paciente entra no cômodo

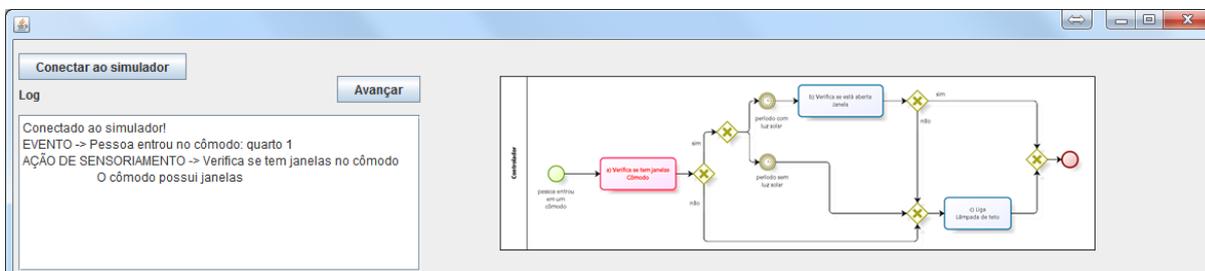


Fonte: Screenshots obtido pela autora com base no controlador

A primeira atividade, atividade (a): < “Verifica se tem janela”, “Cômodo”>, tem ação do tipo “Sensoriamento”, onde se verifica se o cômodo, onde o paciente John entrou, possui janelas (Figura 99). Como há janelas, o controlador, então, verifica se o período do dia possui luz solar (evento de tempo) (Figura 100). Dado que, há luz solar, é então verificado se a janela

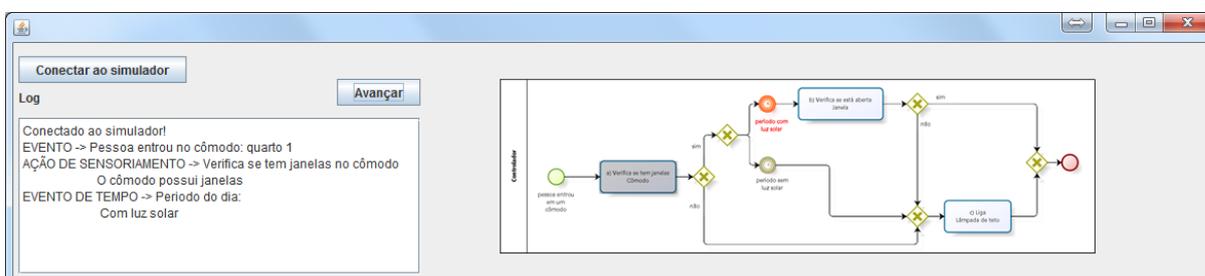
está aberta (atividade (b): <“Verifica se está aberta”, “Janela”>) (ação do tipo “Sensoriamento”) (Figura 101).

Figura 99 – Cenário 3, atividade (a) executada



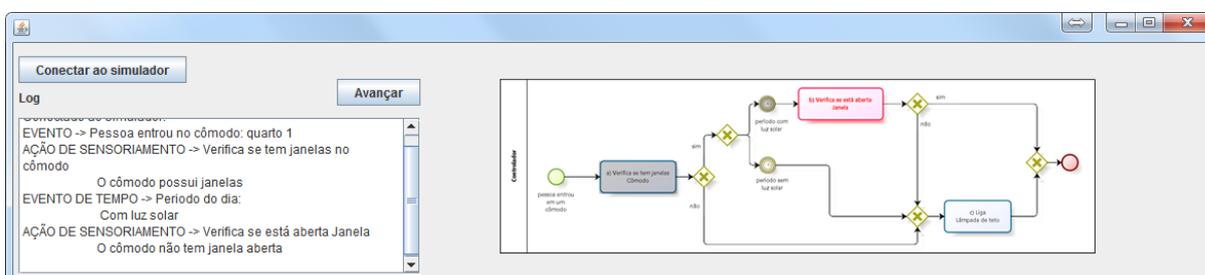
Fonte: Screenshots obtido pela autora com base no controlador

Figura 100 Cenário 3, evento de tempo executado



Fonte: Screenshots obtido pela autora com base no controlador

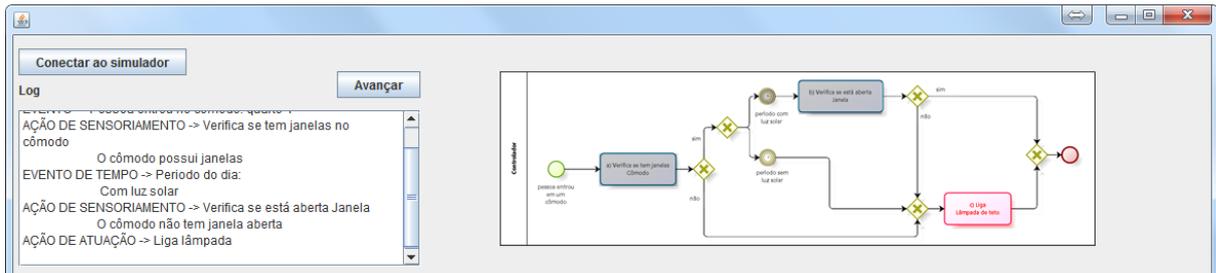
Figura 101 – Cenário 3, atividade (b) executada



Fonte: Screenshots obtido pela autora com base no controlador

Visto que não há janela aberta no cômodo, é acionada a atividade (c): <“Liga”, “Luz do Teto”> para ligar a luz do teto (ação do tipo “Atuação”) (Figura 102).

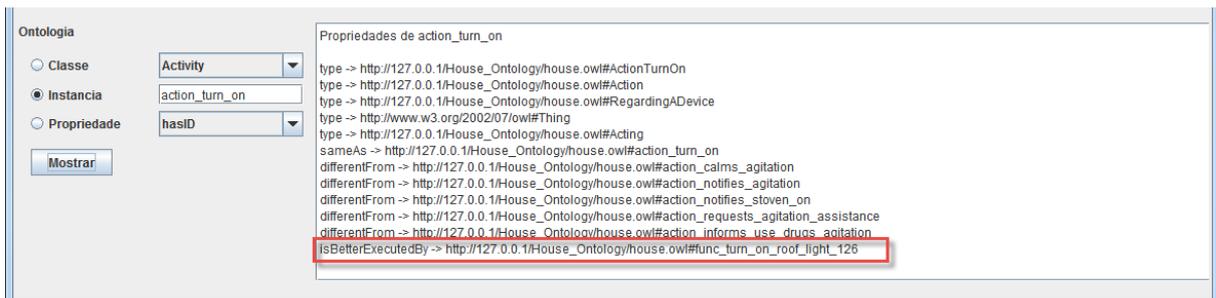
Figura 102 – Cenário 3, atividade (c) a ser executada



Fonte: Screenshots obtido pela autora com base no controlador

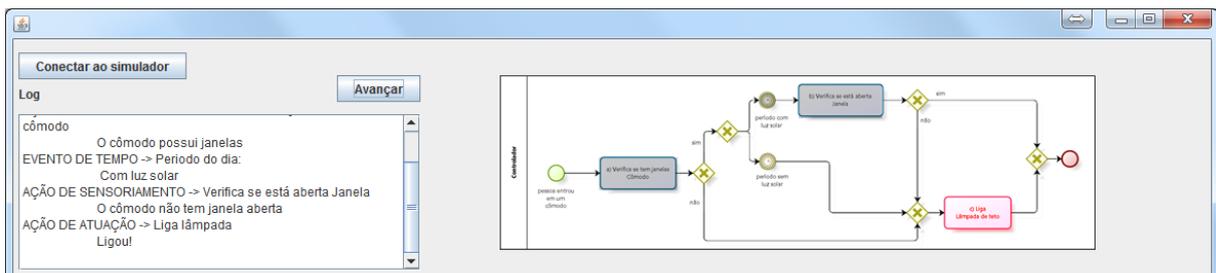
É, então, atualizada a atividade na ontologia, e é obtido o resultado das regras de inferência (Figura 103). Como não há nenhuma informação contida nesta ação de ligar (Figura 103), não é gerado serviços Web abstratos, e desta forma, a análise de similaridade sintática e semântica não são realizadas. Ainda, são verificadas as precondições do FuncDisp retornado pela inferência (*fun_turn_on_roof_light*) (ter o estado ligado = *false*). Como essa condição está satisfeita, é acionada a primeira FuncDisp (*fun_turn_on_roof_light*) no simulador (Figura 105) e o controlador executa a ação (c) (Figura 104.a), finalizando o PN (Figura 106).

Figura 103 – Cenário 3, resultado das regras de inferência para atividade (c)



Fonte: Screenshots obtido pela autora com base no controlador

Figura 104 – Cenário 3, atividade (c) executada pelo controlador



Fonte: Screenshots obtido pela autora com base no controlador

Figura 105 – Cenário 3, atividade (c) executada no simulador

The screenshot shows the 'House Simulator' interface. The top navigation bar includes 'House Simulator', 'Devices', 'Log', and 'Finish simulation'. The main content is divided into two panels:

bedroom 1

Devices

- door**
active(true, false) locked(true, false) open(true, false)
- smart_window**
active(true, false) busy(true, false) on(true, false)
showing_image(true, false) showing_text(true, false)
locked(true, false) open(true, false)
- curtain**
active(true, false) opening_level(close, semi-open, open)
- roof_light**
active(true, false) light_type(intense, dimmed, pulsating) on(true, false)
- television_type_1**
active(true, false) busy(true, false) on(true, false)
playing_sound(true, false) playing_video(true, false)
showing_image(true, false) showing_text(true, false)
- radio_clock**
active(true, false) busy(true, false) on(true, false)
playing_sound(true, false) showing_text(true, false)
- smart_pillow**
active(true, false) busy(true, false) on(true, false)
playing_sound(true, false) playing_video(true, false)
showing_image(true, false) showing_text(true, false)
- smart_wardrobe**
active(true, false) busy(true, false) on(true, false)
showing_image(true, false) showing_text(true, false)
- sprinkler**
active(true, false) busy(true, false) on(true, false)
- mobile_phone**
active(true, false) busy(true, false) on(true, false)
playing_sound(true, false) playing_video(true, false)
showing_image(true, false) showing_text(true, false)
- lamp**
active(true, false) light_type(intense, dimmed, pulsating) on(true, false)
- agenda_type_1**
active(true, false) busy(true, false) on(true, false)
playing_sound(true, false) showing_image(true, false)
showing_text(true, false)
- computer_type_2**
active(true, false) busy(true, false) on(true, false)
playing_sound(true, false) playing_video(true, false)
showing_image(true, false) showing_text(true, false)
- mirror**
active(true, false) busy(true, false) on(true, false)
showing_image(true, false) showing_text(true, false)
zoom_type(normal, in, out)

People

John (Patient)

- agitation(high, intermediate, low, none) asthma_attack(high, intermediate, low, none) dehydration(high, intermediate, low, none) diabetes_complication(high, intermediate, low, none) glucose_level(high, medium, low) shock(high, intermediate, low, none) spinal_injury(true, false) stroke(high, intermediate, low, none) temperature(high, medium, low)

Stuff

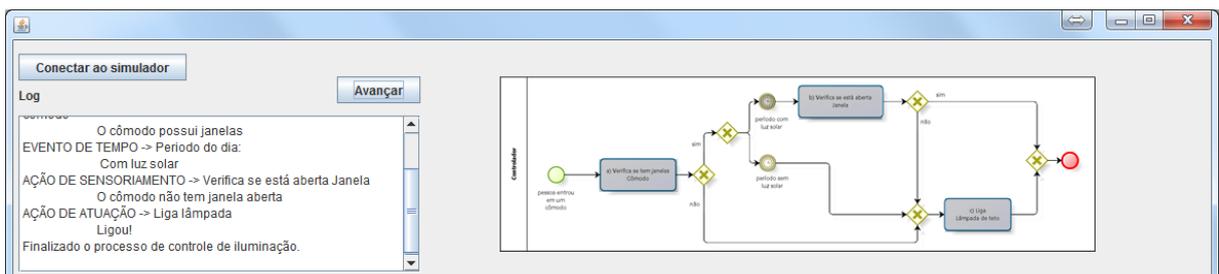
Organizations

- pharmacy (idle)
- health provider (idle)

Log entry: [06/01/2015 16:22] ID:126 NAME:roof_light LOCATION:bedroom 1 ACTION:turn_on_roof_light success

Fonte: Screenshots obtido pela autora com base no simulador

Figura 106 – Cenário 3, fim do PN

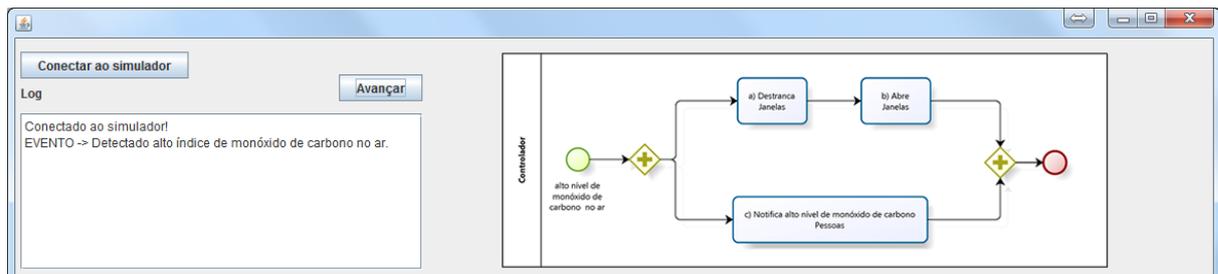


Fonte: Screenshots obtido pela autora com base no controlador

5.2.4 Execução do Cenário 4

O cenário 4 visa apresentar o comportamento da abordagem para o caso de atividade de multiobjeto com ação referente a Dispositivos. Para isto, em um cenário onde o paciente John está na cozinha e a cuidadora Eve está no banheiro, é detectado um alto nível de monóxido de carbono no ar, e acionado o PN para tratar esta situação (Figura 107).

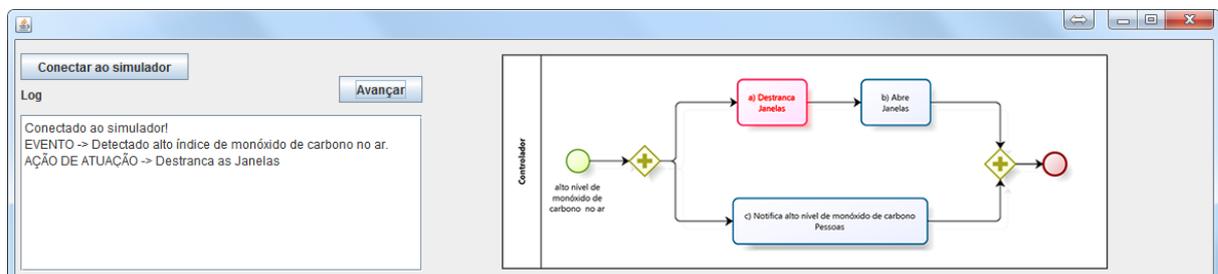
Figura 107 – Cenário 4, detectado alto índice de monóxido de carbono no ar



Fonte: Screenshots obtido pela autora com base no controlador

A primeira atividade a ser executada por este PN é destrancar janelas (atividade (a) : < “Destrançar”, “Janelas”>) (Figura 108), com ação do tipo “Atuação”, onde o objeto alvo da ação são todas as janelas da casa (multiobjeto com ação referente a Dispositivos).

Figura 108 – Cenário 4, atividade (a) a ser executada



Fonte: Screenshots obtido pela autora com base no controlador

Para executar então a atividade (a), é atualizada a instância de “Atividade” na ontologia, e com isto é obtido um conjunto resultado através das regras de inferência (Figura 109). Como a instância de “Ação Destrançar” não possui instâncias de “Informação”, não é criado serviços Web abstratos e não é realizado o *matching* sintático e semântico. Como a atividade é do tipo multiobjeto, todos os FuncDisp deverão ser acionados, logo para todos os FuncDisp é verificado suas condições (ter o estado trancado = *true*). E para os FuncDisp com condições

satisfeitas (no caso todos) são acionados no simulador (Figura 111) e a ação é executada no controlador (Figura 110).

Figura 109 – Cenário 4, resultado das regras de inferência para atividade (a)

The screenshot shows an ontology editor interface. On the left, there are radio buttons for 'Classe', 'Instancia', and 'Propriedade', with 'Instancia' selected. Below these are dropdown menus for 'Activity' (set to 'action_unlock') and 'hasID' (set to 'hasID'). A 'Mostrar' button is also present. The main area displays the 'Propriedades de action_unlock' with a list of URIs and relationships. The last five lines of the list are highlighted with a red box:

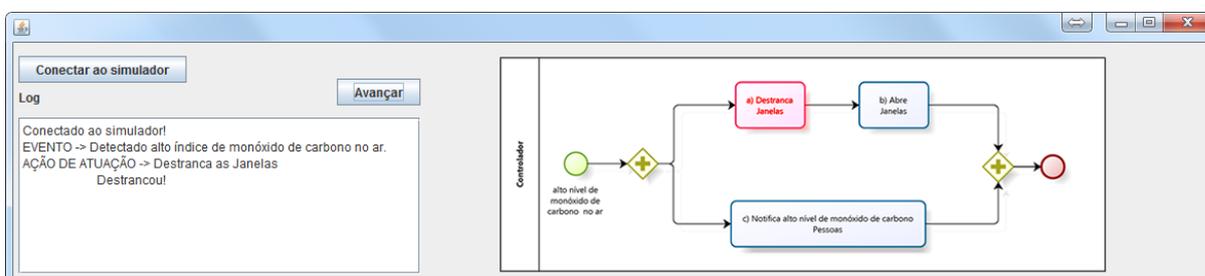
```

isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_unlock_smart_window_100
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_unlock_smart_window_116
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_unlock_smart_window_123
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_unlock_smart_window_97
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_unlock_smart_window_139

```

Fonte: Screenshots obtido pela autora com base no controlador

Figura 110 – Cenário 4, atividade (a) executada pelo controlador



Fonte: Screenshots obtido pela autora com base no controlador

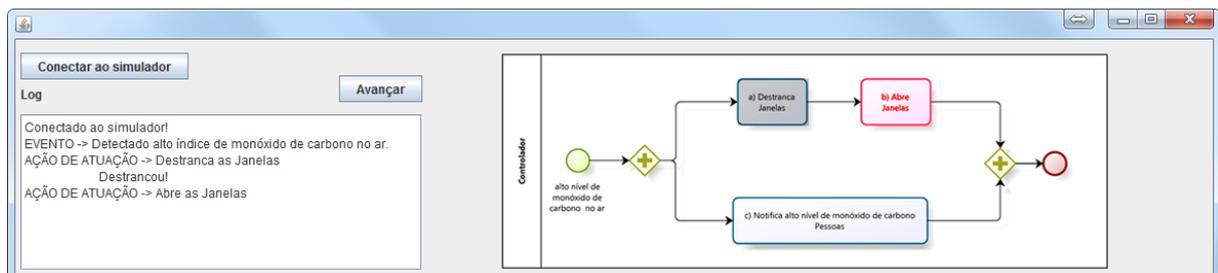
Figura 111 – Cenário 4, atividade (a) executada no simulador

kitchen	bathroom	bedroom 1
Devices stove active(true, false) on(true, false) curtain active(true, false) opening_level(close, semi-open, open) door active(true, false) locked(true, false) open(true, false) roof_light active(true, false) light_type(intense, dimmed, pulsating) on(true, false) smart_window active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) locked(true, false) open(true, false) exhauster active(true, false) on(true, false) microwave active(true, false) busy(true, false) on(true, false) playing_sound(true, false) showing_text(true, false) oven active(true, false) on(true, false)	Devices door active(true, false) locked(true, false) open(true, false) roof_light active(true, false) light_type(intense, dimmed, pulsating) on(true, false) tap active(true, false) open(true, false) shower active(true, false) on(true, false) latrine active(true, false) busy(true, false) on(true, false) mirror active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) zoom_type(normal, in, out) smart_window active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) locked(true, false) open(true, false) sprinkler	Devices door active(true, false) locked(true, false) open(true, false) smart_window active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) locked(true, false) open(true, false) curtain active(true, false) opening_level(close, semi-open, open) roof_light active(true, false) light_type(intense, dimmed, pulsating) on(true, false) television_type_1 active(true, false) busy(true, false) on(true, false) playing_sound(true, false) playing_video(true, false) showing_image(true, false) showing_text(true, false) radio_clock active(true, false) busy(true, false) on(true, false) playing_sound(true, false) showing_text(true, false) smart_pillow active(true, false) busy(true, false) on(true, false) playing_sound(true, false) playing_video(true, false)
living room	bedroom 2	Organizations
Devices roof_light active(true, false) light_type(intense, dimmed, pulsating) on(true, false) smart_window active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) locked(true, false) open(true, false) door active(true, false) locked(true, false) open(true, false) eletronic_photo_album active(true, false) busy(true, false) on(true, false) playing_sound(true, false) showing_image(true, false) showing_text(true, false) luminaire active(true, false) light_type(intense, dimmed, pulsating) on(true, false)	Devices door active(true, false) locked(true, false) open(true, false) smart_window active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) locked(true, false) open(true, false) curtain active(true, false) opening_level(close, semi-open, open) roof_light active(true, false) light_type(intense, dimmed, pulsating) on(true, false) television_type_2 active(true, false) busy(true, false) on(true, false) playing_sound(true, false) playing_video(true, false) showing_image(true, false) showing_text(true, false) radio_clock	pharmacy (idle) health provider (idle) [09/01/2015 19:24] ID:100 NAME:smart_window LOCATION:kitchen ACTION:unlock_smart_window success [09/01/2015 19:24] ID:116 NAME:smart_window LOCATION:bathroom ACTION:unlock_smart_window success [09/01/2015 19:24] ID:123 NAME:smart_window LOCATION:bedroom 1 ACTION:unlock_smart_window success [09/01/2015 19:24] ID:97 NAME:smart_window LOCATION:living room ACTION:unlock_smart_window success [09/01/2015 19:24] ID:139 NAME:smart_window LOCATION:bedroom 2 ACTION:unlock_smart_window success

Fonte: Conjunto de *screenshots* obtido pela autora com base no simulador

A próxima atividade (atividade (b): <“Abrir”, “Janelas”>) também é do tipo multiobjeto com ação referente a Dispositivos (Figura 112). Novamente, é atualizada a instância de “Atividade”, e obtido através das regras de inferência o conjunto dos FuncDisp que podem executar a ação (Figura 112).

Figura 112 – Cenário 4, atividade (b) a ser executada



Fonte: *Screenshots* obtido pela autora com base no controlador

Figura 113 – Cenário 4, resultado das regras de inferência para atividade (b)

The screenshot shows an ontology editor interface. On the left, there are radio buttons for 'Classe', 'Instancia', and 'Propriedade', with 'Instancia' selected. Below them is a 'Mostrar' button. The main area is titled 'Propriedades de action_open' and contains a list of URIs and their relationships. A red box highlights the following properties:

```

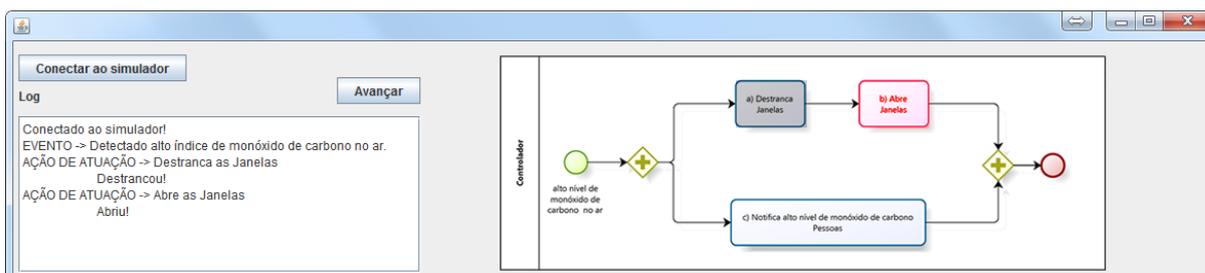
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_open_smart_window_123
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_open_smart_window_97
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_open_smart_window_100
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_open_smart_window_139
isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_open_smart_window_116

```

Fonte: *Screenshots* obtido pela autora com base no controlador

Como a instância de “Ação Abrir” não possui instâncias de “Informação”, novamente não é criado serviços Web abstratos nem realizado o *matching* sintático e semântico, sendo verificado para todos as FuncDisp retornados pelas regras de inferência suas precondições (ter o estado aberto = *false* e trancado = *false*). Com as precondições satisfeitas, é então, acionado todos esses FuncDisp (Figura 115) e a atividade é executado no controlador (Figura 114).

Figura 114 – Cenário 4, atividade (b) executada pelo controlador



Fonte: *Screenshots* obtido pela autora com base no controlador

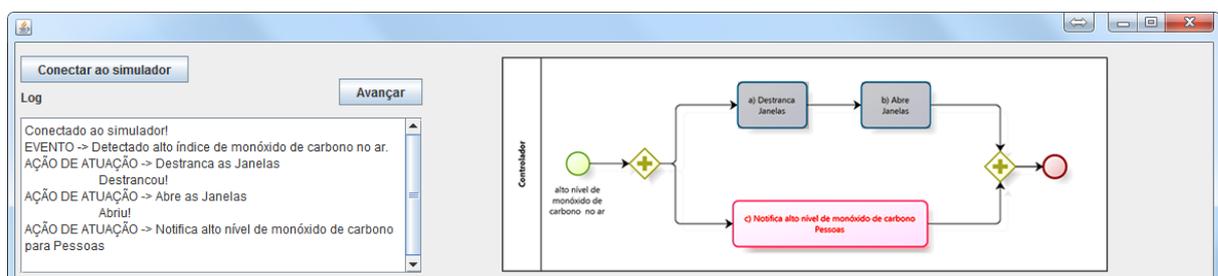
Figura 115 – Cenário 4, atividade (b) executada no simulador

kitchen	bathroom	bedroom 1
Devices stove active(true, false) on(true, false) curtain active(true, false) opening_level(close, semi-open, open) door active(true, false) locked(true, false) open(true, false) roof_light active(true, false) light_type(intense, dimmed, pulsating) on(true, false) smart_window active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) locked(true, false) open(true, false) exhauster active(true, false) on(true, false) microwave active(true, false) busy(true, false) on(true, false) playing_sound(true, false) showing_text(true, false) oven active(true, false) on(true, false)	Devices door active(true, false) locked(true, false) open(true, false) roof_light active(true, false) light_type(intense, dimmed, pulsating) on(true, false) tap active(true, false) open(true, false) shower active(true, false) on(true, false) latrine active(true, false) busy(true, false) on(true, false) mirror active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) zoom_type(normal, in, out) smart_window active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) locked(true, false) open(true, false) sprinkler	Devices door active(true, false) locked(true, false) open(true, false) smart_window active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) locked(true, false) open(true, false) curtain active(true, false) opening_level(close, semi-open, open) roof_light active(true, false) light_type(intense, dimmed, pulsating) on(true, false) television_type_1 active(true, false) busy(true, false) on(true, false) playing_sound(true, false) playing_video(true, false) showing_image(true, false) showing_text(true, false) radio_clock active(true, false) busy(true, false) on(true, false) playing_sound(true, false) showing_text(true, false) smart_pillow active(true, false) busy(true, false) on(true, false) playing_sound(true, false) playing_video(true, false)
living room	bedroom 2	Organizations
Devices roof_light active(true, false) light_type(intense, dimmed, pulsating) on(true, false) smart_window active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) locked(true, false) open(true, false) door active(true, false) locked(true, false) open(true, false) eletronic_photo_album active(true, false) busy(true, false) on(true, false) playing_sound(true, false) showing_image(true, false) showing_text(true, false) luminaire active(true, false) light_type(intense, dimmed, pulsating) on(true, false) sprinkler active(true, false) busy(true, false) on(true, false) television_type_2 active(true, false) busy(true, false) on(true, false)	Devices door active(true, false) locked(true, false) open(true, false) smart_window active(true, false) busy(true, false) on(true, false) showing_image(true, false) showing_text(true, false) locked(true, false) open(true, false) curtain active(true, false) opening_level(close, semi-open, open) roof_light active(true, false) light_type(intense, dimmed, pulsating) on(true, false) television_type_2 active(true, false) busy(true, false) on(true, false) playing_sound(true, false) playing_video(true, false) showing_image(true, false) showing_text(true, false) radio_clock active(true, false) busy(true, false) on(true, false) playing_sound(true, false) showing_text(true, false) smart_pillow active(true, false) busy(true, false) on(true, false)	pharmacy (idle) health provider (idle) [09/01/2015 19:24] ID:100 NAME:smart_window LOCATION:kitchen ACTION:unlock_smart_window success [09/01/2015 19:24] ID:116 NAME:smart_window LOCATION:bathroom ACTION:unlock_smart_window success [09/01/2015 19:24] ID:123 NAME:smart_window LOCATION:bedroom 1 ACTION:unlock_smart_window success [09/01/2015 19:24] ID:97 NAME:smart_window LOCATION:living room ACTION:unlock_smart_window success [09/01/2015 19:24] ID:139 NAME:smart_window LOCATION:bedroom 2 ACTION:unlock_smart_window success [09/01/2015 19:27] ID:123 NAME:smart_window LOCATION:bedroom 1 ACTION:open_smart_window success [09/01/2015 19:27] ID:97 NAME:smart_window LOCATION:living room ACTION:open_smart_window success [09/01/2015 19:27] ID:100 NAME:smart_window LOCATION:kitchen ACTION:open_smart_window success [09/01/2015 19:27] ID:139 NAME:smart_window LOCATION:bedroom 2 ACTION:open_smart_window success [09/01/2015 19:27] ID:116 NAME:smart_window LOCATION:bathroom ACTION:open_smart_window success

Fonte: Conjunto de *screenshots* obtido pela autora com base no simulador

Por fim, a última atividade desse PN (atividade (c): <<“Notificar alto nível de monóxido de carbono”, “Pessoas”>>) é do tipo multiobjeto com “Ação Atuação” referente a Pessoas (Figura 116).

Figura 116 – Cenário 4, atividade (c) a ser executada



Fonte: *Screenshots* obtido pela autora com base no controlador

Atualizando a instância da “Atividade” com, primeiramente, o objeto no paciente John, obteve-se 2 FuncDisp através das regras de inferência (Figura 117). Ainda, a instância da “Ação Notificar Alto Nível de Monóxido de Carbono” possui 4 instâncias de “Informação” (Figura 118): “Texto” com conteúdo do tipo *string* (Figura 119), “Imagem” com conteúdo do tipo

base64Binary (Figura 120), “Áudio” com conteúdo do tipo *string* (Figura 121), e “Vídeo” com conteúdo do tipo *string* (Figura 122).

Figura 117 – Cenário 4, resultado das regras de inferência para atividade (c) com foco no paciente

The screenshot shows an ontology viewer interface. On the left, under 'Ontologia', there are radio buttons for 'Classe', 'Instancia', and 'Propriedade'. The 'Instancia' radio button is selected. Below these are dropdown menus for 'Action' (set to 'Action'), 'rel_of_carbon_monoxide' (set to 'rel_of_carbon_monoxide'), and 'hasID' (set to 'hasID'). A 'Mostrar' button is below the dropdowns. The main area displays the properties of 'action_notifies_high_level_of_carbon_monoxide'. The properties listed are:

- type -> http://127.0.0.1/House_Ontology/house.owl#Action
- type -> http://127.0.0.1/House_Ontology/house.owl#Notifies
- type -> http://127.0.0.1/House_Ontology/house.owl#NotifiesHighLevelCarbonMonoxide
- type -> http://127.0.0.1/House_Ontology/house.owl#RegardingAPerson
- type -> <http://www.w3.org/2002/07/owl#Thing>
- type -> http://127.0.0.1/House_Ontology/house.owl#Acting
- sameAs -> http://127.0.0.1/House_Ontology/house.owl#action_notifies_high_level_of_carbon_monoxide
- differentFrom -> http://127.0.0.1/House_Ontology/house.owl#action_open
- differentFrom -> http://127.0.0.1/House_Ontology/house.owl#action_requests_agitation_assistance
- differentFrom -> http://127.0.0.1/House_Ontology/house.owl#action_unlock
- hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_high_level_of_carbon_monoxide_video
- hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_high_level_of_carbon_monoxide_sound
- hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_high_level_of_carbon_monoxide_image
- hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_high_level_of_carbon_monoxide_text
- isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_microwave_103
- isBetterExecutedBy -> http://127.0.0.1/House_Ontology/house.owl#func_play_sound_smart_refrigerator_105

Fonte: Screenshots obtido pela autora com base no controlador

Figura 118 – Cenário 4, instâncias de “Informação” contidas na instância de “Ação Notificar Alto Nível de Monóxido de Carbono”

The screenshot shows the same ontology viewer interface as Figure 117. The main area displays the properties of 'action_notifies_high_level_of_carbon_monoxide'. The properties listed are:

- type -> http://127.0.0.1/House_Ontology/house.owl#Action
- type -> http://127.0.0.1/House_Ontology/house.owl#Notifies
- type -> http://127.0.0.1/House_Ontology/house.owl#NotifiesHighLevelCarbonMonoxide
- type -> http://127.0.0.1/House_Ontology/house.owl#RegardingAPerson
- type -> <http://www.w3.org/2002/07/owl#Thing>
- type -> http://127.0.0.1/House_Ontology/house.owl#Acting
- sameAs -> http://127.0.0.1/House_Ontology/house.owl#action_notifies_high_level_of_carbon_monoxide
- differentFrom -> http://127.0.0.1/House_Ontology/house.owl#action_requests_agitation_assistance
- differentFrom -> http://127.0.0.1/House_Ontology/house.owl#action_unlock
- hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_high_level_of_carbon_monoxide_video
- hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_high_level_of_carbon_monoxide_sound
- hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_high_level_of_carbon_monoxide_image
- hasInformation -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_high_level_of_carbon_monoxide_text

Fonte: Screenshots obtido pela autora com base no controlador

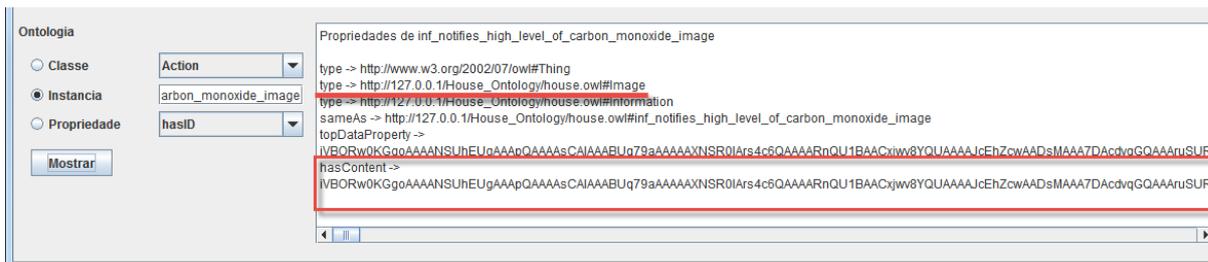
Figura 119 – Cenário 4, instância informação do tipo “Texto” com conteúdo do tipo *string*

The screenshot shows the ontology viewer interface. On the left, under 'Ontologia', the 'Instancia' radio button is selected. The dropdown menus are set to 'Action', 'f_carbon_monoxide_text', and 'hasID'. A 'Mostrar' button is below. The main area displays the properties of 'inf_notifies_high_level_of_carbon_monoxide_text'. The properties listed are:

- type -> http://127.0.0.1/House_Ontology/house.owl#Text
- type -> <http://www.w3.org/2002/07/owl#Thing>
- type -> http://127.0.0.1/House_Ontology/house.owl#Information
- sameAs -> http://127.0.0.1/House_Ontology/house.owl#inf_notifies_high_level_of_carbon_monoxide_text
- topDataProperty -> O ar está com alto nível de monóxido de carbono
- hasContent -> O ar está com alto nível de monóxido de carbono

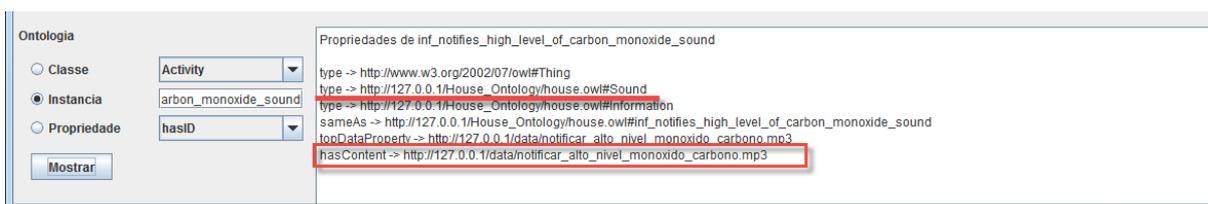
Fonte: Screenshots obtido pela autora com base no controlador

Figura 120 – Cenário 4, instância de “Informação” do tipo “Imagem” com conteúdo do tipo *base64Binary*



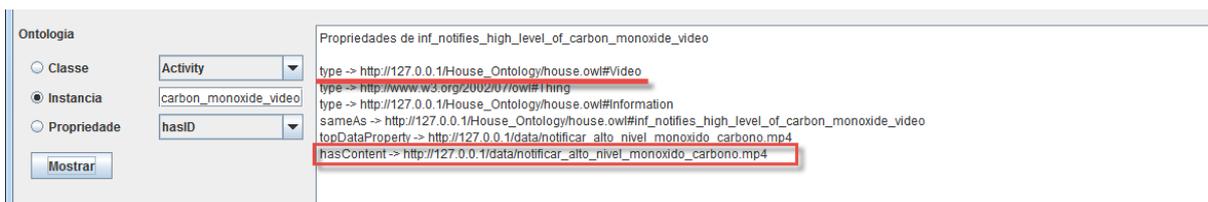
Fonte: Screenshots obtido pela autora com base no controlador

Figura 121 – Cenário 4, instância de “Informação” do tipo “Áudio” com conteúdo do tipo *string*



Fonte: Screenshots obtido pela autora com base no controlador

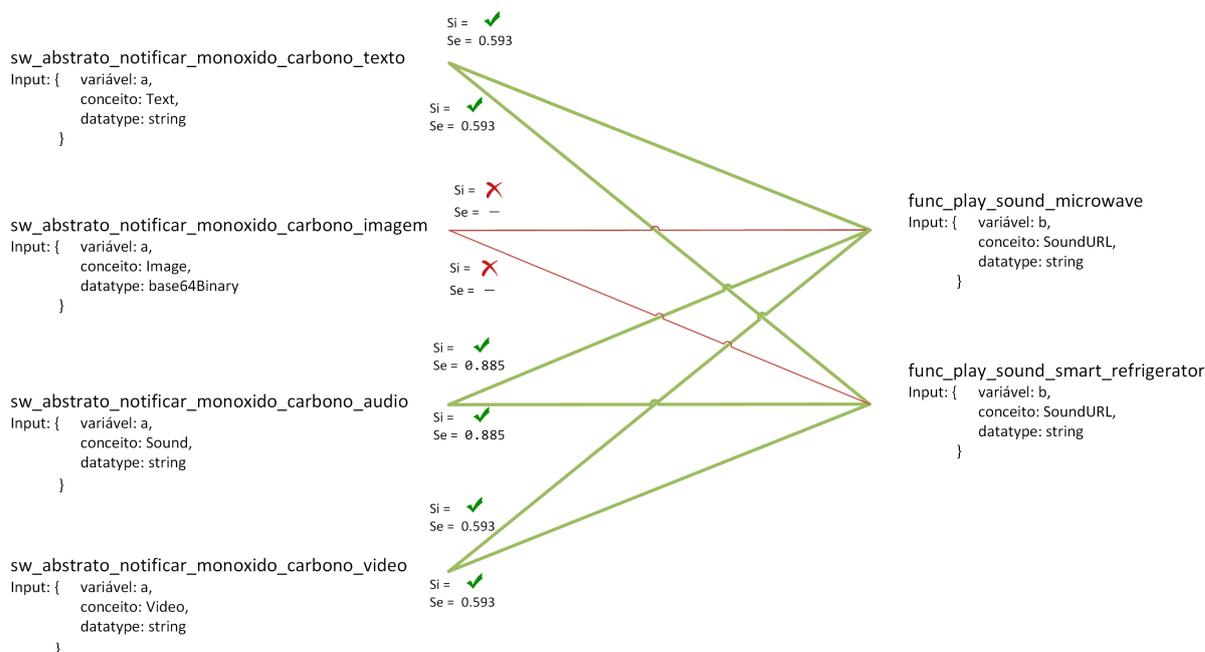
Figura 122 – Cenário 4, instância de “Informação” do tipo “Vídeo” com conteúdo do tipo *string*



Fonte: Screenshots obtido pela autora com base no controlador

Desta maneira, com 4 serviços Web abstratos criados a partir das instâncias de “Informação” contidas na instância de “Ação Notificar Alto Nível de Monóxido de Carbono”, e com as 2 FuncDisp obtidas pelas regras de inferências, forma-se 8 pares de combinações para a análise de similaridade sintática e semântica. Com a análise de similaridade sintática, obteve-se 6 pares com similaridade sintática (Figura 123). A partir desses pares é então feito a análise semântica, e os resultados são ordenados de forma decrescente e aplicado o limiar (Tabela 5.7).

Figura 123 – Cenário 4, resultado das análises de similaridade sintática e semântica para “Ação Notificar Alto Nível de Monóxido de Carbono” com foco no paciente



Fonte: imagem criada pela autora

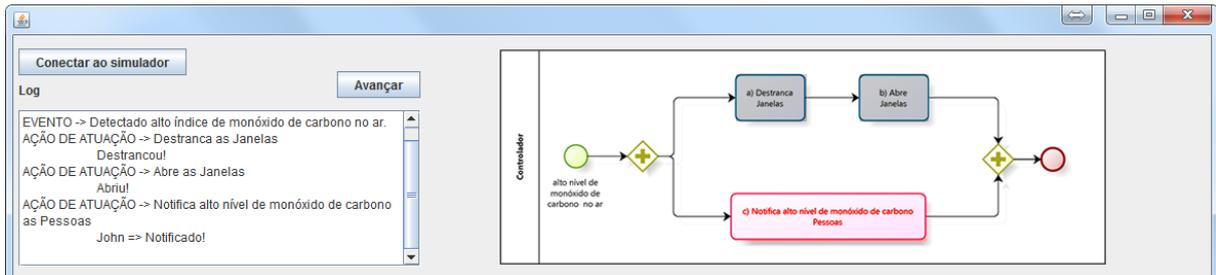
Tabela 5.7 – Cenário 4, ordenação dos pares para a atividade (c) com foco no paciente através da similaridade semântica

<i>SW abstrato</i>	<i>FuncDisp</i>	<i>Se</i>
sw_abstrato_notificar_monoxido_carbono_audio	play_sound_smart_refrigerator_105	0,886
sw_abstrato_notificar_monoxido_carbono_audio	play_sound_microwave_103	0,886
sw_abstrato_notificar_monoxido_carbono_video	play_sound_smart_refrigerator_105	0,595
sw_abstrato_notificar_monoxido_carbono_video	play_sound_microwave_103	0,595
sw_abstrato_notificar_monoxido_carbono_texto	play_sound_smart_refrigerator_105	0,558
sw_abstrato_notificar_monoxido_carbono_texto	play_sound_microwave_103	0,558

Fonte: tabela criada pela autora

Com isso, o primeiro FuncDisp (*play_sound_smart_refrigerator_105*) da ordenação é verificado suas precondições (ter o estado ativo = *true* e ocupado = *false*). Visto que possui as precondições satisfeitas, é acionado no simulador (Figura 105) e a atividade é parcialmente executada no controlador (Figura 124), faltando ainda executar a ação com foco na cuidadora Eve.

Figura 124 – Cenário 4, atividade (c) com foco no paciente executada pelo controlador



Fonte: Screenshots obtido pela autora com base no controlador

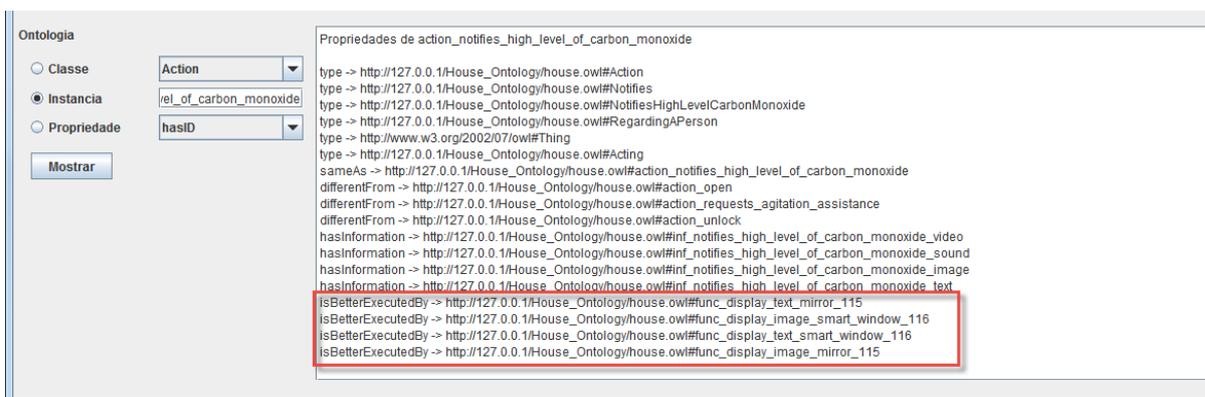
Figura 125 – Cenário 4, atividade (c) com foco no paciente executada no simulador

Fonte: Screenshots obtido pela autora com base no simulador

Atualizando a instância da “Atividade” com objeto focado na cuidadora Eve, obteve-se 4 FuncDisp através das regras de inferência (Figura 126). Com as instâncias de “Informação”

contidas na “Ação Notificar Alto Nível de Monóxido de Carbono”, formam-se 16 pares de combinações (por motivos de visibilidade, não serão mostrados os resultados da análise de similaridade sintática). Dos pares que possuem similaridade sintática (12 pares com similaridade sintática) é, então, realizada a similaridade semântica, e o resultado é ordenado de forma decrescente e aplicado o limiar (Tabela 5.8).

Figura 126 – Cenário 4, resultado das regras de inferência para atividade (c) com foco na cuidadora



Fonte: Screenshots obtido pela autora com base no controlador

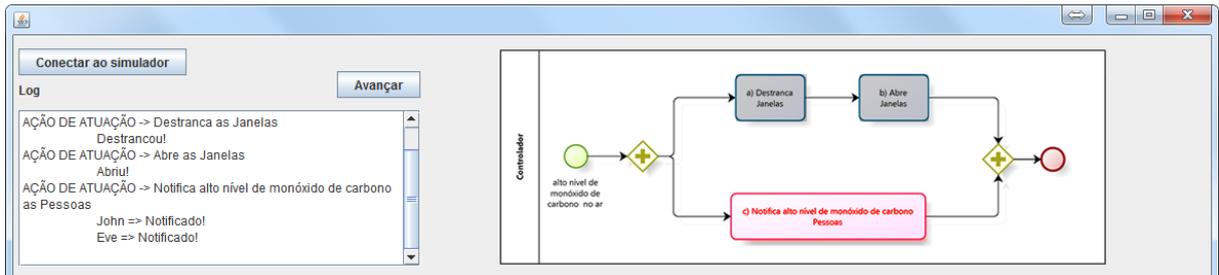
Tabela 5.8 – Cenário 4, ordenação dos pares para a atividade (c) com foco na cuidadora através da similaridade semântica

<i>SW abstrato</i>	<i>FuncDisp</i>	<i>Se</i>
sw_abstrato_notificar_monoxido_carbono_texto	show_text_smart_window_116	1,000
sw_abstrato_notificar_monoxido_carbono_texto	show_text_mirror_115	1,000
sw_abstrato_notificar_monoxido_carbono_audio	show_text_smart_window_116	0,625
sw_abstrato_notificar_monoxido_carbono_audio	show_text_mirror_115	0,625
sw_abstrato_notificar_monoxido_carbono_video	show_text_smart_window_116	0,625
sw_abstrato_notificar_monoxido_carbono_video	show_text_mirror_115	0,625
sw_abstrato_notificar_monoxido_carbono_audio	show_image_mirror_115	0,595
sw_abstrato_notificar_monoxido_carbono_audio	show_image_smart_window_116	0,595
sw_abstrato_notificar_monoxido_carbono_video	show_image_mirror_115	0,595
sw_abstrato_notificar_monoxido_carbono_video	show_image_smart_window_116	0,595
sw_abstrato_notificar_monoxido_carbono_texto	show_image_mirror_115	0,558
sw_abstrato_notificar_monoxido_carbono_texto	show_image_smart_window_116	0,558

Fonte: tabela criada pela autora

Com isso, a primeira FuncDisp (*show_text_smart_window_116*) é verificada suas condições (ter os estados ligado = *true* e ocupado = *false*). Estando as condições satisfeitas, é acionada no simulador a FuncDisp *show_text_smart_window_116* (Figura 127), e a atividade (c) é executada complementamente no controlador (Figura 127), e por fim, finaliza-se o esse modelo PN (Figura 129).

Figura 127 – Cenário 4, atividade (c) com foco na cuidadora executada pelo controlador

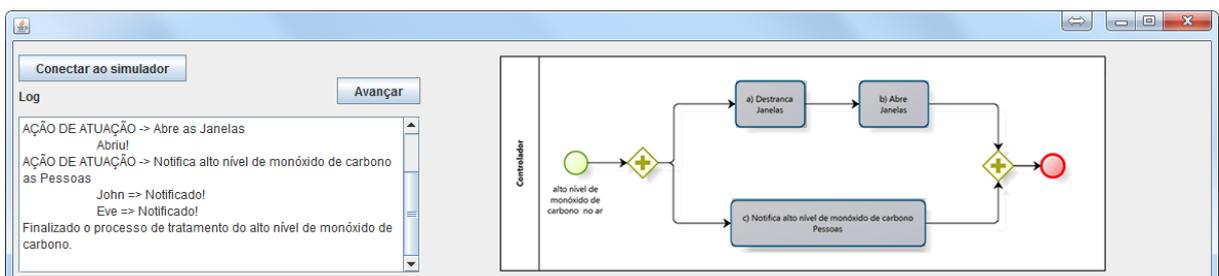


Fonte: *Screenshots* obtido pela autora com base no controlador

Figura 128 – Cenário 4, atividade (c) com foco na cuidadora executada no simulador

Fonte: *Screenshots* obtido pela autora com base no simulador

Figura 129 – Cenário 4, fim do PN



Fonte: *Screenshots* obtido pela autora com base no controlador

5.3 Análise da Abordagem

Nas seções anteriores, Seção 5.1 e Seção 5.2, foi mostrado uma implementação do framework proposto e a execução detalhada de quatro cenário de aplicação respectivamente. Com isso, espera-se demonstrar a aplicabilidade e validação do *framework*.

Retomando, os trabalhos relacionados na Seção 2.2.4.1 e Seção 2.4.1, a abordagem proposta neste trabalho apresenta-se nos quadros (Quadro 5.1 e Quadro 5.2):

Quadro 5.1 – Comparativo entre a abordagem proposta e os trabalhos relacionados de *matching* de SW baseado em pragmático

Trabalho	Matching de SW			Contexto		Motor de Inferência	Usa contexto para
	P	Si	Se	Modelagem	Implementação		
Broens et al. (2004)	-	-	X	ontologia	própria	-	parâmetros
MAAMAR et al. (2005)	-	-	-	-	própria	-	solicitação
Kuck e Gnasa (2007)	X	-	-	-	própria	-	solicitação
Suraci et al. (2007)	X	-	X	ontologia	própria	-	solicitação
Mokhtar et al (2008)	-	-	X	ontologia	OWL	-	solicitação
Sheng et al. (2009)	-	-	-	-	própria	-	solicitação
Sherchan et al (2012)	-	-	X	ontologia	própria	-	solicitação
Han et al. (2014)	-	-	-	ontologia	OWL	X	solicitação/ parâmetro
Abordagem proposta	-	X	X	ontologia	OWL	X	solicitação/ parâmetro

Fonte: quadro criado pela autora

O Quadro 5.1 compara a abordagem proposta neste trabalho com os trabalhos relacionados da área de *matching* de SW baseado em pragmático. É possível ver, nesse quadro, quais os tipos de *matching* de SW que foi utilizado (sintático e semântico), como foi modelado o contexto (através de ontologia), e como o contexto foi implementado (OWL), que utilizou-se motor de inferência (na pré-seleção das funcionalidades de dispositivos), e que o contexto foi empregado na solicitação (filtrando as das funcionalidades de dispositivos) e como parâmetro (na montagem dos serviços Web abstratos).

Quadro 5.2 – Comparativo entre a abordagem proposta e os trabalhos relacionados de coordenação de serviços em *Smart Home* e *Home Care*

Trabalho	Contexto		Usa Motor de Inferência	Forma de coord.	Sel. din. o SW	Utiliza alg. de matching de SW
	Mod.	Imp.				
Gu et al. (2005)	ontologia	OWL	X	regras	-	-
Ardissono et al. (2006)	-	própria	-	PN / regras	-	-
Cho et al. (2007)	NI	RDF	-	PN / regras	NI	-
Wang e Turner (2008)	-	-	-	regras	X	-
XU et al. (2009)	ontologia	NI	X	regras	X	-
Lamine et al. (2010)	-	-	-	PN	NI	-
Valiente-Rocha e Lozano-Tello (2010)	ontologia	OWL	X	regras	NI	-
Paganelli e Giuli (2011)	ontologia	OWL	X	regras	NI	-
Preuveneers e Berbers (2012)	ontologia	OWL	X	regras	X	-
Kao e Yuan (2012)	ontologia	OWL	X	PN / regras	-	-
Kaldeli et al. (2013)	-	própria	-	regras	X	-
Han et al. (2014)	ontologia	OWL	X	regras	X	-
Abordagem proposta	ontologia	OWL	X	PN / regras	X	X

Fonte: quadro criado pela autora

O Quadro 5.2 compara o *framework* proposto com os demais trabalhos relacionados na área de coordenação de serviços em *Smart Home* e *Home Care*. Como se observa nesse quadro, o *framework* proposto modela o contexto utilizando ontologia, implementa o contexto por OWL, utiliza motor de inferência, coordena os serviços por meio de modelo de PN e por regras, seleciona dinamicamente SW e utiliza algoritmos de *matching* de SW.

5.4 Resumo do Capítulo 5

O Capítulo 5 mostrou uma implementação do *framework*. Para a implementação foi necessário: implementar o modelo para aplicações *Home Care* (proposto na Seção 3.2.1); especificar um simulador de casa controlável, ao qual foi desenvolvido por Pötter (2013); especificar um conjunto de descrições semânticas para as funcionalidades de dispositivos representadas por SW, que foram manualmente criadas pelo bolsista de iniciação científica

Paulo Renato Lanzarin; e construir um controlador que se comunicasse com o simulador de casa controlável desenvolvido e executasse os cenários de aplicação descritos na Seção 4.

Além disso, foi demonstrado a execução dos quatro cenários de aplicação, detalhando o funcionamento das onze atividades com ação de atuação. Desta forma, exemplificando como ocorre a instanciação das atividades de atuação de modelo de PN para diferentes combinações de triplas de rótulo de atividade.

Ao final, a abordagem proposta é comparada com os trabalhos relacionados apresentados na Seção 2.2.4.1 e Seção 2.4.1.

6 CONCLUSÕES

Este trabalho apresentou uma nova abordagem para tratar do problema de instanciação de atividades de modelo de PN, em ambiente de *Home Care*, através da seleção de funcionalidade de dispositivos domésticos, representados por serviços Web, de maneira sensível ao contexto. Para resolver esse problema, foi proposto um modelo ontológico de base para descrever os principais conceitos do domínio e seus relacionamentos, e uma estratégia de rotulação de atividades de modelo de PN.

Na abordagem proposta, orientações de como gerenciar situações de interesse são modeladas através de modelo de PN no nível CIM. Quando o modelador está especificando o modelo de PN, ele seleciona conceitos correspondentes no modelo ontológico para preencher os rótulos das atividades do modelo de PN. Depois, quando o PN está executando uma ação de atuação, o controlador raciocina através do motor de inferência quais funcionalidades de dispositivos que podem executar tal determinada ação levando em consideração as informações de contexto. Com base nisso, filtra-se as funcionalidades de dispositivo (representadas por SW) utilizando técnicas de *matching* de SW, para garantir que se consiga acionar o serviço efetivamente (passar corretamente os argumentos de entrada quando for necessário), e constrói uma lista ordenada das possíveis funcionalidades de dispositivos que podem executar a ação de atuação. Ainda, é verificado se as condições da funcionalidade de dispositivo estão satisfeitas, para assim, poder acioná-lo, ou passando para a próxima. Desta forma, instanciando a atividade de modelo de PN.

Para avaliar a abordagem proposta, foi utilizado um simulador de casa automatizada, onde foi possível verificar o funcionamento da abordagem em 4 diferentes cenários baseados na literatura.

As principais contribuições deste trabalho são:

- O modelo ontológico base para Home Care;
- O framework para instanciação dinâmica de funcionalidade de dispositivos domésticos para executar ações de atuação levando em consideração aspectos semânticos, sintáticos e contextuais.

Tais contribuições permitem que o modelo de PN permaneça no nível CIM, permitindo que o modelador seja alguém do domínio de *Home Care*, sem precisar se preocupar com questões de implementação. Em adicional, esta abordagem permite que o sistema seja adaptativo ao usuário, e a casa possa incorporar novos dispositivos e ainda gerenciá-los de acordo com a necessidade.

As contribuições secundárias deste trabalho são:

- Artigo publicado no ICEIS:
 - Kambara-Silva, J. K. ; Machado, G. M.; Thom, L. H.; Wives, L. K. (2014). “Business Process Modeling and Instantiation in Home Care Environments”. In: 16th International Conference on Enterprise Information Systems.
- Coorientação do trabalho de graduação do Eduardo Gischkow Pötter (2013);
- Coorientação da bolsa de iniciação científica do bolsista Paulo Renato Lanzarin.

Uma limitação deste trabalho é que o modelo ontológico precisa ser povoado com todas as ações possíveis do controlador e as funcionalidades dos dispositivos para permitir que o modelador consiga vincular os conceitos ontológicos relacionados ao modelo de PN que está modelando. Outra limitação é que para as ações referentes a pessoa deveria ser considerado o alcance do dispositivo (por exemplo, uma televisão de 42” tem maior alcance visual do que um celular) e um grau de distância mais preciso do dispositivo em relação a pessoa alvo, ao invés de apenas considerar o cômodo da casa.

Como trabalho futuro, pretende-se focar nas ações do tipo sensoriamento, de modo que o controlador também possa dinamicamente executá-las através de dispositivos domésticos e que o modelo de PN ainda continue no nível CIM. Entretanto, esse não é um problema trivial, pois durante a execução do modelo de PN, caminhos são decididos de acordo com ações de sensoriamento. Logo, além de necessitar de uma estruturação dos conceitos ontológicos para o tipo ação sensoriamento, e verificar se com a mesma abordagem de instanciação de atividades de atuação consegue-se instanciar as atividades de sensoriamento, teria que se pensar também em uma estratégia de como implementar os desvios presentes em um modelo de PN de maneira automática, mas que não retire o modelo de PN do nível CIM.

Ainda, quer-se verificar como a abordagem se comporta para criação de serviço Web abstrato com mais de um parâmetro de entrada. Visto que, a execução dos cenários de aplicação não apresentou serviços Web com mais de um argumento de entrada.

REFERÊNCIAS

- ARDISSONO, L. et al. Adaptive Medical Workflow Management for a Context-Dependent Home Healthcare Assistance Service. **Electronic Notes in Theoretical Computer Science**, v. 146, n. 1, p. 59–68, jan. 2006.
- ARIZA, J.; GARCÍA, C. U.; HERNÁNDEZ, F. S. Heterogeneous Device Networking for an AmI Environment. In: INTERNATIONAL CONFERENCE UBIQUITOUS COMPUTING AND AMBIENT INTELLIGENCE, 7, 2013. **Proceedings...**[S.l.]: Springer, 2013. p.183–190.
- ATKINSON, C. et al. A Practical Approach to Web Service Discovery and Retrieval. In: INTERNATIONAL CONFERENCE ON WEB SERVICES (ICWS 2007), jul. 2007. **Proceedings...**[S.l.]: IEEE, 2007, p.241–248.
- AUGUSTO, J. C.; LIU, J.; CHEN, L. Using Ambient Intelligence For Disaster Management. In: KNOWLEDGE-BASED INTELLIGENT INFORMATION AND ENGINEERING SYSTEMS, 2006. **Proceedings...** [S.l.]: Springer Berlin Heidelberg, 2006, p.171–178.
- AUVINEN, A. et al. Defining Service Elements in Home Care. In: ANNUAL SRII GLOBAL CONFERENCE, 2011. **Proceedings...** [S.l.]: IEEE, 2011, p.378–383.
- BALADRÓN, C. et al. User-Driven Context Aware Creation And Execution Of Home Care Applications. **Annals of Telecommunications**, v. 65, n. 9-10, p. 545–556, maio 2010.
- BASTIDE, R.; ZEFOUNI, S.; LAMINE, E. The Homecare Digital Ecosystem: An Information System Support Architecture. In: INTERNATIONAL CONFERENCE ON DIGITAL ECOSYSTEMS AND TECHNOLOGIES, 4, 2010. **Proceedings...** [S.l.]: IEEE, 2010, p.475–480.
- BIRON, P. V.; MALHOTRA, A. **XML Schema Part 2: Datatypes Second Edition**. Disponível em: <<http://www.w3.org/TR/xmlschema-2/>>.
- BONINO, D.; CORNO, F. Dogont-Ontology Modeling for Intelligent Domotic Environments. In: THE SEMANTIC WEB-ISWC, 7, 2008. **Proceedings...** [S.l.]: Springer Berlin Heidelberg, 2008, p.790–803.
- BROENS, T. et al. Context-Aware, Ontology-Based Service Discovery. In: EUROPEAN SYMPOSIUM ON AMBIENT INTELLIGENCE, 2, 2008. **Proceedings...** [S.l.]: Springer Berlin Heidelberg, 2008, p.72–83.
- CHAN, M. et al. A Review Of Smart Homes - Present State and Future Challenges. **Computer Methods and Programs in Biomedicine**, v. 91, n. 1, p. 55–81, jul. 2008.
- CHINOSI, M.; TROMBETTA, A. BPMN: An introduction to the standard. **Computer Standards & Interfaces**, v. 34, n. 1, p. 124–134, jan. 2012.
- CHO, Y.; CHOI, J.; CHOI, J. A Context-Aware Workflow System for a Smart Home. In: INTERNATIONAL CONFERENCE ON CONVERGENCE INFORMATION TECHNOLOGY (ICCIT 2007), 2007. **Proceedings...** [S.l.]: IEEE, 2007, p.95–100.
- COOK, D. J.; AUGUSTO, J. C.; JAKKULA, V. R. Ambient intelligence: Technologies, applications, and opportunities. **Pervasive and Mobile Computing**, v. 5, n. 4, p. 277–298, 2009.
- CROW, B. K. L. Four Types of Disabilities : Their Impact on Online Learning. **TechTrends**, v.52, n. 1, p. 51–55, 2008.

- DEY, A. K. Understanding and Using Context. **Personal and Ubiquitous Computing**, v. 5, n. 1, p. 4–7, 28 fev. 2001.
- DONG, X. et al. Similarity Search for Web Services. In: VLDB CONFERENCE, 2004. **Proceedings...** St Louis: Morgan Kaufmann, 2004, p. 372 – 383.
- FABRA, J. et al. Automatic execution of business process models: Exploiting the benefits of Model-driven Engineering approaches. **Journal of Systems and Software**, v. 85, n. 3, p. 607–625, 2012.
- FENSEL, D. et al. **Enabling Semantic Web Services**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. v. 1
- FERNANDEZ, A.; POLLERES, A.; OSSOWSKI, S. Towards Fine-grained Service Matchmaking by Using Concept Similarity. In: WORKSHOP ON SERVICE MATCHMAKING AND RESOURCE RETRIEVAL IN THE SEMANTIC WEB, 1, 2007. **Proceedings...**[S.l: s.n], 2007, p. 31–45.
- FETHALLAH, H.; AMINE, C.; AMINE, B. Automated Discovery Of Web Services. In: INTERNATIONAL CONFERENCE ON INTELLIGENT SEMANTIC WEB-SERVICES AND APPLICATIONS (ISWSA), 1, 2010. **Proceedings...** New York, USA: ACM Press, 2010, p. 915–922.
- GANJISAFFAR, Y. et al. A Similarity Measure for OWL-S Annotated Web Services. In: INTERNATIONAL CONFERENCE ON WEB INTELLIGENCE (WI), dez. 2007. **Proceedings...**Hong Kong: IEEE, 2007, p. 621–624.
- GAO, X.; YANG, J.; PAPAZOGLU, M. P. The Capability Matching of Web Services. In: INTERNATIONAL SYMPOSIUM ON MULTIMEDIA SOFTWARE ENGINEERING, 4, 2002. **Proceedings...** [S.l.]: IEEE, 2002, p. 56–63.
- GASSEN, J. B. et al. Ontology Support for Home Care Process Design. In: INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS, 14, 2012. **Proceedings...** [S.l.]: SciTePress, 2012, p. 84–89.
- GRUBER, T. R. A Translation Approach to Portable Ontology Specifications. **Knowledge Acquisition**, v. 5, n. 2, p. 199–220, jun. 1993.
- GU, T.; PUNG, H. K.; ZHANG, D. Q. A service-oriented middleware for building context-aware services. **Journal of Network and Computer Applications**, v. 28, n. 1, p. 1–18, jan. 2005.
- GUO, R.; CHEN, D.; LE, J. Matching semantic web services across heterogeneous ontologies. In: INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY (CIT), 5, 2005. **Proceedings...** [S.l.]: IEEE, 2005, p. 264–268.
- HAN, S. N.; LEE, G. M.; CRESPI, N. Semantic Context-Aware Service Composition for Building Automation System. **IEEE Transactions on Industrial Informatics**, v. 10, n. 1, p. 752–761, fev. 2014.
- HAO, Y.; ZHANG, Y. Web Services Discovery Based on Schema Matching. In: AUSTRALASIAN CONFERENCE ON COMPUTER SCIENCE, 30, 2004. **Proceedings...** Darlinghurst, Australia: Australian Computer Society, Inc., 2004, p. 107–113.
- JIANG, J. J.; CONRATH, D. W. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: INTERNATIONAL CONFERENCE ON RESEARCH IN COMPUTATIONAL LINGUISTICS, 10, 1977. **Proceedings...** Taiwan: [s.n.], 1977, p.1–15.

- KALDELI, E. et al. Coordinating the web of services for a smart home. **ACM Transactions on the Web**, v. 7, n. 2, p. 10:1–10:40, 1 maio 2013.
- KALIBATIENE, D.; VASILECAS, O. Ontology axioms for the implementation of business rules. **Technological and Economic Development of Economy**, v. 16, n. 3, p. 471–486, 2010.
- KALIBATIENE, D.; VASILECAS, O. **Survey on Ontology Languages**. In: INTERNATIONAL CONFERENCE ON PERSPECTIVES IN BUSINESS INFORMATICS RESEARCH, 2011. **Proceedings...** [S.l.]: Springer Berlin Heidelberg, 2011, p.124–141.
- KAMBARA-SILVA, J. K. et al. Business Process Modeling and Instantiation in Home Care Environments. In: INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS, 16, 2014. **Proceedings...** [S.l.]: Scitepress, 2014, p.513–525.
- KAO, Y.-W.; YUAN, S.-M. User-configurable semantic home automation. **Computer Standards & Interfaces**, v. 34, n. 1, p. 171–188, jan. 2012.
- KJELDSKOV, J.; SKOV, M. B. Supporting Work Activities in Healthcare by Mobile Electronic Patient Records. In: ASIA PACIFIC CONFERENCE, 6, 2004. **Proceedings...** Rotorua, New Zealand: Springer Berlin Heidelberg, 2004, p.191–200.
- KLUSCH, M.; KAPAHNKE, P. Adaptive signature-based semantic selection of services with OWLS-MX3. **Multiagent and Grid Systems**, v. 8, n.1, p. 69–82, 2012a.
- KLUSCH, M.; KAPAHNKE, P. The iSeM matchmaker: A flexible approach for adaptive hybrid semantic service selection. **Web Semantics: Science, Services and Agents on the World Wide Web**, v. 15, p.1–14, set. 2012b.
- KUCK, J.; GNASA, M. Context-Sensitive Service Discovery Meets Information Retrieval. In: INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS (PerComW), 5, 2007. **Proceedings...**[S.l.]: IEEE, 2007, p.601–605.
- LAMINE, E. et al. A System Architecture Supporting the Agile Coordination of Homecare Services. In: WORKING CONFERENCE ON VIRTUAL ENTERPRISES, 11, 2010. **Proceedings...** St. Etienne, France: Springer Berlin Heidelberg, 2010, p. 227–234.
- LAMPE, U.; SCHULTE, S. Self-Adaptive Semantic Matchmaking Using COV4SWS.KOM and LOG4SWS.KOM. In: BLAKE, B. et al. (Eds.). **Semantic Web Services**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 141–157.
- LEACOCK, C.; CHODOROW, M. Combining Local Context and WordNet Similarity for Word Sense Identification. In: **WordNet: An electronic lexical database**. [S.l.: s.n.]. p. 265–283.
- LI, Y.; BANDAR, Z. A.; MCLEAN, D. An approach for measuring semantic similarity between words using multiple information sources. **IEEE Transactions on Knowledge and Data Engineering**, v. 15, n. 4, p. 871–882, jul. 2003.
- LIN, D. An Information-Theoretic Definition of Similarity. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 15, 1998. **Proceedings...** San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, p. 296–304.
- LIU, F. et al. Measuring Similarity of Web Services Based on WSDL. In: INTERNATIONAL CONFERENCE ON WEB SERVICES (ICWS), 2010. **Proceedings...** [S.l.]: IEEE, 2010, p.155–162.
- LIU, M. et al. An weighted ontology-based semantic similarity algorithm for web service. **Expert Systems with Applications**, v. 36, n. 10, p. 12480–12490, dez. 2009.

- MAAMAR, Z.; MOSTEFAOUI, S. K.; MAHMOUD, Q. H. Context for Personalized Web Services. In: ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 38, 2005. **Proceedings...** [S.l.]: IEEE, 2005, p.166–175.
- MACHADO, A. et al. Situation-awareness as a Key for Proactive Actions in Ambient Assisted Living. In: INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS, 15, 2013. **Proceedings...** [S.l.]: SciTePress, 2013, p.418–426.
- MACHADO, A. et al. A Reactive and Proactive Approach for Ambient Intelligence. In: International Conference on Enterprise Information Systems, 16, 2014. **Proceedings...** [S.l.]: SCITEPRESS, 2014, p.501–512.
- MACHADO, G. M.; DE OLIVEIRA, J. P. M. Context-Aware Adaptive Recommendation of Resources for Mobile Users in a University Campus. In: INTERNATIONAL CONFERENCE ON WIRELESS AND MOBILE COMPUTING, NETWORKING AND COMMUNICATIONS (WiMob), 10, 2014. **Proceedings...** Larnaca: IEEE, 2014, p.427–433.
- MANZORR, S.; CEUSTERS, W.; RUDNICKI, R. Implementation of a Referent Tracking System. **International Journal of Healthcare Information Systems and Informatics**, v. 2, n. 4, p. 41–58, 2007.
- MARTIN, D. et al. **OWL-S: Semantic Markup for Web Services**. Disponível em: <<http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/>>.
- MASUCH, N. et al. SeMa2: A Hybrid Semantic Service Matching Approach. In: **Semantic Web Services**. [s.l.: s.n.].
- MAZANEK, S.; HANUS, M. Constructing a bidirectional transformation between BPMN and BPEL with a functional logic programming language. **Journal of Visual Languages and Computing**, v. 22, n. 1, p. 66–89, 2011.
- MCGEE-LENNON, M. R. Requirements engineering for home care technology. In: ANNUAL CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS (CHI), 26, 2008. **Proceedings...** New York, USA: ACM Press, 2008, p.1439–1442.
- MCGUINNESS, D. L.; HARMELEN, F. VAN. **OWL Web Ontology Language Overview**. Disponível em: <<http://www.w3.org/TR/owl-features/>>.
- MOHEBBI, K. et al. UltiMatch-NL: A Web Service Matchmaker Based on Multiple Semantic Filters. **PLoS ONE**, v. 9, n. 8, 2014.
- MOKHTAR, S. BEN et al. EASY: Efficient semAntic Service discoverY in pervasive computing environments with QoS and context support. **Journal of Systems and Software**, v. 81, n. 5, p. 785–808, maio 2008.
- OTERO-CERDEIRA, L.; RODRÍGUEZ-MARTÍNEZ, F. J.; GÓMEZ-RODRÍGUEZ, A. Ontology matching: A literature review. **Expert Systems with Applications**, v. 42, n. 2, p. 949–971, fev. 2015.
- OUYANG, C. et al. From BPMN process models to BPEL Web services. In: INTERNATIONAL CONFERENCE ON WEB SERVICES PROCEEDINGS (ICWS), 2006. **Proceedings...**[S.l.]: IEEE, 2006, p.285–292.
- OUYANG, C. et al. From business process models to process-oriented software systems: The BPMN to BPEL way. **ACM Transactions on Software Engineering and Methodology**, v. 19, p. 1–37, 2009.
- PAGANELLI, F.; GIULI, D. An ontology-based system for context-aware and configurable services to support home-based continuous care. **IEEE Transactions on Information**

Technology In Biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society, v. 15, n. 2, p. 324–33, mar. 2011.

PAOLUCCI, M. et al. **Semantic Matching of Web Services Capabilities** In: INTERNATIONAL SEMANTIC WEB CONFERENCE ON THE SEMANTIC WEB, 1, 2002. **Proceedings...** London, UK: Springer-Verlag, 2002, p.333–347.

PARK, S. H. et al. Smart home ? digitally engineered domestic life. **Personal and Ubiquitous Computing**, v. 7, n. 3-4, p. 189–196, 1 jul. 2003.

PATWARDHAN, S.; PEDERSEN, T. Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. In: WORKSHOP MAKING SENSE OF SENSE - BRINGING COMPUTATIONAL LINGUISTICS AND PSYCHOLINGUISTICS TOGETHER, 2006. **Proceedings...** Trento, Italy: [s.n.], 2006, p. 1–8.

PERNAS, A. M. et al. Enriching adaptation in e-learning systems through a situation-aware ontology network. **Interactive Technology and Smart Education**, v. 9, p. 60–73, 2012.

PETRAKIS, E. G. M. et al. X-Similarity : Computing Semantic Similarity between Concepts from Different Ontologies object instrumentality. **Journal of Digital Information Management**, 2006.

PLEBANI, P.; PERNICI, B. URBE: Web Service Retrieval Based on Similarity Evaluation. **IEEE Transactions on Knowledge and Data Engineering**, v. 21, n. 11, p. 1629–1642, nov. 2009.

PLEBANI, P.; PERNICI, B. Semantic Annotations and Web Service Retrieval : The URBE Approach. In: **Semantic Web Services**. [s.l: s.n.].

PÖTTER, E. G. **Simulador de Ambientes Inteligentes : foco em home care para pessoas com idade avançada**. 2013. f.50. Trabalho de Graduação.[s.l.] Instituto de Informática, UFRGS, Porto Alegre, 2013.

PREUVENEERS, D.; BERBERS, Y. Intelligent Widgets for Intuitive Interaction and Coordination in Smart Home Environments. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ENVIRONMENTS, 8, 2012. **Proceedings...** [S.l.]: IEEE, 2012, p.157–164.

PUNG, H. et al. Context-aware middleware for pervasive elderly homecare. **IEEE Journal on Selected Areas in Communications**, v. 27, n. 4, p. 510–524, maio 2009.

RADA, R. et al. Development and application of a metric on semantic nets. **IEEE Transactions on Systems, Man, and Cybernetics**, v.19, n.1, p.17–30, 1989.

RESNIK, P. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 14, 1995. **Proceedings...** San Francisco, CA, USA: [s.n.], 1995, p.448–453.

RODRÍGUEZ, M. A.; EGENHOFER, M. J. Determining semantic similarity among entity classes from different ontologies. **IEEE Transactions on Knowledge and Data Engineering**, v. 15, n. 2, p. 442–456, 2003.

RONG, W. R. W.; LIU, K. L. K. A Survey of Context Aware Web Service Discovery: From User's Perspective. In: INTERNATIONAL SYMPOSIUM ON SERVICE ORIENTED SYSTEM ENGINEERING (SOSE), 5, 2010. **Proceedings...** [S.l.]: IEEE, 2010, p.15–22.

SÁNCHEZ, D. et al. Ontology-based semantic similarity: A new feature-based approach. **Expert Systems with Applications**, v. 39, n. 9, p. 7718–7728, jul. 2012.

- SANTOS, P. B.; WIVES, L. K.; DE OLIVEIRA, J. P. M. An Improved Approach for Measuring Similarity among Semantic Web Services. In: INTERNATIONAL CONFERENCE ON WEB INFORMATION SYSTEMS AND TECHNOLOGIES, 8, 2012. **Proceedings...** [S.l.: s.n.], 2012, p.1–6.
- SCHULTE, S. et al. COV4SWS.KOM: Information quality-aware matchmaking for semantic services. In: EXTENDED SEMANTIC WEB CONFERENCE (ESWC), 9, 2012. **Proceedings...** [S.l.]: Springer Berlin Heidelberg, 2012, p. 499–513.
- SECO, N.; VEALE, T.; HAYES, J. An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 16, 2004. **Proceedings...** [S.l.:s.n.], 2004, p.1089–1090.
- SEDDIQUI, H.; AONO, M. Metric of Intrinsic Information Content for Measuring Semantic Similarity in an Ontology. In: ASIA-PACIFIC CONFERENCE ON CONCEPTUAL MODELLING, 7, 2010. **Proceedings...** Darlinghurst, Australia: Australian Computer Society, 2010, p.89–96.
- SHENG, Q. Z. et al. ContextServ: A platform for rapid and flexible development of context-aware Web services. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 1, 2009. **Proceedings...** [S.l.]: IEEE, 2009. p. 619–622.
- SHENG, Q. Z.; BENATALLAH, B. ContextUML: a UML-based modeling language for model-driven development of context-aware Web services. In: INTERNATIONAL CONFERENCE ON MOBILE BUSINESS (ICMB), 2005. **Proceedings...** [S.l.:s.n.], 2005, p.206–212.
- SHERCHAN, W. et al. Context-sensitive user interfaces for semantic services. **ACM Transactions on Internet Technology**, v. 11, n. 3, p. 14:1–14:27, 1 jan. 2012.
- SIVASHANMUGAM, K. et al. Adding Semantics to Web Services Standards. In: INTERNATIONAL CONFERENCE ON WEB SERVICES (ICWS), 2003. **Proceedings...** Las Vegas, NV: CSREA Press, 2003, p.23–31.
- STROULIA, E. Flexible Interface Matching for Web-service Discovery. In: INTERNATIONAL CONFERENCE ON PROPERTIES AND APPLICATIONS OF DIELECTRIC MATERIALS, 7, 2003. **Proceedings...** [S.l.]: IEEE, 2003, p. 147–156.
- STROULIA, E.; WANG, Y. Structural and Semantic Matching for Assessing Web-service Similarity. **International Journal of Cooperative Information Systems**, v. 14, n. 04, p. 407–437, dez. 2005.
- SU, Z.; WANG, X. An improved approach to rapid discovery of semantic Web service. In: INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING AND APPLICATIONS, 5, 2010. **Proceedings...** [S.l.]: IEEE, 2010, p.378–381.
- SURACI, V.; MIGNANTI, S.; AIUTO, A. Context-aware Semantic Service Discovery. In: IST MOBILE AND WIRELESS COMMUNICATIONS SUMMIT, 16, 2007. **Proceedings...** [S.l.]: IEEE, 2007, p.1–5.
- TRUONG, H.-L.; DUSTDAR, S. A survey on context-aware web service systems. **International Journal of Web Information Systems**, v. 5, n. 1, p. 5–31, 2009.
- TVERSKY, A. Features of similarity. **Psychological Review**, v. 84, n. 4, p. 327–352, 1977.
- VALIENTE-ROCHA, P.; LOZANO-TELLO, A. Ontology-based expert system for home automation controlling. In: INTERNATIONAL CONFERENCE ON INDUSTRIAL

ENGINEERING AND OTHER APPLICATIONS OF APPLIED INTELLIGENT SYSTEMS, 23, 2010. **Proceedings...** [S.l.]: Springer-Verlag, 2010, p.661–670.

VASILECAS, O.; KALIBATIENE, D.; GUIZZARDI, G. Towards a formal method for the transformation of ontology axioms to application domain rules. **Information Technology and Control**, v. 38, n. 4, p. 271–282, 2009.

VERNER, L. BPM: The Promise and the Challenge. **Queue**, v. 2, n. March, p. 82, 2004.

WANG, F.; TURNER, K. J. Towards personalised home care systems. In: INTERNATIONAL CONFERENCE ON PERVASIVE TECHNOLOGIES RELATED TO ASSISTIVE ENVIRONMENTS (PETRA), 1, 2008. **Proceedings...** New York, USA: ACM Press, 2008, p.44:1–44:7.

WANG, H. et al. Web services: problems and future directions. **Web Semantics: Science, Services and Agents on the World Wide Web**, v. 1, n. 3, p. 309–320, abr. 2004.

WEIDLICH, M. et al. BPEL to BPMN: The myth of a straight-forward mapping. In: INTERNATIONAL CONFERENCE ON THE MOVE TO MEANINGFUL INTERNET SYSTEMS, 2008. **Proceedings...** [S.l.]: Springer Berlin Heidelberg, 2008, p.265–282.

WESKE, M. **Business Process Management**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

WILSON, M.; KOLBERG, M.; MAGILL, E. H. Considering Side Effects in Service Interactions in Home Automation - an Online Approach. In: INTERNATIONAL CONFERENCE ON FEATURE INTERACTIONS IN SOFTWARE AND COMMUNICATIONS SYSTEMS, 9, 2008. **Proceedings...** [S.l.:s.n], 2008, p.172–187.

World Population Prospects: The 2012 Revision. Disponível em: <http://esa.un.org/unpd/wpp/Documentation/pdf/WPP2012_Press_Release.pdf>.

WU, Z.; PALMER, M. Verb Semantics and Lexical Selection. In: ANNUAL MEETING ON ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 32, 1994. **Proceedings...** [S.l.:s.n], 1994, p.133–138.

XIA, H.; YOSHIDA, T. Web service recommendation with ontology-based similarity measure. In: INTERNATIONAL CONFERENCE ON INNOVATIVE COMPUTING, INFORMATION AND CONTROL (ICICIC), 2, 2007. **Proceedings...** [S.l.]: IEEE, 2008, p.412–412.

XU, B.; LUO, S.; SUN, K. Towards multimodal query in web service search. In: INTERNATIONAL CONFERENCE ON WEB SERVICES (ICWS), 19, 2012. **Proceedings...** [S.l.]: IEEE, 2012, p.272–279.

XU, J. et al. Ontology-Based Smart Home Solution and Service Composition. In: INTERNATIONAL CONFERENCE ON EMBEDDED SOFTWARE AND SYSTEMS, 2009. **Proceedings...** [S.l.]: IEEE, 2009, p.297–304.

ZAMBORLINI, V.; GUIZZARDI, G. On the Representation of Temporally Changing Information in OWL. In: INTERNATIONAL ENTERPRISE DISTRIBUTED OBJECT COMPUTING CONFERENCE WORKSHOPS, 14, 2010. **Proceedings...** [S.l.]: IEEE, 2010, p.283–292.

ZHANG, Y.; LIU, F.; ZHANG, N. Toward Fine Grained Matchmaking of Semantic Web Services Based on Concept Similarity Semantic Web Service Matchmaking System Framework. **Journal of Information and Computational Science**, v. 2, p. 377–384, 2011.

ZHOU, Z. Z. Z.; WANG, Y. W. Y.; GU, J. G. J. A New Model of Information Content for Semantic Similarity in WordNet. In: INTERNATIONAL CONFERENCE ON FUTURE

GENERATION COMMUNICATION AND NETWORKING SYMPOSIA, 2, v. 3, 2008.
Proceedings... [S.l.]: IEEE, 2008, p.85–89.

ZHU, J. Passive Action and Causalism. **Philosophical Studies**, v. 119, n. 3, p. 295–314, jun.
2004.

APÊNDICE A – DISPOSITIVOS E SUAS FUNCIONALIDADES

<i>Dispositivo</i>	<i>Funcionalidade em SW</i>
1. radio_clock	turn_on()
	turn_off()
	show_text(String text)
	play_sound(String soundURL)
2. lamp	turn_on()
	turn_off()
	set_pulsating_light()
	set_dimmed_light()
	set_intense_light()
3. mirror	turn_on()
	turn_off()
	set_zoom_in()
	set_zoom_out()
	set_zoom_normal()
	show_text(String text)
	show_image(String imageURL)
4. sprinkler	turn_on()
	turn_off()
5. microwave	turn_on()
	turn_off()
	show_text(String text)
	play_sound(String soundURL)
6. luminaire	turn_on()
	turn_off()
	set_pulsating_light()
	set_dimmed_light()
	set_intense_light()
7. tap	open()
	close()
8. stove	turn_on()
	turn_off()
9. oven	turn_on()
	turn_off()
10. wash_machine	turn_on()
	turn_off()
11. smart_painting	turn_on()
	turn_off()
	show_text(String text)
	show_image(String imageURL)
12. smart_refrigerator	turn_on()
	turn_off()

	show_text(String text)
	show_image(String imageURL)
	play_sound(String soundURL)
13. exhauster	turn_on()
	turn_off()
14. television_type_1	turn_on()
	turn_off()
	show_text(String text)
	show_image(String imageURL)
	play_sound(String soundURL)
	play_video(String videoURL)
15. agenda_type_1	turn_on()
	turn_off()
	show_text(String text)
	show_image(String imageURL)
	play_sound(String soundURL)
16. computer_type_1	turn_on()
	turn_off()
	show_text(String text)
	show_image(String imageURL)
	play_sound(String soundURL)
	play_video(String videoURL)
17. television_type_2	turn_on()
	turn_off()
	show_text(String text)
	show_image(base64Binary image)
	play_sound(base64Binary sound)
	play_video(base64Binary video)
18. agenda_type_2	turn_on()
	turn_off()
	show_text(String text)
	show_image(base64Binary image)
	play_sound(base64Binary sound)
19. computer_type_2	turn_on()
	turn_off()
	show_text(String text)
	show_image(base64Binary image)
	play_sound(base64Binary sound)
	play_video(base64Binary video)
20. mobile_phone	turn_on()
	turn_off()
	show_text(String text)
	show_image(base64Binary image)
	play_sound(base64Binary sound)
	play_video(base64Binary video)

21. door	open()
	close()
	lock()
	unlock()
22. shower	turn_on()
	turn_off()
23. latrine	turn_on()
	turn_off()
24. gas_controller	open()
	close()
25. electronic_photo_album	turn_on()
	turn_off()
	show_text(String text)
	show_image(String imageURL)
	play_sound(String soundURL)
26. watch	turn_on()
	turn_off()
	show_text(String text)
	play_sound(String soundURL)
27. smart_wardrobe	turn_on()
	turn_off()
	show_text(String text)
	show_image(String imageURL)
28. smart_pillow	turn_on()
	turn_off()
	show_text(String text)
	show_image(String imageURL)
	play_sound(String soundURL)
	play_video(String videoURL)
29. smart_window	turn_on()
	turn_off()
	show_text(String text)
	show_image(String imageURL)
	open()
	close()
	open_halfway()
	lock()
	unlock()
30. curtain	open()
	close()
	open_halfway()
31. roof_light	turn_on()
	turn_off()
	set_pulsating_light()
	set_dimmed_light()

set_intense_light()

APÊNDICE B – EXEMPLO DE DESCRIÇÃO SEMÂNTICA

Segue a de descrição semântica para a “Funcionalidade Abrir” do “Dispositivo Porta”:

```
<?xml version="1.0"?>
<!DOCTYPE uri def [
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns">
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema">
  <!ENTITY owl "http://www.w3.org/2002/07/owl">
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema">
  <!ENTITY service "http://www.daml.org/services/owl-s/1.2/Service.owl">
  <!ENTITY profile "http://www.daml.org/services/owl-s/1.2/Profile.owl">
  <!ENTITY process "http://www.daml.org/services/owl-s/1.2/Process.owl">
  <!ENTITY grounding "http://www.daml.org/services/owl-s/1.2/Grounding.owl">
  <!ENTITY expr "http://www.daml.org/services/owl-s/1.2/generic/Expression.owl">
  <!ENTITY swrl "http://www.w3.org/2003/11/swrl">
  <!ENTITY mind "http://on.cs.unibas.ch/owl-s/1.2/MindswapProfileHierarchy.owl">
  <!ENTITY house "http://127.0.0.1/House_Ontology/house.owl">
  <!ENTITY groundingWSDL "http://housesimulator.herokuapp.com/web_service/wsdl?device=door">
  <!ENTITY this "http://127.0.0.1/OWLS/open_door.owl">
]>

<rdf:RDF
  xmlns:rdf="&rdf;"#
  xmlns:rdfs="&rdfs;"#
  xmlns:owl="&owl;"#
  xmlns:xsd="&xsd;"#
  xmlns:service="&service;"#
  xmlns:profile="&profile;"#
  xmlns:process="&process;"#
  xmlns:grounding="&grounding;"#
  xmlns:expr="&expr;"#
  xmlns:swrl="&swrl;"#
  xmlns:mind="&mind;"#
  xmlns:house="&house;"#
  xml:base="&this;"
  xmlns="&this;"#
>

<!-- ontologias usadas -->
```

```

<owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.2/Profile.owl"/>
  <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.2/Service.owl"/>
  <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.2/generic/Expression.owl"/>
  <owl:imports rdf:resource="http://127.0.0.1/ontology/Expression.owl"/>
  <owl:imports rdf:resource="http://127.0.0.1/House_Ontology/house.owl"/>
</owl:Ontology>

<!-- ===== Service ===== -->
<service:Service rdf:ID="open_door_Service">
  <service:presents rdf:resource="#open_door_Profile"/>

  <service:describedBy rdf:resource="#open_door_Process"/>
  <service:supports rdf:resource="#open_door_Grounding"/>
</service:Service>

<!-- ===== Profile ===== -->
<mind:LanguageService rdf:ID="open_door_Profile">
  <service:presentedBy rdf:resource="#open_door_Service"/>

  <profile:serviceName xml:lang="en">open_door</profile:serviceName>
  <profile:textDescription xml:lang="en">opens the door providing it's active, closed and
unlocked</profile:textDescription>

  <profile:hasInput rdf:resource="#device_id"/>

  <profile:hasExistential rdf:resource="#device_in"/>
  <profile:hasExistential rdf:resource="#status_active"/>
  <profile:hasExistential rdf:resource="#status_closed"/>
  <profile:hasExistential rdf:resource="#status_open"/>
  <profile:hasExistential rdf:resource="#status_unlocked"/>

  <profile:hasOutput rdf:resource="#value"/>

  <profile:hasPrecondition rdf:resource="#hasIDValid"/>
<!-- <profile:hasPrecondition rdf:resource="#isActive"/>
-->

</mind:LanguageService>

<!-- ===== Process ===== -->
<process:AtomicProcess rdf:ID="open_door_Process">

```

```

<service:describes rdf:resource="#open_door_Service"/>

<process:hasInput rdf:resource="#device_id"/>

<profile:hasExistential rdf:resource="#device_in"/>
<profile:hasExistential rdf:resource="#status_active"/>
<profile:hasExistential rdf:resource="#status_closed"/>
<profile:hasExistential rdf:resource="#status_open"/>
<profile:hasExistential rdf:resource="#status_unlocked"/>

<process:hasOutput rdf:resource="#value"/>

<process:hasPrecondition rdf:resource="#hasIDVal id"/>
<process:hasResult rdf:resource="#is0pen"/>

</process:AtomicProcess>

<!--
<process:Input rdf:ID="device_id">
  <process:parameterType rdf:datatype="&xsd;#anyURI">http://127.0.0.1/House_Ontology/
house.owl#Device_ID </process:parameterType>
  <rdfs:label>device's id</rdfs:label>
</process:Input>
-->

<process:Input rdf:ID="device_id">
  <process:parameterType rdf:datatype="&xsd;#anyURI">&xsd;#int</process:parameterType>
  <rdfs:label>device's id</rdfs:label>
</process:Input>

<process:Existential rdf:ID="device_in">
  <process:parameterType
rdf:datatype="&xsd;#anyURI">http://127.0.0.1/House_Ontology/house.owl#Door</process:parameterT
ype>
  <rdfs:label>device</rdfs:label>
</process:Existential>

<process:Existential rdf:ID="status_active">
  <process:parameterType
rdf:datatype="&xsd;#anyURI">http://127.0.0.1/House_Ontology/house.owl#ActiveTrue</process:para
meterType>
  <rdfs:label>status of device</rdfs:label>
</process:Existential>

```

```

    <process: Existential rdf: ID="status_closed">
      <process: parameterType
rdf: datatype="&xsd; #anyURI">http://127.0.0.1/House_Ontology/house.owl#OpenFalse</process: parameterType>
      <rdfs: label>status of device</rdfs: label>
    </process: Existential>

    <process: Existential rdf: ID="status_open">
      <process: parameterType
rdf: datatype="&xsd; #anyURI">http://127.0.0.1/House_Ontology/house.owl#OpenTrue</process: parameterType>
      <rdfs: label>status of device</rdfs: label>
    </process: Existential>

    <process: Existential rdf: ID="status_unlocked">
      <process: parameterType
rdf: datatype="&xsd; #anyURI">http://127.0.0.1/House_Ontology/house.owl#LockedFalse</process: parameterType>
      <rdfs: label>status of device</rdfs: label>
    </process: Existential>

    <expr: SWRL-Condition rdf: ID="hasIDValid">
      <rdfs: label>verifies if the given ID matches the requested device and if the device can be opened</rdfs: label>
      <expr: expressionLanguage rdf: resource="&expr; #SWRL"/>
      <expr: expressionObject>
        <swrl: AtomList>
          <rdfs: first>
            <swrl: ClassAtom>
              <swrl: classPredicate
rdf: resource="http://127.0.0.1/House_Ontology/house.owl#Door" />
              <swrl: argument1 rdf: resource="#device_in"/>
            </swrl: ClassAtom>
          </rdfs: first>
          <rdfs: rest>
            <swrl: AtomList>
              <rdfs: first>
                <swrl: ClassAtom>
                  <swrl: classPredicate
rdf: resource="http://127.0.0.1/House_Ontology/house.owl#ActiveTrue"/>
                  <swrl: argument1 rdf: resource="#status_active"/>
                </swrl: ClassAtom>
              </rdfs: first>
            </swrl: AtomList>
          </rdfs: rest>
        </expr: expressionObject>
      </expr: SWRL-Condition>

```

```

</rdf: first>
<rdf: rest>
  <swrl: AtomList>
    <rdf: first>
      <swrl: ClassAtom>
        <swrl: classPredicate
rdf: resource="http://127.0.0.1/House_Ontology/house.owl#OpenFalse"/>
          <swrl: argument1 rdf: resource="#status_closed"/>
        </swrl: ClassAtom>
      </rdf: first>
    <rdf: rest>
      <swrl: AtomList>
        <rdf: first>
          <swrl: ClassAtom>
            <swrl: classPredicate
rdf: resource="http://127.0.0.1/House_Ontology/house.owl#LockedFalse"/>
              <swrl: argument1 rdf: resource="#status_unlocked"/>
            </swrl: ClassAtom>
          </rdf: first>
        <rdf: rest>
          <swrl: AtomList>
            <rdf: first>
              <swrl: DataValuedPropertyAtom>
<!-- <swrl: IndividualPropertyAtom -->
                <swrl: propertyPredicate
rdf: resource="http://127.0.0.1/House_Ontology/house.owl#hasID"/> <!-- hasID -->
                  <swrl: argument1 rdf: resource="#device_in"/>
                  <swrl: argument2 rdf: resource="#device_id"/>
                </swrl: DataValuedPropertyAtom>
<!-- </swrl: IndividualPropertyAtom -->
              </rdf: first>
            <rdf: rest>
              <swrl: AtomList>
                <rdf: first>
                  <swrl: IndividualPropertyAtom>
                    <swrl: propertyPredicate
rdf: resource="http://127.0.0.1/House_Ontology/house.owl#hasStatus"/>
                      <swrl: argument1 rdf: resource="#device_in"/>
                      <swrl: argument2 rdf: resource="#status_active"/>
                    </swrl: IndividualPropertyAtom>
                  </rdf: first>
                <rdf: rest>
                  <swrl: AtomList>

```



```

    <process: hasResultVar rdf: resource="#open"/>
    <process: inCondition rdf: resource="http://www.daml.org/services/owl-
s/1.2/generic/Expression.owl#AlwaysTrue"/> -->
  </process: Result>

  <!--
  <process: ResultVar rdf: ID="open">
    <process: parameterType
rdf: datatype="&xsd:anyURI">http://127.0.0.1/House_ontology/house.owl#OpenTrue</process: parame
terType>
  </process: ResultVar>

  <expr: SWRL-Expression rdf: ID="effect">
    <rdfs:label>seta o estado de ligado do aparelho para busy = true</rdfs:label>
    <expr: expressionLanguage rdf: resource="#expr;#SWRL"/>
    <expr: expressionObject>
      <swrl: AtomList>
        <rdf: first>
          <swrl: IndividualPropertyAtom>
            <swrl: propertyPredicate
rdf: resource="http://127.0.0.1/House_ontology/house.owl#hasStatus"/>
            <swrl: argument1 rdf: resource="#device_in"/>
            <swrl: argument2 rdf: resource="#status_open"/>
          </swrl: IndividualPropertyAtom>
        </rdf: first>
      </swrl: AtomList>
    </expr: expressionObject>
  </expr: SWRL-Expression>
  -->

  <!-- ===== Grounding ===== -->
  <grounding: WsdlGrounding rdf: ID="open_door_Grounding">
    <service: supportedBy rdf: resource="#open_door_Service"/>
    <grounding: hasAtomicProcessGrounding rdf: resource="#open_door_AtomicProcessGrounding"/>
  </grounding: WsdlGrounding>

  <!-- ===== Atomic Process Grounding ===== -->
  <grounding: WsdlAtomicProcessGrounding rdf: ID="open_door_AtomicProcessGrounding">
    <grounding: owlProcess rdf: resource="#open_door_Process"/>
    <grounding: wsdlDocument
rdf: datatype="&xsd:anyURI">&groundingWSDL;</grounding: wsdlDocument>
    <grounding: wsdlOperation>

```

```

    <grounding: WsdlOperationRef>
      <grounding: portType
rdf: datatype="xsd:anyURI">&groundingWSDL; #web_service_port</grounding: portType>
      <grounding: operation
rdf: datatype="xsd:anyURI">&groundingWSDL; #open_door</grounding: operation>
    </grounding: WsdlOperationRef>
  </grounding: wsdlOperation>

```

```

<grounding: wsdlInputMessage rdf: datatype="xsd:anyURI "
>&groundingWSDL; #open_door</grounding: wsdlInputMessage>
<grounding: wsdlInput>
  <grounding: WsdlInputMessageMap>
    <grounding: wsdlMessagePart rdf: datatype="xsd:anyURI "
    >&groundingWSDL; #device_id</grounding: wsdlMessagePart>
    <grounding: owlParameter rdf: resource="#device_id"/>
  </grounding: WsdlInputMessageMap>
</grounding: wsdlInput>

```

```

<grounding: wsdlOutputMessage rdf: datatype="xsd:anyURI "
>&groundingWSDL; #open_door_response</grounding: wsdlOutputMessage>
<grounding: wsdlOutput>
  <grounding: WsdlOutputMessageMap>
    <grounding: wsdlMessagePart rdf: datatype="xsd:anyURI "
    >&groundingWSDL; #value</grounding: wsdlMessagePart>
    <grounding: owlParameter rdf: resource="#value"/>
  </grounding: WsdlOutputMessageMap>
</grounding: wsdlOutput>

```

```

</grounding: WsdlAtomicProcessGrounding>

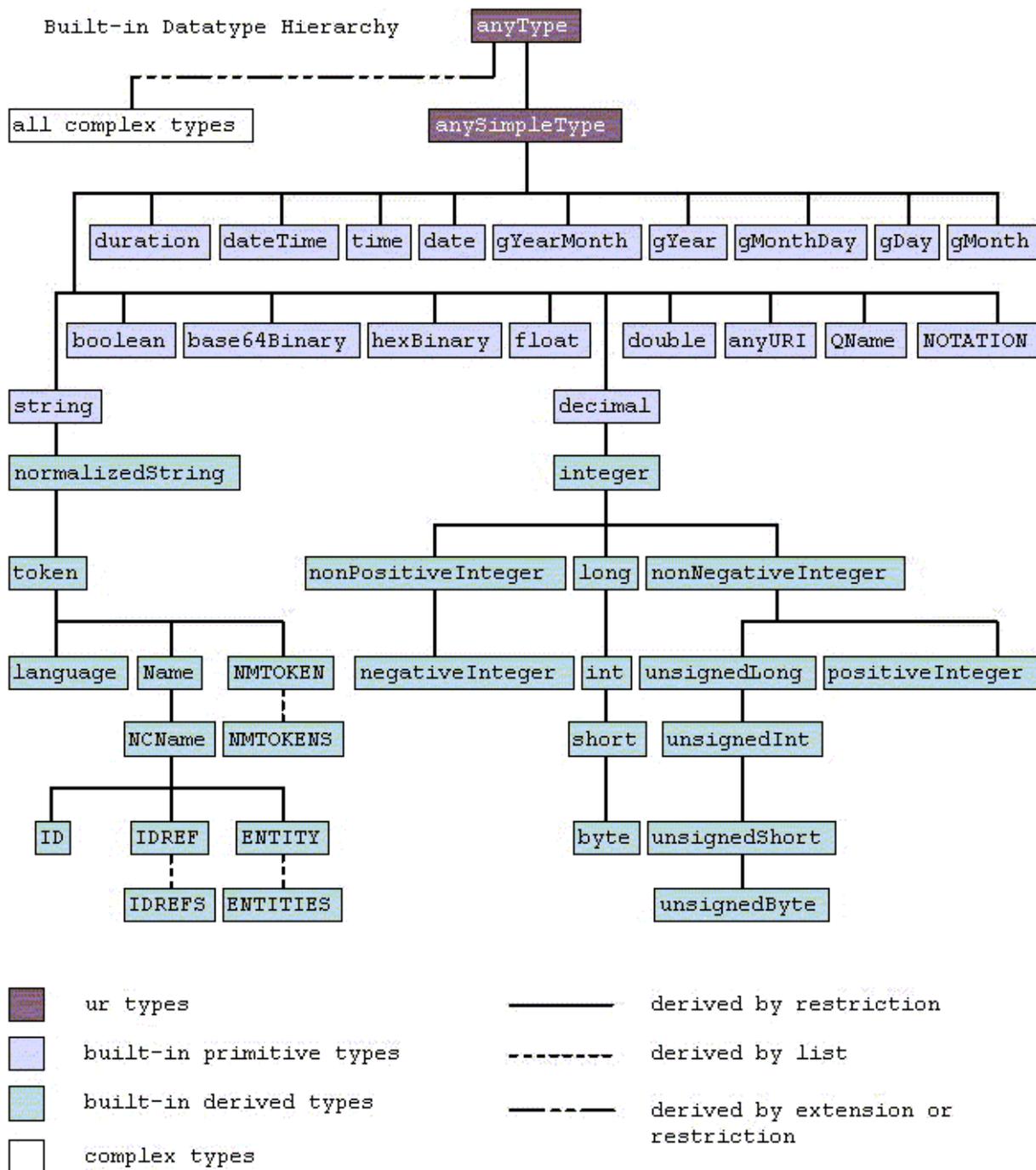
```

```

</rdf: RDF>

```

ANEXO A – HIERARQUIA DOS DATATYPES DO XML



Fonte: imagem retirada de Biron e Malhotra (2004)