

Universidade Federal do Rio Grande
Pós-Graduação em Modelagem Computacional

Sistema Toogle, Ferramenta de Design e Navegação para CPS

Marcos André do Amaral Bichet

Rio Grande, 25 de Outubro de 2013

Universidade Federal do Rio Grande
Pós-Graduação em Modelagem Computacional

Sistema Toogle, Ferramenta de Design e Navegação para CPS

Marcos André do Amaral Bichet

Proposta de Dissertação de Mestrado apresentada ao Pós-Graduação em Modelagem Computacional da Universidade Federal do Rio Grande.

Orientador(a): Prof. Dr. Silvia Silva da Costa Botelho

Co-orientador(a):

Rio Grande, 25 de Outubro de 2013

Este trabalho foi analisado e julgado adequado para a obtenção do título de Mestre e aprovado em sua forma final pelo orientador.

Prof. Dr. Silvia Silva da Costa Botelho

Banca Examinadora:

Prof. Dr. Rodrigo da Silva Guerra

DPEE / CT / UFSM

Prof^ª. Dr. Danúbia Bueno Espíndola

C3 – FURG

Prof. Dr. Alessandro de Lima Bicho

C3 – FURG

Conteúdo

Lista de Figuras	iv
Lista de Abreviaturas	vi
Resumo	vii
Abstract	viii
1 INTRODUÇÃO	1
1.1 Ambientes Virtuais-Reais	2
1.2 Motivação	6
1.3 Objetivo Geral	7
1.4 Estrutura da Dissertação	8
2 SISTEMAS FÍSICOS CIBERNÉTICOS	9
2.1 Definições	11
2.2 Desafios de Projeto	12
2.3 Aplicações	14
2.4 Trabalhos Relacionados	15
2.4.1 Arquiteturas para CPS	15
2.4.2 Protocolos, Semânticas e Ontologias para Rede de Sensores e Atuadores	22
2.5 Considerações Finais	25
3 UM MODELO CONCEITUAL PARA CPS E HIPERAMBIENTES	26
3.1 O Modelo Conceitual	27

3.1.1	As Entidades do Modelo	28
3.1.2	Recursos	29
3.1.3	Relacionamento entre as Entidades do Modelo	30
3.2	Hiperambientes	30
3.3	Modelando um Sistema Físico Cibernético	33
3.4	Considerações Finais	38
4	TOOGLE: UMA PLATAFORMA DE DESENVOLVIMENTO DE CPS	40
4.1	Toogle Dispositivo	42
4.2	Toogle Middleware	44
4.3	Toogle Editor	45
4.4	Toogle Navegador	49
4.5	Considerações Finais	53
5	IMPLEMENTANDO HIPERAMBIENTES COM A TOOGLE	54
5.1	Passo 1: Adicionando dispositivos - sensores e atuadores	55
5.2	Passo 2: Adicionando serviços e recursos	55
5.3	Passo 3: Modelando um ambiente físico	56
5.4	Passo 4: Procedimentos básicos do Sistema Toogle	57
5.4.1	O Editor Toogle	57
5.4.2	O Editor Toogle Persistência de Dados	58
5.5	Considerações Finais	59
6	TESTES E RESULTADOS	61
6.1	Caso de Estudo: Estaleiro	62
6.1.1	Etapa 1: Modelo 3D do Estaleiro Rio Grande	62
6.1.2	Etapa 2: Maquete do Estaleiro Rio Grande	62
6.1.3	Etapa 3: Adicionando os dispositivos ao cenário	64
6.2	Resultados do primeiro teste	65
6.3	Plataforma Robotica Tatubot	67
6.4	Considerações Finais	73

<i>CONTEÚDO</i>	iii
7 CONCLUSÃO	74
7.1 Resumo do Trabalho	75
Bibliografia	77

Lista de Figuras

2.1	Componentes de um Sistema Físico Cibernético.	10
2.2	Um Mapa Conceitual de CPS, composto por diversas áreas de estudo e aplicações, acessado em http://cyberphysicalsystems.org/	12
2.3	Exemplo de estrutura de um CPS	13
2.4	Arquitetura SOA para CPS.	18
2.5	Arquitetura básica do ROS.	20
3.1	Modelo Conceitual da Arquitetura, demonstra no diagrama UML o relacionamento entre os seus elementos.	27
3.2	Ambiente Exemplo	32
3.3	Ambiente físico exemplo.	34
3.4	Exemplo do ambiente proposto em EEML	38
3.5	Exemplo do smart object cadeira em EEML	39
4.1	Sistema Toogle e um exemplo de hiperambiente	42
4.2	Tela do Toogle-Dispositivo	43
4.3	Sistema Toogle Navegador/Editor, adicionando novo nodo RFID	46
4.4	Interface do Editor de Hiperambientes da Toogle/Blender	47
4.5	Painel Toogle/Blender - Objetos da Toogle	48
4.6	Editor de propriedades do Objeto	48
4.7	Arquitetura proposta do Navegador web para Hiperambientes	51
4.8	Arquitetura dos componentes para o Navegador web de Hiperambientes	52
5.1	Arquitetura de serviços e recursos	56

5.2	Exemplo de objeto modelado em 3D	57
5.3	Exemplo do Hiperambiente em execução pelo Editor/Navegador Toogle	58
5.4	Exemplo do Fluxo de dados gerados para reutilização posterior	60
6.1	Modelo 3D do Estaleiro Rio Grande	63
6.2	Maquete 3D do Estaleiro Rio Grande	63
6.3	Leitor e Etiquetas RFIDs	64
6.4	Estrutura Semântica do Modelo Conceitual aplicada no teste	65
6.5	Arquitetura da Plataforma Toogle para o Hiperambiente Estaleiro	66
6.6	Toogle-Navegador com informações do tópico tagsLidas e propriedades do objeto	67
6.7	Ilustração da disposição dos cabos, dubo e do Tatubot.	68
6.8	Arquitetura Toogle Tatubot	69
6.9	Editor/Visualizador de dispositivos físicos Toogle	70
6.10	Semantica Toogle Tatubot	71
6.11	Representação grafica 3D do Tatubot	71
6.12	Plataforma robotica real Tatubot	72
6.13	Visualização do Sistema Toogle na CAVE	72

Lista de Abreviaturas

FURG Fundação Universidade Federal do Rio Grande

ROS Robotic Operacional System

IoT Internet of Toogles

CAVE Cave automatic virtual environment

GUI Graphic User Interface

BGE Blender Game Engine

IFC Industry Foundation Classes

AUV Autonomous underwater vehicle - Veículo autônomo subaquático

CPS Ciber Physical System - Sistema Ciber Físico

RFID Radio-Frequency IDentification - Identificação por Rádio Frequência

WIFI Wi-Fi é uma marca registrada pela Wi-Fi Alliance

Resumo

A popularização do uso de dispositivos eletrônicos com poder para percepção e atuação no meio computacional, apresenta um cenário propício para interação entre as diferentes tecnologias e dispositivos. Neste novo contexto tecnológico novas relações entre os seres humanos e seu meio, seja físico e/ou virtual, podem se estabelecer, requerendo o estudo, para desenvolver métodos capazes de suportar o potencial de troca e fluxo de informação, decorrente deste novo cenário sócio-interacionista. Esta dissertação visa estudar estas relações provenientes da intercessão de ambientes reais e virtuais, através de modelos e ferramentas computacionais que os dêem suporte. O trabalho resgata os conceitos de Sistemas Físicos Cibernéticos (CPS), internet das coisas (IoT), Hipertexto e protocolos de comunicação, buscando na sua integração, subsídios para tratar os desafios tecnológicos presentes nestes novos ambientes de interação da sociedade.

Abstract

The popularization of the use of electronic devices with power for perception and action in the computational environment, presents a scenario conducive to interaction between different technologies and devices. In this new technological context new relationships between humans and their environment, whether physical and/or virtual, can be established, justifying the study to development of methods capable of withstanding the potential for exchange and flow of information, under this new scenario socio-interactionist. This dissertation aims to study these relationships from the intersection of real and virtual environments, through models and computational tools that provide the foundation. The work rescues the concepts of Cyber Physical Systems (CPS), Internet of Things (IoT), Hypertext and communication protocols, seeking integration, subsidies to address the technology challenges present in these new environments of interaction of society.

Capítulo 1

INTRODUÇÃO

Neste capítulo é contextualizado o problema de desenvolvimento de sistemas que integrem elementos físicos e virtuais. Inicialmente é apresentada a motivação desta pesquisa, bem como os principais conceitos envolvidos no trabalho. São abordadas as áreas de Sistemas Físicos e Cibernéticos, Internet das Coisas, bem como Protocolos e Métodos de Descrição para Redes de Sensores, ressaltando-se os aspectos relacionados a arquiteturas, modelos de representação e protocolos de comunicação. Após o objetivo desta dissertação é apresentado o qual constitui-se no estudo e desenvolvimento de uma plataforma para a criação, o acesso e a manipulação de informações envolvidas em Sistemas Físicos Virtuais. Finalmente, apresenta-se a estrutura de organização dos demais capítulos da dissertação

A popularização do uso de dispositivos eletrônicos com poder para percepção e atuação no meio computacional, tais como smartphones, tablets, notebooks RFIDs, redes de sensores e outros, mais a popularização da internet, traz um cenário propício para interação entre estas diferentes tecnologias e dispositivos. Com as diversas possibilidades de interação, como redes sociais, compartilhamento de arquivos, jogos colaborativos; tem-se cada vez mais o mundo virtual sobreposto ao real.

Neste novo contexto tecnológico novas relações entre os seres humanos e seu meio, seja ele físico e/ou virtual, começam a se estabelecer, criando uma demanda pelo estudo,

a análise e o desenvolvimento de novos conceitos, ferramentas, protocolos e aplicativos capazes de suportar o potencial de trocas e fluxo de informação decorrentes deste novo panorama sócio-interacionista.

1.1 Ambientes Virtuais-Reais

Na literatura, muitas são as áreas envolvidas no estudo de ambientes que integram sistemas virtuais e reais. Em computação pode-se citar as áreas de Redes de Computadores, Robótica, Computação Gráfica, Sistemas Distribuídos, Interface Humano-computador. Diferentes terminologias estão sendo criadas e adaptadas de forma a cobrir os diferentes aspectos teórico-tecnológicos envolvidos [Carvalho, 2011].

Ambientes Virtuais Surgem da possibilidade de simular, via computador, ambientes compostos por elementos virtuais. No que tange ao surgimento de novos ambientes virtuais pesquisas em realidade mista envolvendo cenários virtuais e realidade aumentada permitem a intersecção de mundos virtuais com agentes reais [Milgram, 1994]. Em função da predominância de cada ambiente (real ou virtual), surgem diferentes taxonomias. Na *realidade virtual*, encontram-se ambientes puramente virtuais; na *realidade aumentada*, ambientes predominantemente reais e completos, com adições de informações virtuais; na *realidade mista*, encontram-se ambientes que possuem informações reais e virtuais de forma complementar, isto é, tanto as informações reais quanto as virtuais são indispensáveis [e Joseph A. Paradiso, 2009], por exemplo, pode-se citar o *Physical Visualization Sub-league* [Guerra, 2008], onde mini-robôs reais jogam futebol sobre uma projeção a qual apresenta o campo e a bola virtuais.

Ambientes Ubíquos Recentes desenvolvimentos em novas mídias para percepção e atuação em redes de alta escalabilidade, como RFID's [Finkenzyler, 2003] e redes de sensores, vem permitindo o estabelecimento de novos ambientes digitais/reais, aqui denominados *ambientes ubíquos*¹, onde informações circulam entre seus diferentes elementos. Tais ambientes baseiam-se nas idéias lançadas por Weiser, envolvendo computação ubíqua. Na sua proposta a computação estaria presente em qualquer tempo e em qualquer lugar

¹O termo computação pervasiva também pode ser utilizado para descrever sistemas ubíquos

[Weiser, 1991], envolvendo diferentes objetos que dotados de computação embarcada, poderiam interagir entre si, percebendo e adaptando-se ao ambiente de forma transparente, tornando mais simples a interação humano-computador. Neste novo contexto, novos paradigmas surgem visando tratar as relações entre os agentes existentes de forma a otimizar os diferentes critérios do grupo. Estudos vêm sendo realizados visando a criação de ambientes inteligentes, também chamados de ambientes cientes de contexto, do inglês *context-aware environments* [Schilit, 1994].

Paralelamente, surgem os estudos associados à área denominada *Internet das Coisas* (IoT) [Zouganeli, 2009, Yinghui Huang, 2010], denominando-se *coisa*² os diferentes objetos reais embarcados com algum tipo de computação que os possibilitem ter um esquema de endereçamento único, a capacidade de interagirem uns com os outros e de colaborarem com os seus vizinhos para alcançar objetivos comuns [Atzori, 2010]. A IoT, em conformidade com as ideias da computação ubíqua, tem buscado a definição de modelos para a interconexão de qualquer "*coisa*" (objetos, computadores, animais, pessoas e etc) em uma rede, de forma análoga à Internet. O EPCglobal [Yinghui Huang, 2010], IPSO [Dunkels, 2008], UID architecture [uID Center, 2011] e o CASAGRAS [Forum, 2011] são apenas algumas das propostas iniciais para integrar as "*coisas*" em uma rede universal, observando que ainda não prevalece um padrão para este tipo de sistema.

Considerando a grande quantidade de informações que serão geradas pelos incontáveis nodos da IoT, mecanismos de gerenciamento escalonáveis deverão ser previstos. Supondo que cada nodo é capaz de perceber, reagir, adaptar-se e colaborar com outros nodos de forma a alcançar objetivos, espera-se a emergência de complexos comportamentos inteligentes conduzindo a ambientes ubíquos adaptativos, flexíveis e evolutivos. Assim, a complexidade decorrente da necessária escalabilidade dos nodos processadores faz surgir uma série de propostas visando o desenvolvimento de diferentes *middlewares*³ que objetivam tratar os principais desafios da ubiquidade, tais como a comunicação e interação entre dispositivos heterogêneos, a entrada e saída de novos dispositivos no ambiente, a adaptabilidade do ambiente, o gerenciamento de informações contextuais, questões de

²Tradução do inglês *Thing*

³Camada de software que encapsula e resolve problemas de baixo nível tornando menos complexo o desenvolvimento de aplicações específicas.

segurança e privacidade, entre outros.

Dentre os trabalhos que envolvem ambientes reais e virtuais pode-se citar a "dual reality" [Lifton, 2007] ou "cross-reality" [Joseph A. Paradiso, 2009]. Estes constituem-se de ambientes resultantes da interligação entre elementos reais e virtuais através da mediação de redes de sensores e atuadores (*cross devices*). Embora ambos sejam auto-contidos, eles se beneficiam da capacidade de mutuamente refletir, influenciar e fundir-se um ao outro. Essa abordagem considera explicitamente a possibilidade de co-existência de ambientes virtuais e reais, focalizando-se no estudo de dispositivos de cruzamento/interfaceamento entre eles. Entretanto, os autores consideram cada ambiente como auto-contido com seus elementos próprios e distintos: ambiente real composto de agentes reais, ambiente virtual povoado por elementos virtuais e redes de sensores e atuadores realizando as trocas de informação entre eles [e Joseph A. Paradiso, 2009, Mirco Musolesi, 2008]. Também os dispositivos de cruzamento são considerados elementos passivos, não exibindo qualquer tipo de inteligência que aborde aspectos de adaptabilidade e sensibilidade ao contexto.

Sistemas Físicos-Cibernéticos O termo CPS - *Cyber Physical Systems* surgiu por volta do ano de 2006, quando foi cunhado por Helen Gill na Fundação Nacional de Ciência dos Estados Unidos, tendo como objetivo estudar, desenvolver e prover soluções capazes de realmente tratar Sistemas Híbridos que envolvessem elementos virtuais e reais, integrando a computação com os processos físicos [Lee, 2013]. Os CPS seriam uma evolução dos Sistemas Ubíquos, possibilitando que os processos físicos além de serem integrados à Sistemas Computacionais através do seu monitoramento e controle, estes também afetem a computação e vice-versa. CPS's englobam a intersecção, e não somente a união, do físico com o virtual (cibernético), conduzindo a extensão dos demais conceitos associados a ambientes virtuais e ubíquos.

À medida que a complexidade e escala destes sistemas híbridos aumenta, ressalta-se a incapacidade de modelagem rigorosa das possíveis interações entre o físico e o virtual. Torna-se difícil estimar o grau de segurança destes sistemas, aumentando a chance de falhas e diminuindo a possibilidade de sua predição. Para o seu perfeito entendimento não basta estudar os componentes físicos e computacionais separadamente e sim estudar sua interação, requerendo a compreensão da dinâmica conjunta de computadores, software,

redes e processos físicos, constituindo-se em uma nova área de investigação científico-tecnológica.

Muitas são as aplicações associadas atualmente a CPSs, e muitas outras não são sequer vislumbradas. Dentre os sistemas já atualmente presentes no cotidiano pode-se citar aplicações em sistemas de comunicação Zhang [2013], rastreamento de insumos, infraestrutura Maiden [2005], monitoramento e supervisão de sistemas de distribuição de energia Zhang [2013], Lin [2008], telemedicina e monitoramento do paciente Wang [2010], rastreamento e acompanhamento de processos industriais, logística militar, robótica distribuída e transporte em cidades inteligentes, dentre outros. Para Ji Eun Kim [2008] e Kaiyu Wan [2010] o projeto de CPS envolve os seguintes requisitos a serem atendidos:

- Heterogeneidade. O desenvolvimento de CPS passa por arquiteturas que dêem suporte a diferentes tipos de sensores e atuadores, integrando-os e possibilitando a sua interação;
- Modelagem física de diferentes ambientes. O ambiente de desenvolvimento deve permitir a modelagem e simulação de expressões matemáticas, incorporando descrições de modelos físicos específicos;
- Tratamento da Escalabilidade. Capacidade de tratar desde uma pequena quantidade de elementos (dezenas), até escalas maiores (milhares) de sensores e atuadores.
- Suporte a Mobilidade. A arquitetura deve considerar a natureza distribuída e móvel dos seus elementos. A comunicação é fator importante no sistema;
- Integração com soluções proprietárias e de apoio a normas abertas: soluções proprietárias e de normas de código aberto, incluindo protocolos, infra-estruturas e software existente capaz de ser facilmente incorporado em uma estrutura genérica;
- Reutilização de software: a estrutura deve apoiar a reutilização de software através de técnicas de exploração e geração de código, interconectando bibliotecas ou usando componentes configuráveis;
- Usabilidade: Representação gráfica de modelagem e ambiente de simulação devem ser previstas de forma a permitir o desenvolvimento fácil de novas aplicações. Mo-

delagem 3D de ambientes de domínio específico também pode ser considerado, permitindo a fácil edição e navegação nestes.

Frente a crescente demanda por aplicações que envolvam CPS e aos requisitos apontados para o seu desenvolvimento começam a ser propostos *frameworks* capazes de permitir a sua implementação de maneira simples, intuitiva e confiável [Dillon, 2010] [Ji Eun Kim, 2008][Kaiyu Wan K.L., 2010]. Entre os diversos desafios para a sua implementação, Jianhua Shi [2011] aponta a dificuldade em conceber, modelar e tratar sistemas em escalas e complexidades cada vez maiores, com uma metodologia e semânticas adequadas; a necessidade de concepção de sistemas de controle híbridos (capazes de tratar o domínio de tempo contínuo e discreto); a garantia de confiabilidade e segurança; a necessidade de protocolos de comunicação capazes de dar suporte a heterogeneidade dos elementos, aspectos de verificação, validação e certificação; e finalmente o desenvolvimento de ferramentas para modelagem de sistemas físicos e cibernéticos.

1.2 Motivação

Como visto, os avanços tecnológicos atuais conduzem a uma convergência de sistemas onde elementos cibernéticos e físicos coexistirão em grande escala, interagindo em diferentes níveis de colaboração. Percebe-se a necessidade de entendimento e tratamento dos diferentes papéis possíveis a cada elemento, bem como das possíveis relações que possam vir a ocorrer. Assim, surge a necessidade de estabelecer-se novos paradigmas capazes de conviver com a intrínseca escalabilidade e heterogeneidade dos seus membros constituintes, possibilitando a emergência de comportamentos inteligentes decorrente das complexas interações entre seus componentes frente a objetivos a serem alcançados.

No âmbito do Grupo de Automação e Robótica Inteligente - NAUTEC/FURG, Carvalho [2011] apresentou um estudo envolvendo combinação de diferentes ambientes. Neste foi explorado o termo *hiperambiente*, o qual descreve um tipo de ambiente que integra diferentes categorias de componentes: elementos reais, virtuais (avatares) e "*coisas*", combinando diferentes paradigmas tecnológicos, tais como ambientes ubíquos, ambientes virtuais e duais. No hiperambiente seus membros poderão interrelacionar-se de forma a atender determinados objetivos, provocando a emergência de inteligência no sistema. Como

trabalhos futuros, esta proposta apontou a necessidade de um *framework* genérico de desenvolvimento de hiperambientes, o qual possibilitasse a criação, acesso e gerenciamento de informações advindas de elementos reais e virtuais.

Recuperando o crescente interesse no desenvolvimento de CPS, e sendo estes uma evolução dos ambientes virtuais e ubíquos, busca-se estender o conceito de hiperambientes de forma a fornecer uma semântica adequada ao tratamento dos componentes destes sistemas. Protocolos de acesso e compartilhamento de dados do tipo "*Semantic Sensor Web*" [Sheth, 2008] e [Haque, 2013] poderão ser utilizados visando a interoperabilidade de sensores remotos em grids de sensores heterogêneos de alta escalabilidade ao redor do mundo. A associação de semântica a estas informações permitiria que a rede, seus sensores e os dados resultantes, pudessem ser organizados, instalados, gerenciados, consultados, compreendidos e controlados por meio de especificações de alto nível. Ontologias para sensores poderiam fornecer uma estrutura para descrever dispositivos, serviços e recursos, com um forte foco em Geolocalização. Ferramentas de simulação e visualização de sistemas físicos poderiam permitir a modelagem e simulação de seus elementos físicos, facilitando sua usabilidade.

Partindo da hipótese que hiperambientes sejam o *locus* de integração das diferentes informações advindas de elementos reais e virtuais, e que estes comporão Sistemas Físicos-Cibernéticos, seria interessante resgatar conceitos e metodologias de diferentes áreas da computação de forma a fornecer uma plataforma para o desenvolvimento, acesso e representação de hiperambiente os quais descrevem CPSs. Esta deveria atender os requisitos de projeto de CPS apresentados.

1.3 Objetivo Geral

Esta dissertação tem como objetivo apresentar uma plataforma, denominada Toogle, para o desenvolvimento de CPS. Esta plataforma permitirá a criação, acesso e manipulação das informações associadas aos elementos reais e virtuais, encapsuladas em um hiperambiente. Os requisitos de projeto de CPS serão tratados, sobretudo aqueles relacionados as questões de heterogeneidade, abstração semântica e modularidade, interoperabilidade e simulação, bem como edição e navegação.

De forma mais específica, Edward A. Lee [2010] propõe os seguintes objetivos:

- Revisão bibliográfica sobre CPS e protocolos de comunicação de dados para web;
- Definição de um modelo conceitual, seus componentes e relações, com o objetivo de possibilitar o uso de CPS e atendendo aos seus requisitos;
- Extensão do Conceito de Hiperambiente e a utilização do protocolo "*Semantic Sensor Web*" EEML para a sua descrição;
- Desenvolvimento da plataforma Toogle para implementação, edição e navegação de CPS utilizando Hiperambientes; e
- Teste e validação da plataforma Toogle em cenários virtuais e físicos desenvolvidos neste trabalho.

1.4 Estrutura da Dissertação

Neste capítulo, foi introduzido o problema de interação e integração entre domínios físicos e virtuais. Foram apresentados, também, os objetivos, justificativas e contribuições deste trabalho. O capítulo 2 apresenta um resumo dos trabalhos relacionados, a fim de proporcionar uma compreensão do panorama das pesquisas em CPS pelo qual se decide adotar os métodos empregados neste trabalho. Após, é descrito no capítulo 3 o modelo conceitual adotado, e a associação do hiperambiente a este modelo. O capítulo 4 é dedicado a descrição da plataforma Toogle. Com a apresentação da Toogle, o capítulo 5 e 6 trata questões sobre a aplicação da proposta em casos de estudo.

Capítulo 2

SISTEMAS FÍSICOS CIBERNÉTICOS

Este capítulo visa apresentar um panorama geral de pesquisas relacionadas, a fim de possibilitar uma compreensão dos objetivos e da abordagem adotada neste trabalho. Primeiramente, são apresentados os trabalhos atuais e os desafios associados a Sistemas Físicos e Virtuais, discutindo as direções das pesquisas. Após, são discutidas as eventuais abordagens sobre modelos conceituais de arquitetura. Então, são apresentadas propostas de semânticas para compartilhamento de dados para CPS.

Sistemas Físicos Cibernéticos estão sendo utilizados para descrever sistemas projetados para determinado fim, os quais são compostos por diferentes tipos de componentes físicos e virtuais ¹. A Figura 2.1 apresenta uma representação de um CPS. Nesta percebe-se a presença de *entidades físicas* as quais compõem o espaço real. Neste ambiente está imerso também o domínio dos *dispositivos*, o qual se constitui de redes de sensores e atuadores, que conectados em rede disponibilizam *recursos* de percepção e a *atuação* no espaço real. O espaço cibernético é composto por *entidades digitais* as quais realizam computação.

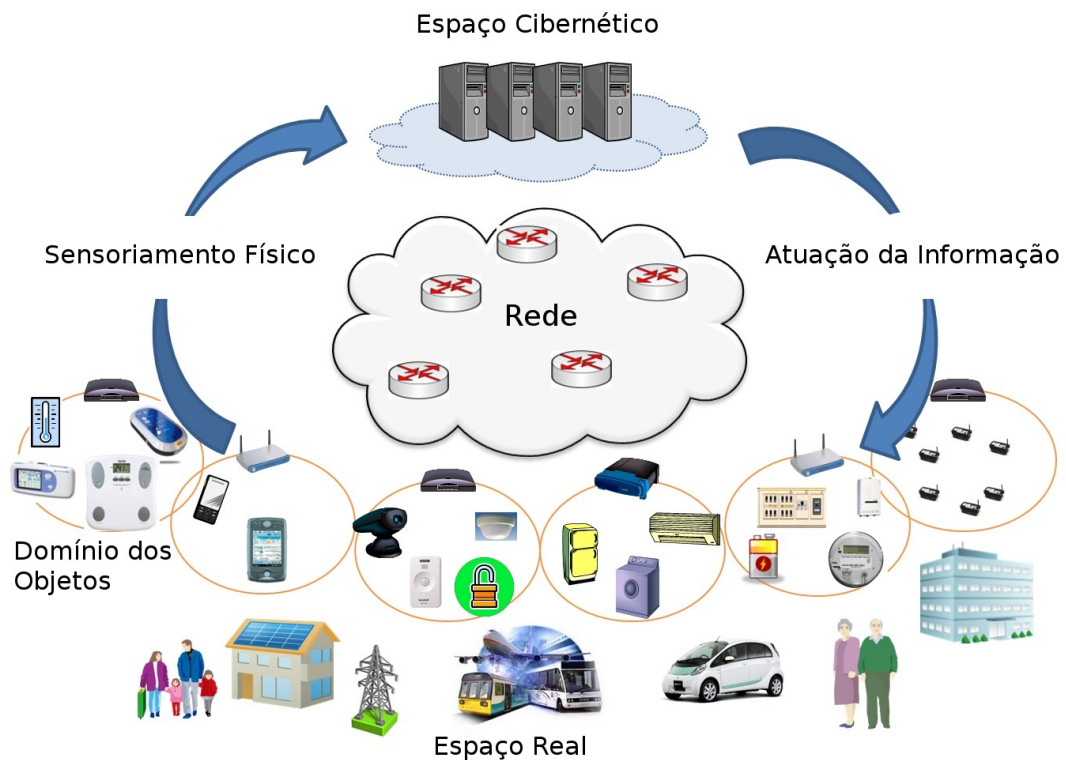


Figura 2.1: Componentes de um Sistema Físico Cibernético.

O estudo e desenvolvimento de sistemas CPS ainda é recente, envolvendo pesquisas com sistemas embarcados, redes de computadores e software embarcado em dispositivos, tais como automóveis, brinquedos, dispositivos médicos e instrumentos científicos, cuja a missão principal não é a computação. CPS's integram a dinâmica dos processos físicos com os de software, proporcionando abstrações, modelagem, projeto e técnicas de análise para a integração do todo. A Figura 2.2 apresenta uma visão geral das questões associadas ao

¹As palavras *virtual*, *cibernético* e *digital* são utilizadas como sinônimas neste manuscrito.

desenvolvimento de CPS. Nela percebe-se três grandes categorias de análise: *i.* definições; *ii.* desafios de projeto; e *iii.* aplicações.

Mais do que a união destes elementos distintos, como no caso de abordagens envolvendo Grid de Sensores e Atuadores baseadas em Internet das Coisas, Sistemas Ubíquos e Realidade Mista, CPS vão além, explorando ainda as integrações dos processos dinâmicos que decorrem da computação e da evolução de processos físicos [Lee, 2008]. Pode-se dizer que uma propriedade central de um sistema físico é a sua dinâmica, a evolução contínua do seu estado ao longo do tempo. Em ambientes virtuais, a dinâmica é reduzida à sequências discretas de mudanças de estado sem semântica temporal. Tendo como objetivo o desenvolvimento de sistemas confiáveis e robustos, busca-se com CPS considerar a integração de elementos advindos de ambos os ambientes, considerando a integração destes aspectos distintos quanto a sua dinâmica de evolução.

2.1 Definições

Sistemas Físico Cibernéticos são sistemas de controle, monitoramento e supervisão que envolvem elementos físicos e virtuais, admitindo inclusive a presença de seres humanos no processo de interação e realimentações. Estes são sistemas distribuídos e/ou em rede que se estabelecem através de redes de sensores e atuadores geralmente sem fio. As interações entre os elementos geram comportamentos inteligentes e adaptativos [Lee, 2013], envolvendo aspectos de predição e tempo real Broman [2013]. Devido à capacidade de processamento, heterogeneidade e escalabilidade do grid surge a necessidade de maior autonomia dos elementos, requerendo metodologias de auto-organização e reorganização de redes móveis. Com a vasta quantidade de dados brutos coletados torna-se essencial o refinamento da informação, envolvendo o estudo de técnicas de *big data*² e mineração.

Um sistema cyber-físico típico terá uma estrutura como representada na figura 2.3, a qual apresenta três plataformas de computação em rede, cada uma com seus próprios sensores e atuadores. Os atuadores afetam os dados fornecidos pelos sensores, através da planta física. Numa aplicação de automação, por exemplo, os atuadores controlam

²*big data* se trata de um conceito, no qual o foco é o grande armazenamento de dados e maior velocidade.

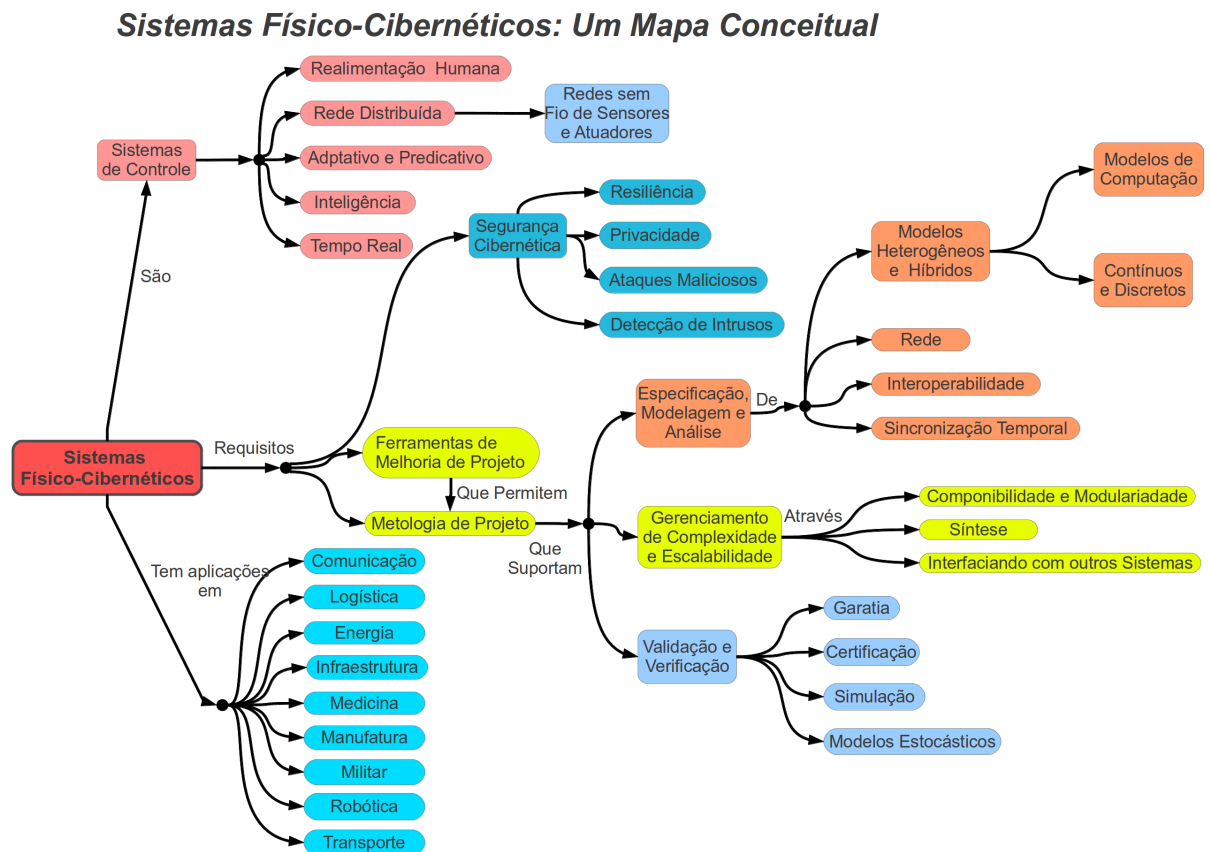


Figura 2.2: Um Mapa Conceitual de CPS, composto por diversas áreas de estudo e aplicações, acessado em <http://cyberphysicalsystems.org/>

as máquinas de impressão de alta velocidade, sensores poderiam detectar as interrupções e os algoritmos de controle podem incluir modos de desligamento rápidos, para evitar danos ao equipamento em caso de atolamentos de papel. Esses desligamentos devem ser rigorosamente planejados em todo o sistema de prevenção de desastres.

2.2 Desafios de Projeto

O projeto de CPS's seguros e confiáveis envolve um conjunto de aspectos. De forma mais genérica, pode-se dividir os requisitos de projeto a serem atendidos em duas categorias de análise *i.* técnicas de segurança, e *ii* metodologias de projeto.

Técnicas de segurança se fazem necessárias para tratar as questões referentes à pri-

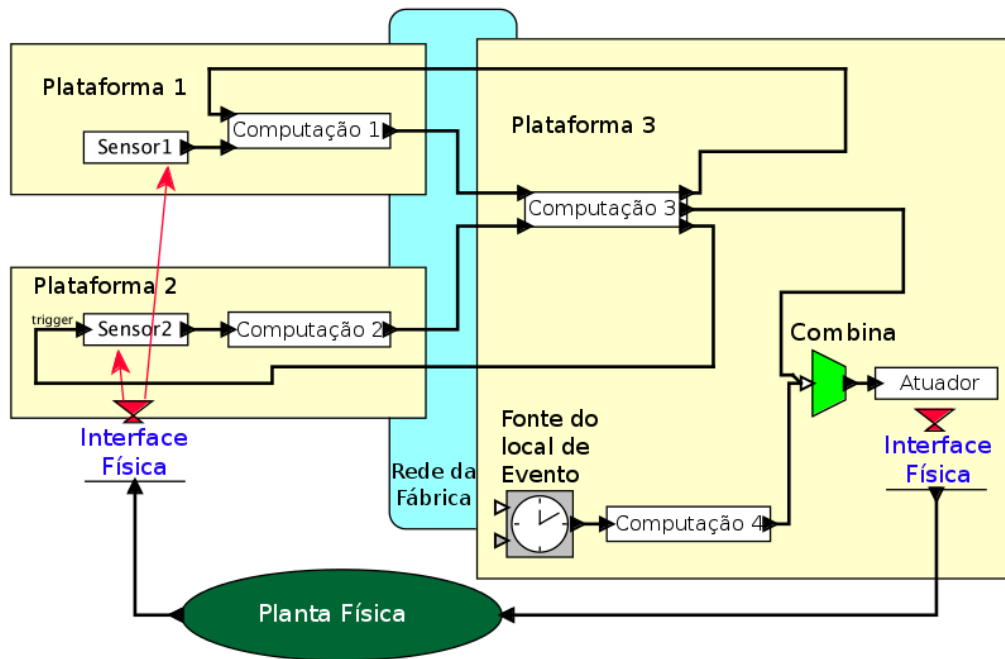


Figura 2.3: Exemplo de estrutura de um CPS

vacidade [Kaiyu Wan K.L., 2010] e à resiliência [Zhu, 2013], envolvidas em redes de alta escalabilidade, principalmente aquelas formadas por dispositivos WIFI, bem como aos aspectos de ataques e invasões [Akella, 2013].

Por outro lado Metodologias de Projeto devem suportar a especificação, a modelagem e a análise de Sistemas Contínuos e Discretos [Ziemer, 1998]. Embora existam vários métodos de desenvolvimento baseadas em modelos, eles ainda não atendem os requisitos e demandas em CPSs. Computação, comunicações e as dinâmicas físicas devem ser abstraídas e modeladas em diferentes níveis de escala, localização e granularidade de tempo [Lee, 2013]. Se faz necessária a adequação e o desenvolvimento de novas teorias matemáticas capazes de mesclar sistemas baseados em eventos (sistemas discretos) com sistemas baseados em tempo contínuo para o controle e feedback [Shi, 2011].

Questões referentes a sincronização temporal Broman [2013], a interoperabilidade [Elmroth, 2010] e a natureza distribuída dos componentes devem ser tratados. Busca-se fornecer níveis de abstrações a serem aplicadas aos sistemas embarcados em tempo-real e suas computações, novos esquemas de alocação de recursos e de tolerância a falhas que envolvam o tratamento de alta escalabilidade e heterogeneidade do sistema, visando a

sua otimização [Lee, 2013] e [Shi, 2011].

Outro desafio refere-se ao desenvolvimento de modelos semânticos de tempo que representem dados temporais de forma consistente, tratando os aspectos relacionados a computação discreta, a medição do tempo contínuo e a sincronização de diversos relógios. Novos modelos de computação e complexidade podem surgir [Du, 2011], resultando em novos paradigmas para a Especificação e Desenvolvimento de Projetos [e Lee Edward A e Vincentelli A Sangiovanni, 2012].

O gerenciamento da escalabilidade e da complexidade também deve ser atadado através de projetos modulares, baseados em componentes [Donassolo, 2010], os quais sintetizam os diferentes serviços e processo envolvidos. A utilização de padrões de interface bem definidos permitiriam a inclusão de novas funcionalidades e processos com baixo custo de desenvolvimento [Buyya, 2009].

Finalmente a Metodologia de Projeto deve prover ferramentas de validação e verificação [Baheti, 2011] de forma a permitir o emprego de normas e certificações. O desenvolvimento e utilização de simuladores e modelos estocásticos de análise [Clarke, 2011], bem como de sistemas de visualização capazes de permitir o monitoramento da evolução dinâmica dos elementos envolvidos, são importantes ferramenta de auxílio a validação e verificação de correteude dos CPS.

2.3 Aplicações

Muitas são as aplicações associadas atualmente a CPS's, dentre os sistemas atualmente presentes no cotidiano pode-se citar:

- Energia: sistemas de gerenciamento de distribuição e geração de energia inteligente [Raab, 2011] e [Li, 2010];
- Infraestrutura: construção de estruturas inteligentes, logística e transporte inteligentes [Maiden, 2005];
- Medicina: sistema de alta confiabilidade com telemedicina, e gerenciamento de rotas mais eficientes de transporte de socorro médico [Wang, 2010];

- Setor aéreo: sistemas de controle de voo, anti-colisão, sistema de informação para o piloto [Bianco, 2006] [Maiden, 2005];
- Militar: substancial aumento da eficácia das unidades militares, através de rede de veículos autônomos, logística de suprimentos [Wing, 2008] e [Lee, 2008];
- Comunicação: o rádio cognitivo pode se beneficiar enormemente de um consenso distribuído sobre a largura de banda disponível e de tecnologias de controle distribuídas
- [Zhang, 2013] e [Lin, 2008]; e Robótica: utilização de sistemas robóticos distribuídos, bem como telepresença;

2.4 Trabalhos Relacionados

Com base nos desafios associados com o desenvolvimento de CPS, este trabalho busca apresentar uma plataforma para o desenvolvimento de CPS que envolve uma metodologia de especificação utilizando o conceito de hiperambientes. A proposta tratará os aspectos relacionados com o fornecimento de abstrações semânticas capazes de permitir a descrição e integração de sistema híbridos e heterogêneos envolvendo cenários virtuais e reais. Busca-se desenvolver uma arquitetura capaz de integrar as diferentes informações circulantes de forma modular e baseada em componentes, com suporte a escalabilidade. A seguir apresenta-se alguns trabalhos relacionados, principalmente associados a arquiteturas para CPS, bem como uma visão geral dos protocolos para Redes de Sensores e Atuadores, os quais poderão ser utilizados no âmbito da definição de padrões e semânticas para CPS.

2.4.1 Arquiteturas para CPS

Web-of-Things: Dillon [2011] apresenta uma integração da Web-of-Things com CPS e uma arquitetura com implementação em um *framework* para CPS. O trabalho utiliza tecnologias e incrementa protocolos da Web atual para lidar com requisitos específicos do CPS. O termo *Web-of-things* [Guinard, 2011] refere-se a um avanço com relação a Internet-of-things, que trata de coisas *things* conectadas a internet sendo Web-of-things o

próximo nível o qual trataria de "coisas inteligentes" (*smart-things*). O framework consiste de cinco camadas, são elas:

1. **WoT-device** - fornece uma nível de abstração capaz de gerenciar os elementos físicos e os recursos. Cada elemento físico é modelado como um recurso WoT que possui um identificador universal, um nome, um estado que pode ser medido, calculado, e manipulado. Um evento representa um instantâneo do estado atual. Enquanto um evento é imutável, o estado interno de um recurso WoT pode ser modificado através de um único Universal Resource Identifier (URI). O elemento físico pode estar associado a recursos múltiplos, tais como os diferentes tipos de dados e sub-componentes.
2. **WoT Kernel** - fornece um *middleware* de baixo nível para a comunicação, programação e gestão de recursos WoT. A solicitação de recurso WoT será encaminhada diretamente para um Kernel WoT, que resolve URIs para avaliar os recursos e gerencia o mapeamento entre os URI de endereços de recursos diferentes. Neste sentido, o WoT Kernel pode ser construído a partir de um servidor Web com comunicação de recursos, renderização e processamento de HTTP. O Kernel do WoT também é responsável pela detecção e identificação de novas conexões / desconexões de dispositivos físicos e seus recursos associados. A descoberta de recursos é fundamental para a auto-configuração e plug-and-play do sistema.
3. **WoT Overlay** - fornece uma driven de aplicação, a rede sobre a infra-estrutura atual da Internet, como TCP/IP. O principal objetivo do WoT Overlay é lidar com o comportamento volátil, como latência, perda de dados, largura de banda e assim por diante, permitindo selecionar nós para caminhos com melhor desempenho.
4. **WoT Context** - apoia tanto o contexto ativo que controla o comportamento das aplicações do CPS quanto o contexto passivo que é de interesse para aplicações CPS e / ou usuários. O WoT Context produz contextos passivos representados em padrões da Web (por exemplo, CSV, XML, RDF, GeoML, etc).
5. **WoT API** fornece abstração na forma de interfaces que permitem que os desenvolvedores de aplicativos possam interagir com o WoT framework.

Service Oriented Architectures (SOA): Ribeiro [2008] descreve SOA como um paradigma emergente para a modelagem distribuída de sistemas, o bloco básico de construção de SOA consiste na abstração de serviço. Apesar de uma parte significativa da pesquisa em SOA ter seu foco em modelagem para indústrias em geral, existe uma convergência de fatores favoráveis, como baixo custo de dispositivos embarcados de alta performance, a expansão e baixo custo de redes baseadas em Ethernet, que estão tornando-o atraente para o estabelecimento de redes de dispositivos, sistemas heterogêneos são normalmente o alvo de aplicação de ambientes SOA. As principais características de SOA são descritos a seguir:

- **Autonomia:** não há dependências diretas entre os serviços.
- **Interoperabilidade:** é especificado no nível de interface omitindo detalhes desnecessários.
- **Independência de plataforma:** os serviços são descritos utilizando interoperabilidade baseada em formatos XML.
- **Encapsulamento:** proporcionar serviços auto-suficientes com funcionalidades que são expostas pelo usuário através de interfaces.
- **Disponibilidade:** os serviços podem ser publicados em registros públicos e disponibilizados para uso geral.

Em [M P Papazoglou, 2005] São apresentadas e identificadas as seguintes áreas de pesquisa para SOA: fundamentos de serviço, composição, gestão de serviços, projeto de serviços e desenvolvimento. Os principais desafios abertos, também descritos no trabalho incluem: arquiteturas de dinâmicas para reconfiguração em tempo de execução, descoberta de serviços, composição autônoma de serviços, auto-diagnóstico, entre outros.

Uma arquitetura para CPS é proposta por Yu [2012] utilizando SOA e é apresenta a seguir: A arquitetura do CPS é mostrada na figura 2.4. Essa arquitetura inclui camada física, camada de serviço e camada de aplicação. Na camada física conecta-se uma variedade de dispositivos físicos por diferentes protocolos de comunicação, tais como Wi-Fi, Blue-Tooth, ZigBee. As aplicações de controle usam mensagens, que são coletadas por

dispositivos físicos para produzir informações de controle. Controladores controlam dispositivos físicos de acordo com o controle de informação. Dispositivo Bundle pode ser fornecido por um *driver* também fornecido pelo sistema operacional e pode controlar diretamente dispositivos físicos. Pode-se usar o pacote de dispositivos para controlar um dispositivo físico ou coletar informações de dispositivo físico. Consumidores de serviços de acesso ou prestadores de serviços fornecem serviços por meio de agentes. A vantagem é de que, quando ocorre uma falha no serviço em que esta sendo utilizado, o agente pode redirecionar para outros serviços com a mesma função, resultando em uma melhor confiabilidade do serviço.

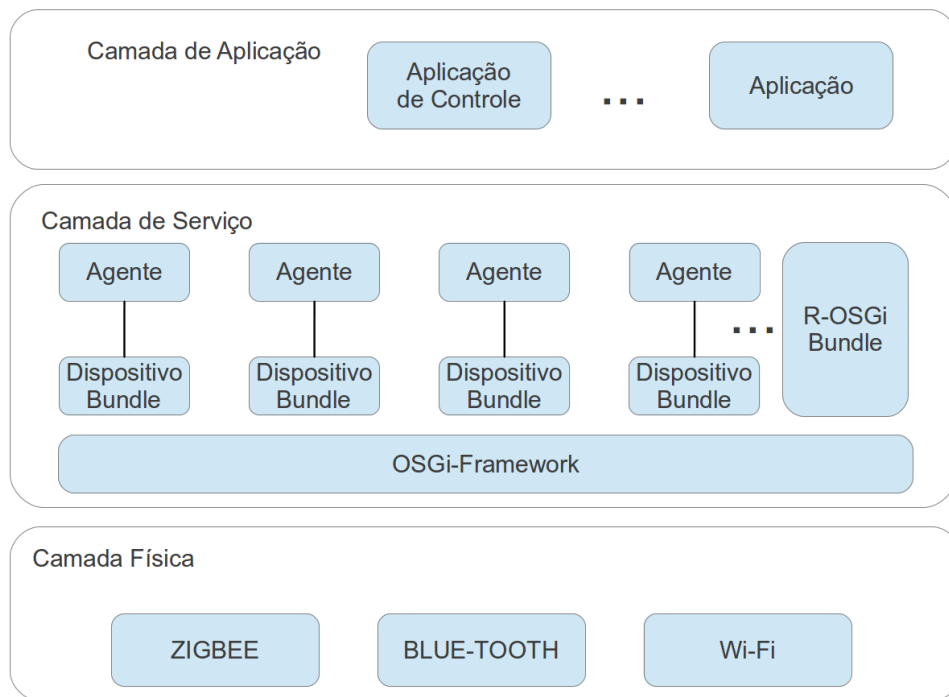


Figura 2.4: Arquitetura SOA para CPS.

Robotic Operation Systems (ROS): Quigley [2009] descreve que ROS foi projetado para atender a um específico conjunto de desafios encontrados no desenvolvimento de robôs de serviço em larga escala, mas a arquitetura resultante demonstra ser genérica quanto a aplicações, podendo talvez ser usada como uma arquitetura para CPS, sendo

este middleware utilizado como parte da proposta de arquitetura do Modelo Conceitual desta dissertação. Os objetivos gerais do ROS podem ser resumidos como:

- **Ponto a ponto:** Um sistema construído usando ROS consiste em uma série de processos, possivelmente em um número de diferentes hospedeiros, ligados em tempo de execução em uma topologia ponto-a-ponto. Apesar de quadros com base em um servidor central, também pode perceber os benefícios do multi-processo e design multi-hospedeiro, um servidor central de dados é problemático se os computadores estão conectados em uma rede heterogênea.
- **Base de Ferramentas:** Em um esforço para gerenciar a complexidade de ROS, este projeto optou por um micro-kernel, onde um grande número de pequenas ferramentas são usadas para criar e executar os vários componentes do ROS, ao invés de construir um desenvolvimento monolítico de execução.
- **Multi-lingual:** o ROS foi projetado para ser uma linguagem de programação neutra. Atualmente o ROS suporta quatro linguagens de programação diferentes, C++, Python, Octave, e LISP, com outras linguagens de programação em vários estágios de conclusão.
- **"thin" Fino:** Todos os drivers e desenvolvimentos de algoritmos podem ser colocados em bibliotecas independentes que não têm dependências do ROS. O sistema de compilação ROS realiza construções modulares dentro da árvore de código-fonte, e seu uso do CMake torna relativamente fácil de seguir essa característica "thin"-fina.
- **Livre e Open-Source:** O código-fonte completo de ROS está disponível publicamente.

Os conceitos fundamentais da implementação do ROS são os nós, mensagens, tópicos e serviços, os nós são processos que executam cálculos. ROS é projetado para ser modular em uma escala de granulação fina, um sistema ROS normalmente é composto de muitos nós. O ROS possui ferramentas de navegação e visualização de informações circulantes no sistema, um programa de visualização que utiliza uma arquitetura de encaixe: isso é feito no programa RVIZ, que é distribuído com ROS. Painéis de visualização podem ser instanciados dinamicamente para ver uma grande variedade de tipos de dados, tais

como imagens, nuvens de pontos, primitivas geométricas (tais como os resultados de reconhecimento de objeto), redenrização de robô com poses e trajetórias, etc. Plugins podem ser facilmente escritos para exibir mais tipos de dados. Uma arquitetura básica é apresentada na figura 2.5.

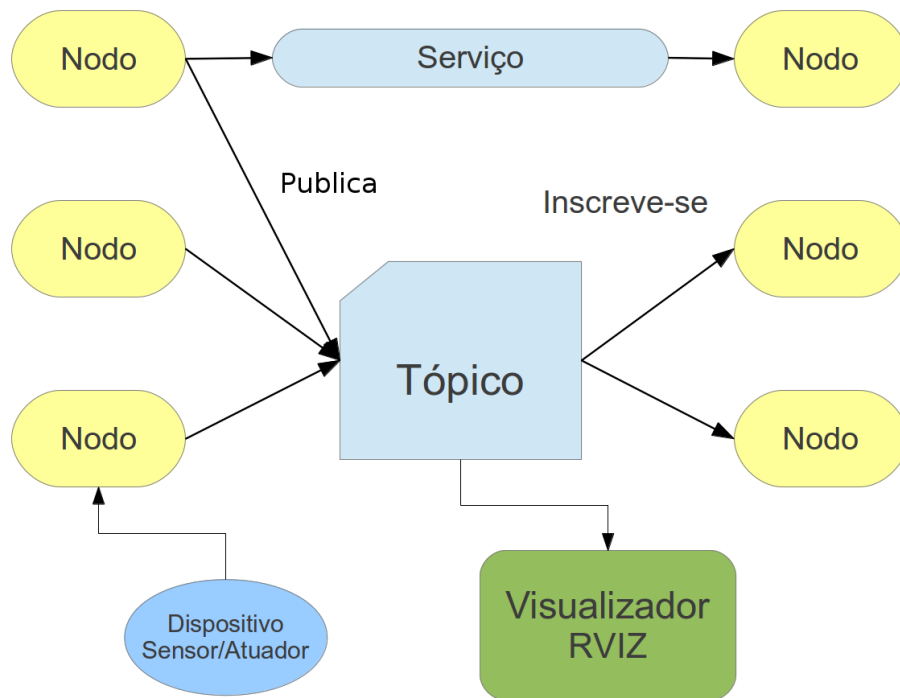


Figura 2.5: Arquitetura básica do ROS.

CPS bio-inspirado: Neste artigo Rammig [2008] apresenta uma tentativa de construir os altamente complexos e altamente sofisticados sistemas embarcados do futuro. Inspirações de colônias de formigas, do sistema hormonal e do sistema imunológico são utilizados como estudos de caso. Esta abordagem observa a existência de sistemas de grande sucesso e relativamente estáveis na forma de nossa biosfera. Assim, parece ser aconselhável tomar estas inspirações da conquista da natureza. Esta abordagem tornou-se bastante popular sob o termo "Sistemas de inspiração biológica" ou "computação biológica".

Sistemas computacionais tradicionais não oferecem grande flexibilidade, parece que este limite de flexibilidade biológica é muito mais amplo do que no artefactos técnicos

convencionais. A partir desta observação, não surpreende que um dos mais estáveis, mais robustos e mais adaptáveis artefatos técnicos complexos seja a internet. Na verdade, a internet segue princípios básicos de sistemas biológicos como o projeto distribuído, adiando decisões e ações para a fase operacional, a auto-organização, a redundância emergente, só para citar alguns deles. Algumas características comuns aos sistemas altamente complexos do futuro são: redes complexas e voláteis em que os componentes cooperaram, bem como, possivelmente, competem entre si; o controle descentralizado e componentes atuando de forma autônoma; um estado não observável do sistema global e, assim, apenas o local de conhecimento dos componentes; a otimização dos próprios benefícios que são a força motriz de um componente de cooperação; adaptar-se e aprender com as mudanças ambientais como uma capacidade universal dos componentes e a disponibilidade limitada de recursos combinada com a segurança e a segurança de requisitos. Todas as propriedades citadas estão presentes em sistemas biológicos. Por isso, inspirações neste domínio parecem ser interessantes. Sistemas biológicos parecem seguir estratégias ótimas (ou pelo menos perto do ideal), na presença de informações parciais ou mesmo confiáveis. São apresentados alguns dos princípios comuns entre CPS e Sistemas de inspiração biológica: construir sistemas que podem atuar de forma robusta perante o inesperado; construir sistemas seguindo uma abordagem rigorosa orientada ao componente; comunicação apenas por intercâmbio de dados puros e nunca negligenciar os chamados de propriedades funcionais. Estas características fomentam a tentativa desta abordagem em desenvolver uma arquitetura para CPS bio-inspirada.

PRET Broman [2013] apresenta uma infra-estrutura de precisão de tempo (PRET), consistindo de linguagens, compiladores e microarquitecturas, onde o tempo é explicitamente considerado. O problema geral é compilar ou sintetizar automaticamente partes cibernéticas de um modelo, de tal forma que o modelo simulado e o sistema real coincidam. A maior contribuição desta proposta é a propor um modelo capaz de garantir o correto comportamento de tempo.

Um protótipo de arquitetura para CPS é apresentado em [Tan, 2008] e [Teodora Sanislav,]. Este representa uma abstração do mundo físico real através de eventos digitais, governado por leis semânticas, evoluindo a arquitetura típica dos sis-

temas embarcados e alinhando-as às exigências tecnológicas atuais. A proposta envolve aspectos referentes a sistemas de referências de tempo global, abstrações da evolução de estados físicos através de eventos, confiabilidade das abstrações, gerenciamento da evolução através de métodos *publish/subscribe*, leis de controle e novas técnicas de roteamento e gerenciamento de dados advém desta semântica.

2.4.2 Protocolos, Semânticas e Ontologias para Rede de Sensores e Atuadores

Com o advento das redes de sensores e atuadores e sua integração em redes de dados como a web, muitas tem sido as propostas de organização, estruturação e normatização das informações circulantes. Diversos consórcios estão sendo criados visando o desenvolvimento e ampliação de protocolos para propiciar o registro, e recuperação de dados advindos de grids capazes de perceberem e atuarem autonomamente no ambiente. De forma a permitir abstrair e organizar diferentes níveis semânticos presentes nas informações circulantes em um CPS, foram resgatados da bibliografia os principais protocolos atualmente desenvolvidos para aplicações que envolvam Grids de Sensores e Atuadores. A seguir descrevem-se algumas das principais propostas encontradas na literatura, detalhando-se o padrão EEML o qual será utilizado neste trabalho.

O conceito do Sensor Grid foi definido pela primeira vez no Discovery Net project, onde foi feita uma distinção entre as "redes de sensores" e "grades de sensores". Um *Sensor Grid* integra redes de sensores sem fio para permitir em tempo real a coleta de dados, o compartilhamento de recursos computacionais, o armazenamento para o processamento de dados de sensores e de gestão. É uma tecnologia para a construção de infra-estrutura em larga escala, integrando sensores heterogêneos, dados e recursos computacionais implantados em uma área ampla, para realizar tarefas de vigilância complicadas, como monitoramento ambiental [Lim, 2005].

Outra proposta envolve o consórcio *Open Geospatial* [Botts, 2007]. Este consórcio produziu o padrão, denominado SensorML, o qual fornece um modelo e codificação XML para descrever sensores e processos de medição. O padrão pode ser usado para descrever uma ampla gama de sensores, incluindo plataformas móveis e estacionárias.

Outro consórcio o *World Wide Web Consortium (W3C)* atualmente desenvolve o *Semantic Sensor Network (SSN)* o qual constitui-se de uma proposta de ontologia capaz de modelar dispositivos como sensores, sistemas, processos e observações. A Rede Sensor fornece uma semântica capaz de permitir a representação expressiva de sensores, observações dos sensores e o conhecimento do meio ambiente. A ontologia SSN é codificada na *Web Ontology Language (OWL)* e esta sendo aceita pela comunidade de desenvolvimento de sensores em larga escala. O SSN está sendo usado por várias organizações (a academia, o governo e a indústria), para a melhoria da gestão dos dados de grids de sensores na web, envolvendo anotação, integração, publicação e pesquisa [Compton, 2012] e [Neuhaus, 2009].

O *Extended Environments Markup Language (EEML)* é outro protocolo para gerenciamento de informações advindas de elementos reais. Ele foi desenvolvido pela Haque Design Research ltd, com o objetivo de compartilhar dados ambientais remotos em tempo real, usando diversas tecnologias como Flash, Processing, Arduino ou qualquer outro aplicativo que analisa XML. O EEML é compatível com o formato estabelecido pela indústria da construção civil, Industry Foundation Classes [Classes, 2013].

O EEML é baseado em uma descrição XML, permitindo a descrição de mapas neste formato capazes de permitir a troca de dados entre diferentes serviços e modelos de dados em um pacote de software com Industry Foundation Classes (*IFC*) [Classes, 2013]. É um formato de arquivo baseado em objeto com um modelo para facilitar a interoperabilidade, é um formato comumente usado para a modelagem de informações na construção civil. Essa característica possibilita que as principais marcas de software CADs como AutoCAD Architecture, Revit Build 2008 e Tekla Structures, que utilizam o IFC, possam ser compatíveis com aplicações desenvolvidas usando EEML [Language, 2013]. Arquitetos e engenheiros podem usar essa característica para, por exemplo, informar a verificação e validação de simulações de edifícios a partir de seus elementos, componentes ou sistemas. O EEML oferece a possibilidade de acesso a dados do mundo real, permitindo a melhoria do projeto e simulação do sistema em tempo real.

O EEML pode ser aplicado em obras civis na verificação e validação de simulações de edifícios no que tange seus elementos estruturais, componentes ou sistemas. O padrão oferece a possibilidade de acesso online a dados da planta, permitindo a melhoria do projeto

e simulação do sistema em tempo real. Um exemplo bem sucedido de utilização do padrão EEML é visto no serviço *Xively* [Xively, 2013], que provê conexão e compartilhamento de dados de dispositivos, através por exemplo do *open hardware Arduino* entre outros na web, com geolocalização e visualização de dados em tempo real.

De forma a poder tratar grandezas físicas associadas aos elementos a serem descritos, o EEML apresenta as cinco categorias de unidades de medidas, advindas das classificações IFC, referentes ao estado possíveis de cada entidade, são elas [des Poids et Mesures, 2006]:

- Categoria de Unidades Básicas SI *Sistema Internacional de Unidades*: esta categoria inclui um conjunto de grandezas físicas cujas unidades de medida estão relacionadas 7 unidades de medida básica do Sistema Internacional de Unidades (SI), sendo elas: m - metros de comprimento, kg - de massa, s - segundo para o tempo, A - ampère para a corrente elétrica, K - kelvin para a temperatura, cd - candela de intensidade luminosa e mol - mole para a quantidade de substância;
- Categoria de Unidades Derivadas da SI: uma lista maior de unidades de medidas que são derivadas a partir das unidades de base, por exemplo, newtons, ohms, hertz;
- Categoria de Unidades Básicas Convertidas: inclui conjunto de unidades de medidas resultantes da conversão de unidades básicas SI, por exemplo, nesta categoria encontra-se a unidade polegadas, resultante da conversão de metros;
- Categoria de Unidades Derivadas: esta categoria engloba unidades de medida de grandezas físicas derivadas de combinações de unidades básicas SI e conversão, por exemplo, "milhas por hora";
- Categoria de Unidades Dependente do Contexto: permite a definição de qualquer unidade de medida. Por exemplo, "respostas por convite", "batimentos cardíacos por minuto."

Os protocolos apresentados, contém em geral, as mesmas características, com enfoques ligeiramente diferentes quanto à aplicação final e por serem desenvolvidos por diferentes iniciativas, tais protocolos poderão ser utilizados em grids de sensores e atuadores. Após analisar o grupo de protocolos, optou-se por utilizar o EEML, por apresentar um esquema

XML de descrição focado em ambientes contendo dispositivos, diferente dos demais que focam principalmente em dispositivos em rede. Esta característica converge com o modelo conceitual proposto para CPS, descrito no próximo capítulo.

Simplificado, porém robusto, sua estrutura de descrição se mostra adequada para o acesso as informações provenientes dos diferentes elementos heterogêneos presentes em um CPS, contando ainda com outro diferencial interessante, que é a classificação das unidades básicas, derivadas e de contexto, como as medidas métricas, elétricas entre outras. Estas categorias permite a descrição de unidade de medida dos fenômenos físicos a serem monitorados, supervisionados e controlados no CPS.

2.5 Considerações Finais

Com base nas definições e requisitos para CPS, busca-se neste trabalho explorar a possibilidade de tratar diferentes níveis semânticos em uma plataforma para desenvolvimento de CPS capaz de arranjar sistemas híbridos, físicos-virtuais de alta escalabilidade. Para tal foram tratados os aspectos de definição, desafios e principais aplicações atuais de CPS. O capítulo também apresentou uma síntese dos principais trabalhos relacionados com o tema, com foco nas principais arquiteturas para CPS e protocolos capazes de acessar a informações circulantes em grids de sensores e atuadores presentes em ambientes físicos. Em especial foi detalhado o protocolo EEML o qual, por ter adotado a normatização IFC, começa a ser utilizado por um grande número de empresas desenvolvedoras de aplicações e soluções de engenharia e projeto.

No próximo capítulo será apresentado o modelo conceitual adotado neste trabalho, capaz de descrever sistemas físico-cibernéticos heterogêneos de alta escalabilidade.

Capítulo 3

UM MODELO CONCEITUAL PARA CPS E HIPERAMBIENTES

Este capítulo aborda a descrição do Modelo Conceitual para CPS, a descrição do conceito de hiperambientes e logo após, faz-se uma análise e comparações entre hiperambientes / hipertextos e hiperambientes / CPS. Por último uma modelagem para CPS é feita a seguir, utilizando os modelos conceituais apresentados.

A identificação dos possíveis elementos e suas interações presentes em um CPS constituem-se um primeiro passo no desenvolvimento de um framework o desenvolvimento destes sistemas.

3.1 O Modelo Conceitual

Neste trabalho propõe-se um modelo conceitual para descrição de CPS, baseado no trabalho proposto por Serbanati [2011]. A Figura 3.1 apresenta o modelo a ser adotado. Este modelo será detalhado e utilizado nesta dissertação por ser um projeto amadurecido e com documentação detalhada, apresentando muitas das característica identificadas no capítulo 2, as quais relacionam-se com as diferentes interações entre os elementos tanto reais como virtuais integrando ambientes mistos. O modelo prevê a existência de diferentes entidades e seus relacionamentos, os quais são descritos a seguir.

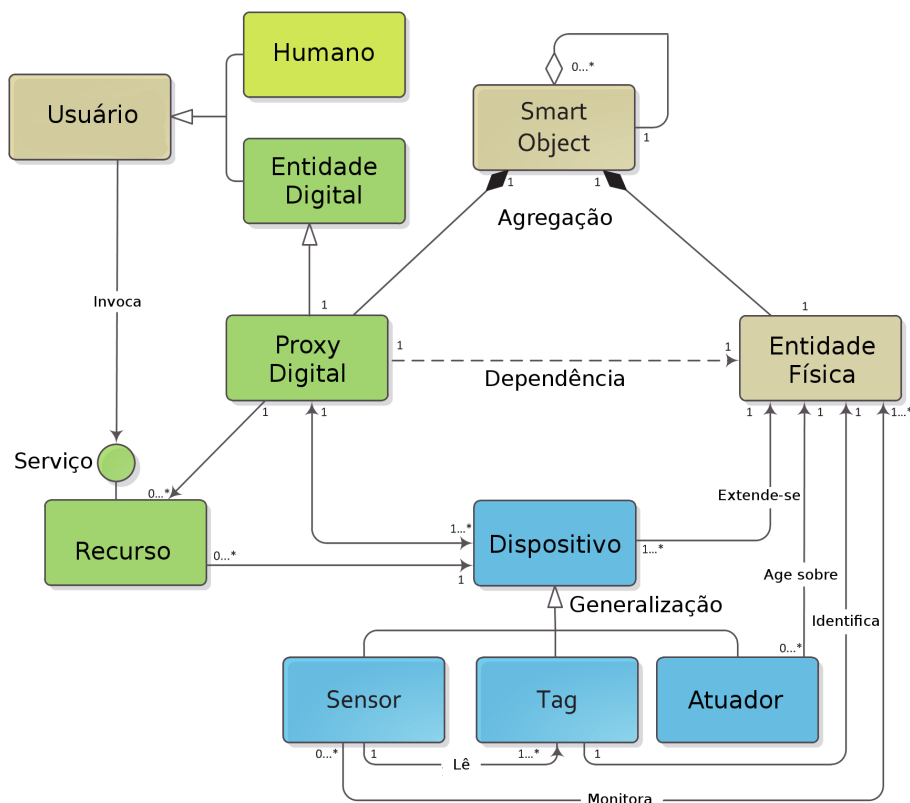


Figura 3.1: Modelo Conceitual da Arquitetura, demonstra no diagrama UML o relacionamento entre os seus elementos.

3.1.1 As Entidades do Modelo

O modelo apresenta um conjunto de entidades básicas que descrevem os principais componentes do sistema. São elas:

- **Usuários:** Pessoa ou um agente de software com objetivos a serem realizados. Para alcançar estes objetivos o usuário fará uso do modelo conceitual e sua implementação para estabelecer diferentes níveis de interação com os diferentes elementos (físicos/virtuais) presentes;
- **Entidade Digital:** Em um mundo digital entidades digitais são entidades de software que podem ser agentes que têm objetivos autônomos, pode ser serviços ou entradas de dados coerentes. Algumas entidades digitais podem também interagir com outras entidades digitais ou com usuários, a fim de cumprir seu objetivo;
- **Entidade Física:** A entidade física é um elemento discreto, parte identificável do ambiente físico que pode ser de interesse para o usuário para a realização de seu objetivo, Entidades físicas podem representar qualquer objeto ou ambiente, desde seres humanos, animais, carros, máquinas até plantas industriais, passando por insumos e itens do estoque de uma loja ou cadeia logística;
- **Proxy Digital:** Relação de endereços dos componentes existentes;
- **Smart Object:** Embora existam diferentes definições de objetos inteligentes na literatura [Kortuem, 2009] define-se um *smartobject* como a extensão de uma entidade física com o seu *Proxy Digital* associado; e
- **Dispositivos:** São componentes com a função de integrar *entidade física* com o ambiente virtual, são os responsáveis pelo acesso às informações das grandezas físicas associadas a estas entidades, onde este acesso se traduz em ações de percepção e atuação no mundo físico. Estabelece-se três tipos de dispositivos:
 - Sensor - Os sensores podem fornecer informações sobre a entidade física que monitoram. As informações neste contexto associam diferentes grandezas físicas a uma determinada entidade física unicamente identificada. A identidade pode ser inerentemente ligada à entidade, tal como no caso de dispositivos

embarcados, ou pode ser derivada a partir da observação das características desta ou presente explicitamente em rótulo fixado nesta (*tag*). Sensores poderão estar encapsulados por entidades físicas que os embarcam, por exemplo um sensor de carga de bateria de um robô, ou não estar vinculado a nenhuma entidade, como por exemplo uma câmera de monitoramento, neste caso o sensor seria ele próprio uma entidade física a ser representada. Neste caso os sensores tornam-se software complexo que normalmente observam um ambiente específico em que se pode identificar e monitorar entidades físicas, através da utilização de algoritmos e técnicas de rastreamento. O exemplo mais comum desta categoria são sistemas de reconhecimento facial que utilizam o espectro óptico. Sensores também podem ser leitores;

- Tag - É usada por dispositivos sensores especializados geralmente chamados de leitores, a fim de apoiar o processo de identificação. Este processo pode ser óptico, como no caso dos códigos de barras e Qrcode, ou pode ser por rádio frequência, como no caso dos sistemas de reconhecimento da placa de carro de micro-ondas e RFID;
- Atuadores - Atuadores modificam grandezas associadas ao estado físico de uma entidade física. Atuadores pode mover (transladar, girar ...) simples entidades físicas, ativar/desativar suas propriedades, etc..

3.1.2 Recursos

São componentes identificáveis digitalmente que implementam diferentes capacidades, estando associados tanto a *entidades digitais* como *entidades físicas*. Mais do que um recurso pode ser associada a um *proxy digital* e, assim, a um *smartobject*. A seguir as cinco classes gerais de recursos previstas no modelo:

- recuperação de propriedades físicas de uma entidade física percebidas através sensores;
- modificação das propriedades físicas de uma entidade física realizada através de atuadores;

- recuperação de propriedades digitais do *Proxy Digital* associado;
- modificação das propriedades digitais do *Proxy Digital* associado;
- uso de hardware complexo ou serviços de software fornecidas pelo *smartobject* associado.

3.1.3 Relacionamento entre as Entidades do Modelo

Neste modelo destaca-se o *smartobject* como elemento chave. Este pode relacionar-se através de associação por agregação com outros *smartobjects* com o objetivo de formar um novo objeto composto. Outro relacionamento fundamental para o *smartobject* é o de associação por composição, onde este é formado por um proxy digital e uma entidade física, um *smartobject* não pode existir sem um destes dois elementos. No mesmo diagrama visualiza-se também a questão da cardinalidade ou multiplicidade, no caso do *smartobject*, este relaciona-se um para vários *smartobject* e um para um com o seu próprio proxy digital e entidade física. O proxy digital concentra grande parte dos relacionamentos entre os elementos do diagrama, destacando-se o relacionamento com o elemento *Dispositivo*, o qual herda características dos sensores e atuadores associados a ele. Seu relacionamento com o proxy se dá através um único proxy, no qual poderá ter diversos dispositivos associados. O relacionamento das principais entidades do modelo conceitual, demais entidades do diagrama da figura 3.1, seguem as normas do diagrama de classe UML, que descreve seu comportamento relacional entre entidades.

3.2 Hiperambientes

O conceito de hiperambientes relaciona-se com a possibilidade de congregarem em um único locus informações advindas de elementos reais, presentes na natureza, bem como provenientes de instâncias e simulações meramente virtuais sem necessidade de existência real [Carvalho, 2011]. Os hiperambientes materializam-se em ambientes multi tecnológicos, que agregam objetos reais/virtuais cujas características (informações) podem ser acessadas, tratadas e modificadas, envolvendo diferentes paradigmas tecnológicos, tais como ambientes robóticos, ubíquos, ambientes virtuais e mistos.

Neste trabalho o conceito de hiperambiente será resgatado de forma a incorporar as entidades e relações presentes no modelo conceitual apresentado de forma a possibilitar utiliza-lo como modelo de descrição de CPS.

O conceito de hiperambiente surge na tentativa de resgatar as características atribuídas por Pierre Levy [2010] aos hipertextos, sejam elas:

- Princípio de metamorfose. Constante construção e renegociação;
- Heterogeneidade. As informações, nós e comunicação são de diferentes formas;
- Princípio de multiplicidade e de encaixe das escalas. Qualquer nó quando analisado, pode revelar-se como sendo composto por toda uma rede, e assim por diante;
- Princípio de exterioridade. Seu crescimento e modificação dependem de um exterior indeterminado, um hipertexto pode conter infinitos hipertextos, não permitindo definir sua real extensão observando apenas uma camada de hipertexto; e
- Topologia. O funcionamento do sistema dá-se por proximidade e vizinhança, não necessitando de entidades centrais para controle e sequenciamento.

Ao analisar-se as necessidades de um CPS, pode-se perceber requisitos semelhantes ao da World Wide Web, no que tange as características de hipertextos citadas por [Lévy, 2010]. Tais características diferem apenas na utilização de algumas nomenclaturas e termos [Ji Eun Kim, 2008], [Kaiyu Wan K.L., 2010], [Lee, 2013] e [Shi, 2011], a seguir uma tabela demonstra esse comparativo:

	Hipertexto	CPS
1	Princípio da metamorfose	Um CPS é dinâmico, suas entidades podem mudar afetando o cenário inicial.
2	Heterogeneidade	Normalmente CPSs consistem de entidades não-homogêneas.
3	Princípio de multiplicidade e de encaixe das escalas	Escalabilidade é essencial para o CPS, por tratar um grande número de entidades em uma rede.
4	Princípio de exterioridade	Uma entidade em CPS pode depender de diversos nós externos indeterminados.
5	Topologia	CPS é composto por nodos descentralizados.

Tabela de Comparação entre características do Hipertexto e CPS.

O hiperambiente estabelece-se como este local onde as diferentes informações (ou subconjunto) circulantes no sistema podem ser recuperadas. Da mesma forma que em hipertextos, de forma não linear por seus usuários.

Em analogia ao hipertexto, hiperambientes podem ser criados, editados e visualizados, permitindo o acesso não linear, leitura, escrita e tratamento das informações de proxy digitais, bem como sua representação. No hiperambiente seus membros poderão inter-relacionar-se de forma a atender determinados objetivos, provocando a emersão de inteligência no sistema.

Uma configuração possível de hiperambiente entre diversas é demonstrada na figura 3.2, tendo como topo da hierarquia o próprio hiperambiente, composto por vários ambientes, que podem também ser hiperambientes, contendo *smart objects* (entidades digitais), dispositivos e recursos, podendo relacionarem-se uns com os outros.

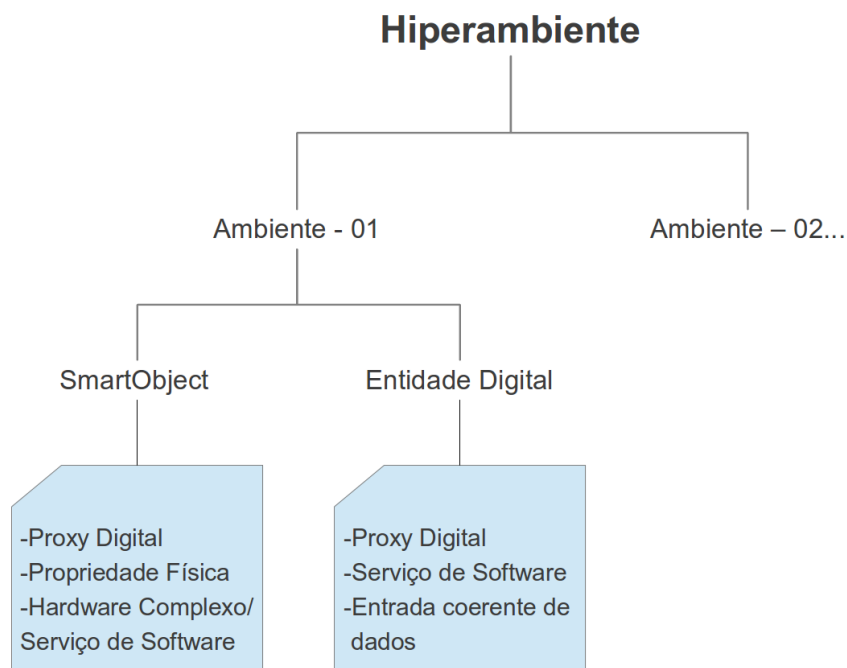


Figura 3.2: Ambiente Exemplo

3.3 Modelando um Sistema Físico Cibernético

Com base no modelo conceitual apresentado e na descrição do padrão hiperambiente é possível descrever um sistema compostos por entidades físicas e virtuais. O cenário a ser modelado é composto dos seguintes elementos:

- Câmera de vídeo - Capaz de capturar imagens do ambiente.
- Cadeira Inteligente - Objeto físico inanimado, contém um sensor de pressão embarcado, capaz de perceber a presença de Humanos ou objetos utilizando o assento.
- Cafeteira - Dispositivo eletrônico com o objetivo de fazer café, mas incapaz de controlar seu próprio acionamento.
- Mesa - Objeto físico inanimado, incapaz de perceber o ambiente em que se encontra.
- Lâmpada - Dispositivo físico com capacidade de iluminar o ambiente.
- Sensor de luminosidade - fornece a rede informação sobre a quantidade de luz no ambiente.

Os elementos do hiperambiente De forma a melhor descrever o padrão hiperambiente de descrição do modelo conceitual apresentado, descreve-se a seguir os diferentes elementos que compõem o cenário apresentado na figura 3.3.

Elementos Básicos: De forma a fornecer a semântica capaz de implementar o modelo conceitual de um CPS, o hiperambiente deve conter os seguintes elementos básicos:

id: Campo de identificação da entidade associada ao proxy;

tipo: Determina o tipo da entidade como *Entidade Digital* ou *smartobject*

propriedades: características que não variam no tempo *recurso leitura:* recurso que fornece esta característica (id de quem fornece, nome do recurso);

estados: características que variam no tempo, *recurso escrita:* recurso que fornece esta característica (id-de-quem-fornece, nome do recurso), frequência-real (frequência de leitura no mundo real), frequência de leitura no mundo virtual *recurso* que muda esta característica (id-de-quem-muda, nome do recurso),

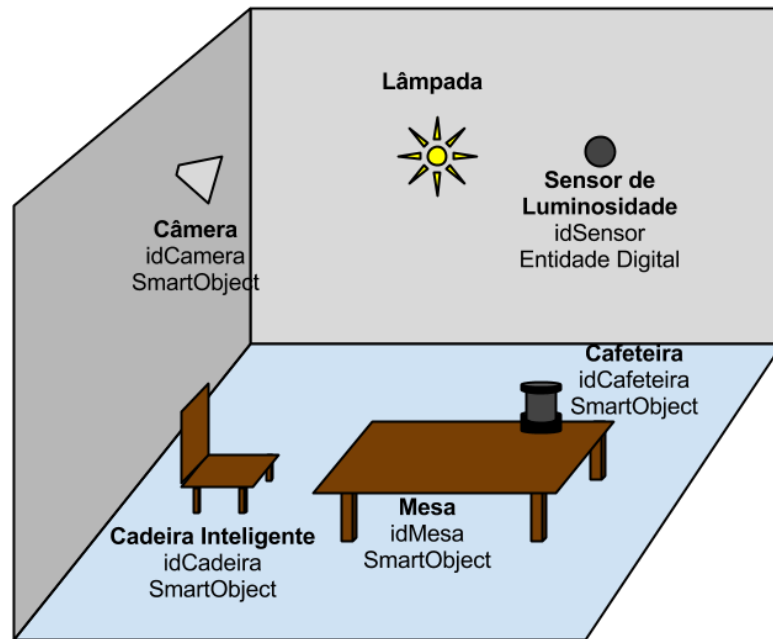


Figura 3.3: Ambiente físico exemplo.

Tendo sido apresentados os *elementos básicos* do hiperambiente, segue-se a descrição do cenário exemplo.

Este cenário apresenta a *mesa* que é um simples objeto que não possui nenhum tipo de sensor ou atuador. Entretanto, tendo em vista a câmera de vídeo e sua disponibilidade para rastreamento de objetos ¹, a mesa pode ser considerada uma entidade física e em função do seu rastreamento ter um identificador único, conduzindo a um proxy digital. Desta forma teríamos o seguinte *smart object* descrito da seguinte forma:

- proxy digital: `idDaMesa`
- tipo: `smart object`
- estado1: `x, y (R: servico(<idControle>, rastreio), W()10,100)`

Onde o estado associado a posição da mesa é lido (R) onde a posição x,y da mesa é lida pelo serviço rastreamento disponibilizado pelo elemento de id <idControle>. Este serviço é amostrado a uma taxa de 10Hz. Nenhum serviço de atuação estará disponível para

¹Admitimos que a câmera possui implementado recurso de identificação e rastreamento.

atualizar a posição da mesa. Entretanto, o estado x,y da mesa pode ser emulado pelo CPS ² a uma taxa de 100Hz³.

Também o cenário apresenta uma cadeira a qual possui um sensor que fornece o seu estado (ocupada/vazia). Da mesma forma que a mesa, a cadeira pode ter também a sua posição rastreada pela câmera. Assim, este *smart object* teria a seguinte descrição:

- proxy digital:<idCadeira>
- tipo: smart object
- estado1: vazia (R: servico(<idCadeira>, pressaoCadeira), W:(),W()10,100)
- estado2: x, y (R: servico(<idControle>, rastreo),Fr:(), W()W()10,100)
- servico1: pressaoCadeira

Percebe-se que a cadeira possui um dispositivo do tipo *sensor* o qual está embarcado nela, fornecendo o recurso `pressaoCadeira`.

Um terceiro tipo de entidade física presente neste cenário são os equipamentos cafeteira e lâmpada. Ambos possuem informações referentes a sua localização, também fornecidas pela câmera e relacionadas com o seu estado de ligado ou desligado. No hiperambiente, estes elementos teriam a seguinte descrição ⁴:

- proxy digital:<idCafeteira>
- tipo: smart object
- estado1: ON (R: servico(<idCafeteira>, estadoCafeteira), W:(<idCafeteira>, controlaCafeteira),
- estado2: x, y (R: servico(<idControle>, rastreo), W()W()10,100)

²No proximo capitulo será apresentado a plataforma de desenvolvimento de um CPS, onde um simulador, emulará estados contínuos de entidades cujas grandezas são amostradas de forma discreta

³Admite-se que não existe nem entidade física, nem entidade virtual capaz de modificar este estado

⁴Apresenta-se apenas a cafeteira, uma vez que a lâmpada tem descrição análoga

- `servico1: estadoCafeteira`

Os equipamentos disponibilizam serviços associados à percepção do seu estado. Também percebe-se que o estado da cafeteria quanto a ligada/desligada é modificado em função de um recurso. Neste caso o recurso advém de um processo de tomada de decisão encapsulado em uma entidade virtual a ser descrita.

A câmera é um outro tipo de entidade. Por se tratar de um dispositivo sensor com uma localização física que pode ser importante no hiperambiente, optou-se neste exemplo por tratá-la como também um *smart object*. Assim sua descrição seria a seguinte:

- `proxy digital:<idCamera>`
- `tipo: smart object`
- `estado1: ON (R: servico(<idCamera>, Fr:(),estadoCamera), W: servico(<idControla>, controlaCamera),Fv:()`
- `estado2: x,y (R () W()10,100)`
- `servico1: estadoCamera`

A câmera poderia localizar a si própria no hiperambiente ⁵.

Um conjunto de serviços associados a entidades virtuais controlam os diferentes dispositivos, realizando algum tipo de computação para a tomada de decisão. Estes recursos realizam a leitura e escrita de informações sem qualquer vínculo com o ambiente físico, sendo considerados elementos de processamento, são eles:

- `proxy digital: <idControle>`
- `tipo: entidade virtual`
- `servico1: controlaCamera`
- `servico2: controlaCafeteira`
- `servico3: controlalampada`

⁵Para este exemplo não utilizaremos este estado no hiperambiente.

- `servico 4: rastreio`

Um tipo específico de sensor, o de luminosidade, em cujo contexto a sua localização, bem como qualquer outra grandeza física é irrelevante à aplicação do CPS, considera-se também como um mero recurso virtual, assim:

- `proxy digital:<idSensor>`
- `tipo: entidade virtual`
- `servico1: nivelluminosidade`

Desta forma os elementos do CPS vão sendo descritos e compondo o hiperambiente resultante deste cenário.

Estrutura semântica utilizando EEML: Para a Estrutura Semântica, é utilizado um esquema XML, que nesse caso é o EEML. O EEML preverá os campos necessários para a descrição apresentada anteriormente. O campo *enviromemnt* será utilizado para a descrição de cada elemento, uma vez que *smartobject* encapsulam os objetos físicos com suas as propriedades e recursos podendo ser considerado neste caso como um tipo de micro-ambiente. Há também a possibilidade de modificar-se o esquema EEML, com a troca do termo *enviromemnt* para o tipo apropriado, como consequência o EEML gerado seria uma versão fora do padrão estabelecido pelos desenvolvedores do EEML. A identificação do *enviromemnt/smartobject*, se dá pelo campo *"id"*, que equivale ao proxy digital do modelo conceitual acompanhado de outro campo *"Title"* que serve como forma de identificação e pesquisa de *smartobject* no sistema. O campo *"Location"* contém subcampos, que guardam as características de geolocalização, são eles: O campo *"domain"* determina o domínio da entidade, se este é físico ou cibernético; o *"disposition"*, determina se a entidade é móvel ou fixa e um conjunto de três campos que compõe a geolocalização no globo terrestre, em *"lat"*-latitude, *"lon"*-longitude e *"ele"*-elevação. Após seguem subgrupos de campos que podem tratar de recursos ou de estados da entidade. Esta definição é feita pelo campo *Tag* que é classificador, podendo determinar a entidade como *smartobject* ou *entidade digital*. O campo *"current-valuer"* refere-se respectivamente ao valor composto por subcampos tempo, valor, frequência real e frequência virtual; *"Unit*

symbol" classificação do tipo de dado da propriedade da entidade ou *Dispositivos*, estes podem corresponder como serviços a outros *smartobject*. Com esse arranjo pode-se manter o padrão original do EEML, o que traz benefícios de compatibilidade ao Sistema Toogle.

A figura 3.4 mostra um exemplo de ambiente modelado em EEML e a figura 3.5 mostra o formato efetivo do smart object cadeira, que irá compor o arquivo XML.

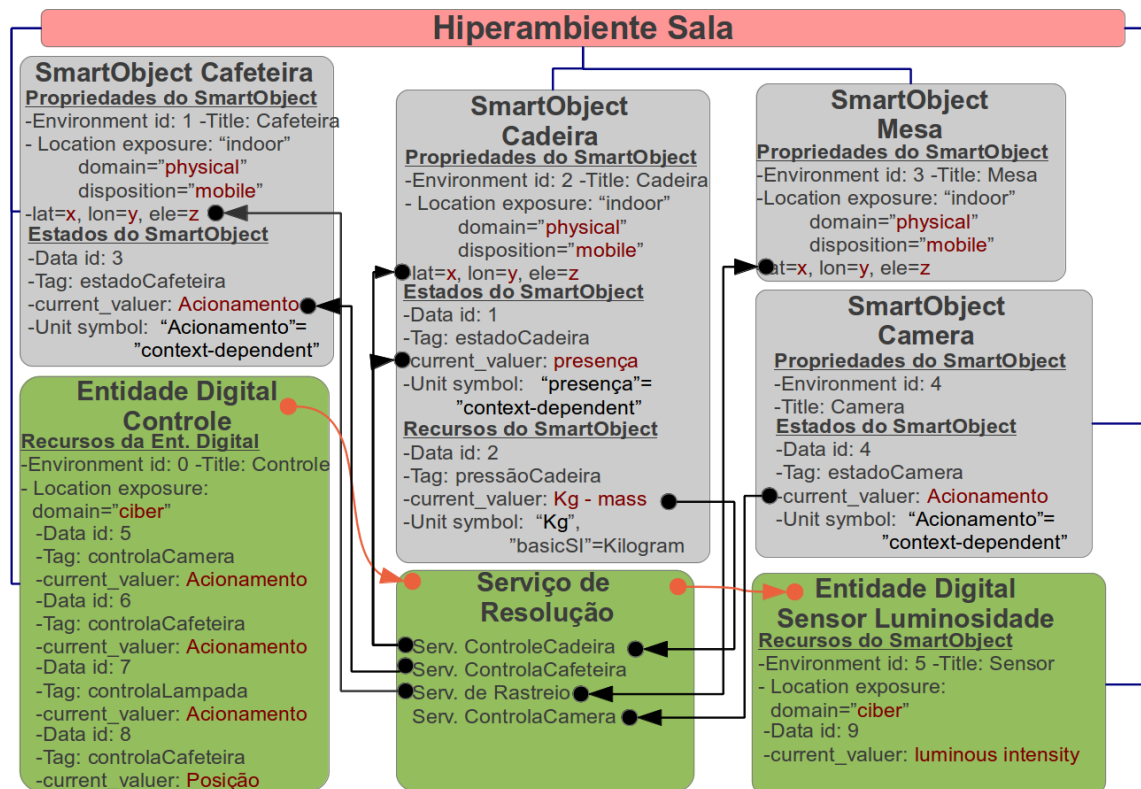


Figura 3.4: Exemplo do ambiente proposto em EEML

3.4 Considerações Finais

Neste capítulo foi apresentado o modelo conceitual proposto por Serbanati [2011]. Este modelo conceitual é utilizado também pelo padrão de descrição de CPS adotado neste trabalho, o hiperambiente. Este padrão foi adequado para a utilização na descrição de Sistemas Físicos Cibernéticos. De posse do modelo conceitual e do hiperambiente, o próximo capítulo apresenta a plataforma Toogle para desenvolvimento de CPS.

environment id	2
title=	SmartObject Cadeira
location=	domain="physical"
	disposition="mobile"
	lat=x, lon=y, ele=z,"Fr=28""Fv=30"
data id=	"1"
tag=	estadoCadeira
current_value=	At="2012-11-10T20:14:23"<on>"Fr=28""Fv=30"
Unit symbol=	"context-dependent"
data id=	"2"
tag=	pressãoCadeira
current_value=	At="2012-11-10T20:14:23"<Kg>"Fr=28""Fv=30"
Unit symbol=	"basicSI"=Kilogram

Figura 3.5: Exemplo do smart object cadeira em EEML

Capítulo 4

TOOGLE: UMA PLATAFORMA DE DESENVOLVIMENTO DE CPS

Este capítulo visa apresentar uma plataforma para o desenvolvimento de CPS, seus requisitos e descrição de seus módulos.

Tendo sido apresentado o modelo conceitual, bem como o padrão hipertexto de descrição de CPS, este capítulo apresenta uma plataforma para o desenvolvimento de Sistemas Físico-Cibernéticos. A plataforma busca atender aos seguintes requisitos:

- Edição e Visualização. A Toogle deverá conter um motor de busca e edição capaz de permitir a criação e acesso a *hiperambientes*, os quais descrevem um determinado projeto de CPS;
- Abstração semântica e modularidade. A plataforma deverá ser capaz de implementar as diferentes entidades previstas no modelo conceitual de um CPS, além de permitir tratar de forma modular e com semântica adequada as relações entre seus recursos conforme previsto no padrão hiperambiente. Informações de diversas naturezas, distribuídas e dinâmicas serão tratadas em diferentes categorias conforme semântica IFC;
- Heterogeneidade. A plataforma deverá dar suporte a grids de sensores e atuadores heterogêneos, de diferentes fabricantes, para tal será utilizado o middleware ROS (Robotic Operational System) como elemento integrador;
- Verificação, Validação e Simulação. A plataforma deverá propiciar o acesso e a visualização das informações relacionadas aos diferentes fenômenos que acontecem junto a suas entidades, auxiliando na verificação e validação do correto funcionamento do CPS. Para tal uma interface de visualização em Blender será disponibilizada [Blender Foundation, 2012]. Também o sistema deverá portar simulador dinâmico capaz de simular grandezas físicas associadas a *entidades físicas* presentes no mundo real, utilizando-se para tal o módulo Bullet [Bullet, 2012].

Tais requisitos serão atendidos através da plataforma a qual permitirá através de hiperambientes o desenvolvimento de CPS. A Toogle possibilitará a edição e navegação em hiperambientes. Através dela será possível criar um hiperambiente completo, configurando-se desde os recursos e serviços para acesso a informação, até os níveis mais altos de interação humano-computador, como a interface de visualização e navegação. A figura 4.1 apresenta a plataforma Toogle com um exemplo de Hiperambiente integrando

diferentes *smartobjects*, sua simulação e visualização. Percebe-se que a plataforma apresenta módulos principais: *i.* Toogle Dispositivo (configuração de dispositivos); *ii.* Toogle Middleware (criação e gerenciamento de *smart-objects*); *iii.* Toogle-Editor (criação e edição de hiperambientes); e *iv.* Toogle-Navegador (navegação de hiperambientes).

A seguir serão apresentada a plataforma através dos seus módulos.

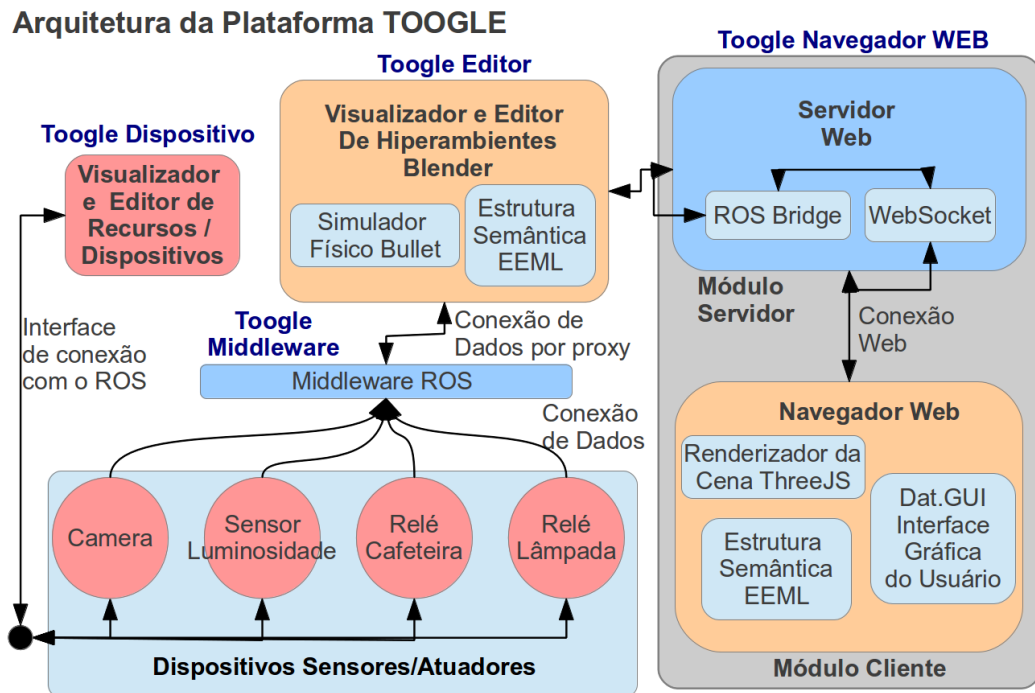


Figura 4.1: Sistema Toogle e um exemplo de hiperambiente

4.1 Toogle Dispositivo

O módulo Toogle-dispositivo fornece um nível de abstração capaz de gerenciar os *dispositivos* de um CPS [Núñez, 2013]. O módulo permite a criação desta entidade no sistema, bem como a sua configuração. Através de uma interface gráfica, o usuário é capaz de adicionar dispositivos sensores e atuadores, como por exemplo processadores Arduinos, câmeras, centrais inerciais, dentre outros ao hiperambiente criado, configurando as grandezas físicas a serem monitoradas dos diferentes *smartobjects* do sistema.

O módulo foi desenvolvido utilizando a linguagem de programação C++ através do framework para desenvolvimento de aplicações multi-plataforma QT Designer, o qual é amplamente utilizado para o desenvolvimento de aplicativos com interface gráfica de usuário. O QT Designer, assim como todas as tecnologias utilizadas neste trabalho, é uma ferramenta open-source e o desenvolvimento do Toogle-dispositivo, a fim de manter compatibilidade com o ROS, foi realizado sobre a plataforma Linux Ubuntu. A tela do programa pode ser observada na figura 4.2, próximo ao topo da tela encontram-se as guias com os *dispositivos* adicionados ao sistema, estando selecionada a guia arduino0, que mostra a configuração e visualização em tempo real de todas as propriedades do nó selecionado. Ao adicionar novos dispositivos, o usuário começa a criar o hiperambiente, configurando-o de acordo com a necessidade. Cada *dispositivo* adicionado, cria uma janela na interface do programa com as opções de configuração próprias a ele.

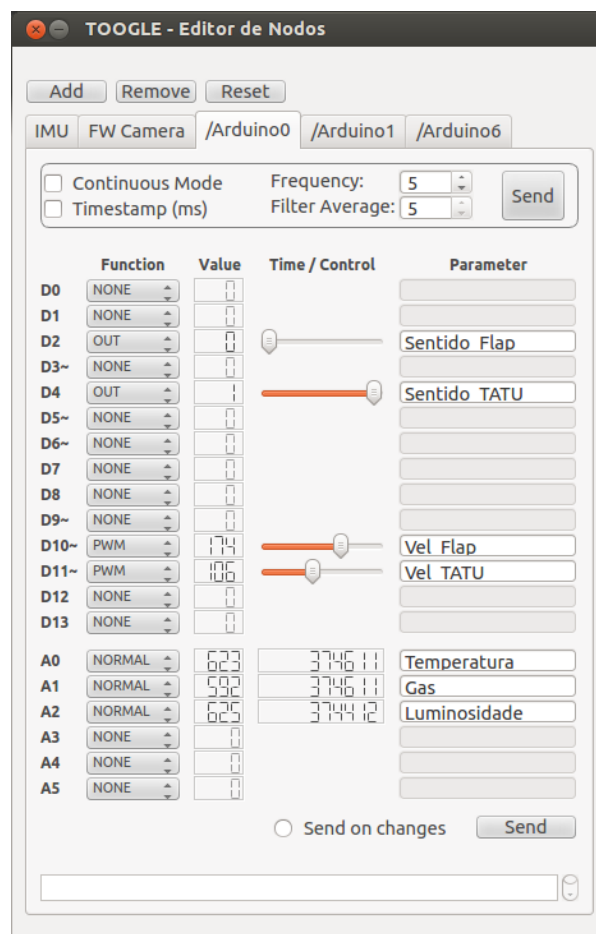


Figura 4.2: Tela do Toogle-Dispositivo

Uma biblioteca de dispositivos é disponibilizada. Nesta versão, três tipos de *dispositivos* que cobrem várias grandezas presentes em um ambiente físico estão acessíveis para uso, envolvendo placas de aquisição e atuação associadas ao kit de desenvolvimento arduino, câmera de vídeo e central inercial. Estas características podem ser percebidas ou modificadas no meio através de sensores e atuadores robóticos.

Dentre as características do ambiente que podem ser percebidas através dos sensores atuais disponibilizados encontram-se: temperatura, luminosidade, proximidade/presença, som, campo magnético, umidade, gás/fumaça, fogo/chama, inclinação, balanço/oscilação, luminosidade, pressão, aceleração, posição e imagem. Sendo que as duas últimas, são captadas no Toogle-dispositivo através da câmera firewire e de IMU, sendo todos disponibilizados como *recursos* providos do *dispositivo* Plataforma Hardware arduino associado a um *smartobject*.

Dentre as ações que modificam as propriedades físicas e que são executadas pelos atuadores na versão Toogle-dispositivo atual, encontram-se: cor/Intensidade de cor, som/intensidade de som, controles do tipo liga/desliga, laser, controle de ângulo/inclinação e velocidade de giro/torque. Sendo que as duas últimas, são captadas pelo módulo através da câmera firewire e de IMU, neste caso utilizando a plataforma de hardware arduino. Estas características se dão principalmente através da utilização de atuadores como LEDs, buzzers, auto-falantes, relés, motores DC, servomotores, entre outros.

4.2 Toogle Middleware

As informações associadas a entidades físicas no hiperambiente poderão ser acessadas e modificadas através dos recursos e serviços. O toogle middleware integra os diferentes *smartobjects* permitindo a efetiva construção e navegação de hiperambientes.

Como descrito no modelo conceitual os serviços de aquisição e modificação de informações de entidades físicas podem ser compostos por vários tipos de dispositivos: tags RFID, sensores e atuadores heterogêneos. O Toogle message/passar é o middleware responsável pela comunicação necessária para que estes serviços e recursos possam ser estabelecidos, permitindo a abstração de sua realização por parte do usuário.

O módulo é composto pelo sistema ROS, onde cada *smartobject* presente em um CPS

é representado por um nó, um termo utilizado pelo ROS para definir uma aplicação independente de outras aplicações rodando em tempo de execução do sistema ROS. O ROS consiste em um middleware para robótica que fornece bibliotecas e ferramentas para criação de sistemas robóticos distribuídos, fornecendo abstração de hardware, drivers de dispositivos, bibliotecas, visualizadores, sistema de troca de mensagens, gerenciamento de pacotes e muito mais.

Quanto ao funcionamento, ROS consiste de procedimentos sendo executados na mesma ou em diferentes máquinas, encapsulados por identificadores únicos denominados nós. Os procedimentos usam dois tipos de comunicação de envio de mensagens entre si: a primeira é síncrona e é chamada de serviço; a segunda é assíncrona, onde tópicos são publicados por um nó. Os processos, podem se inscrever para ter acesso às mensagens de outros nós a cada vez que eles publicam um tópico. Vários processos podem estar paralelamente inscritos e publicando no mesmo tópico e um único nó pode publicar e se inscrever em vários tópicos.

No Toogle, os nós-ROS podem publicar tópicos (mensagens) advindas do serviço de *recursos* com informações sobre o *smart object* que estes representam. Estas mensagens contêm informações sobre sensores e atuadores que serão utilizadas pelo Toogle Navegador/Editor na descrição de um hiperambiente. Por exemplo, um tópico pode conter um valor de temperatura lido a partir de um sensor ou uma grandeza de torque a ser aplicado em um motor. Estes tópicos são acessados pelo Toogle Navegador/Editor, que utiliza as informações do tópico para simular virtualmente o objeto real, como por exemplo, a simulação virtual da trajetória do veículo, baseado no torque aplicado ao motor do exemplo anterior.

4.3 Toogle Editor

O Toogle Editor possibilita construir e editar hiperambientes, permitindo o desenvolvimento de sistemas CPS, conforme modelo conceitual apresentado. O Toogle editor permite a criação, a edição, a ligação e a remoção das informações associados a diferentes entidades virtuais e *smart objects* presentes em um determinado hiperambiente.

Pode-se fazer uma analogia a um editor de páginas para a internet: um hiper-

texto constitui-se de um agregado de informações e suas conexões, os hiperambientes constituem-se de um agregado de informações reais/virtuais sobre determinado cenário e suas conexões. Portanto, o editor é capaz de criar e representar as diferentes informações das entidades e fazer as conexões/ligações entre estas. Para cada hiperambiente, é possível criar e adicionar novas entidades, verificar seu estado, adicionar entidades pré-existentes, associar a estas recursos e serviços.

O Toogle Editor utiliza recursos da ferramenta Blender de desenvolvimento para criar e editar e visualizar os cenários tridimensionais. O Blender é uma ferramenta de desenvolvimento gráfico que oferece recursos gráficos tridimensionais para visualização dos objetos e representação das informações. O Blender vem sendo adotado como ferramenta de visualização no desenvolvimento de arquiteturas para ambientes ubíquos, Marder [2010] devido as suas características de ser open-source, multi-plataforma, fornecer suporte a estereoscopia e multiprojeção, além de englobar a ferramenta de simulação física Bullet Physics Library [2013].

A figura 4.3 mostra o Toogle Editor. Os objetos na figura são elementos envolvidos no hiperambiente e, para cada hiperambiente, é possível criar e adicionar novos objetos e defini-los como *smartobjects* (como destacado na figura), assim podendo associar *Entidades Físicas* com *Dispositivos*, *Entidades Digitais* com *Serviços* através da associação de *Proxy's Digitais*, capacitando a *smartobjects* de visualizar e trocar informações no sistema.

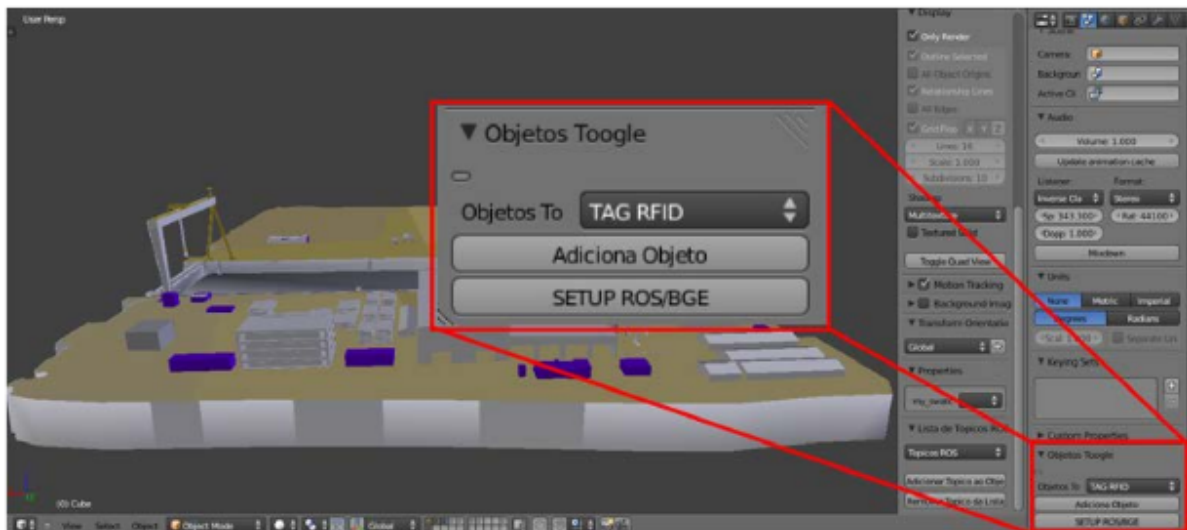


Figura 4.3: Sistema Toogle Navegador/Editor, adicionando novo nodo RFID

Na figura 4.4 mostra uma visão geral do editor de hiperambientes da Toogle, com vários módulos da Interface Gráfica do Usuário: Editor de código Python, responsável por editar códigos para o Sistema Toogle; o editor de estados e lógica do objeto, este possui ferramentas para aplicar lógica aos objetos, possibilitando controle e inteligência aplicando comportamentos aos objetos; editor e visualizador gráfico, nele são feitas as modelagens 3D no cenário e visualização dos comportamentos em tempo real dos objetos neste cenário; e o painel do editor Toogle, onde ficam todos os controles de interface responsáveis pela edição dos elementos do hiperambiente.

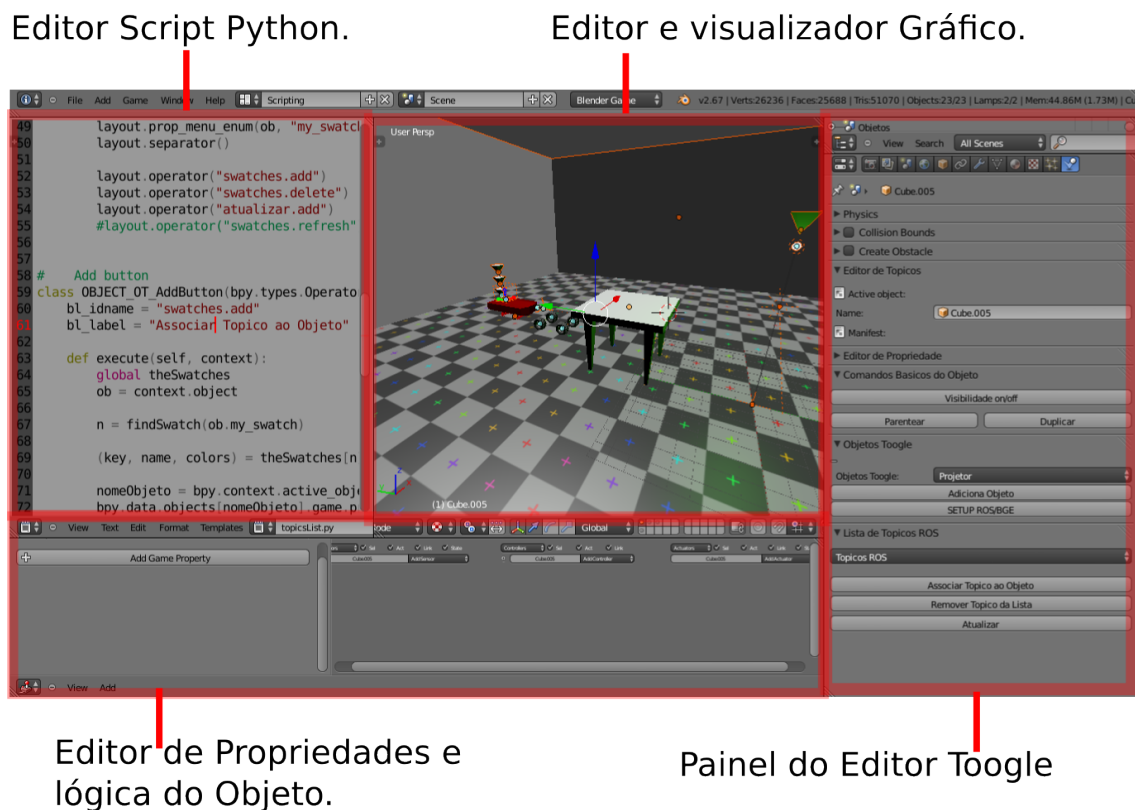


Figura 4.4: Interface do Editor de Hiperambientes da Toogle/Blender

Neste painel, na figura 4.5 encontra-se o serviço de biblioteca de representação gráfica, responsável por associar uma representação gráfica a um objeto da cena ou *smartobject*,

Para associar um objeto de representação gráfica do cenário virtual a um proxy digital, basta abrir a listas de proxy's que estão ativos no Sistema Toogle fornecidos pelo Serviço de Resolução e selecionar um dos destes da lista, estes proxy's são os nodos e tópicos do middleware ROS. Novos proxy's podem ser criados no editor Toogle, como um novo

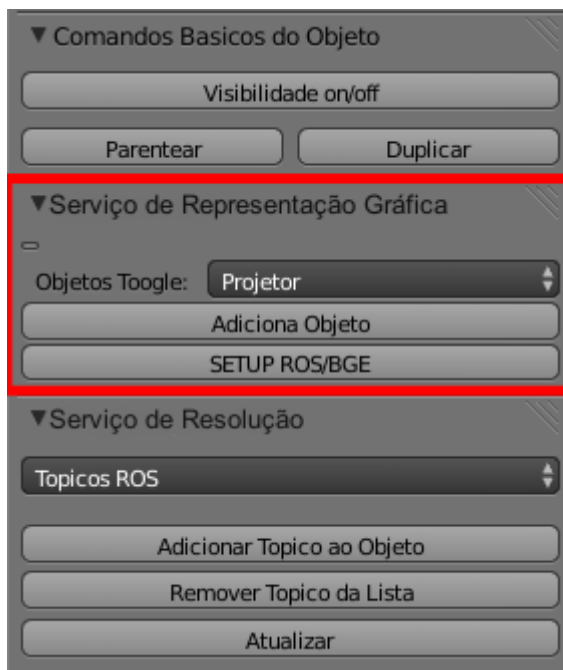


Figura 4.5: Painel Toogle/Blender - Objetos da Toogle

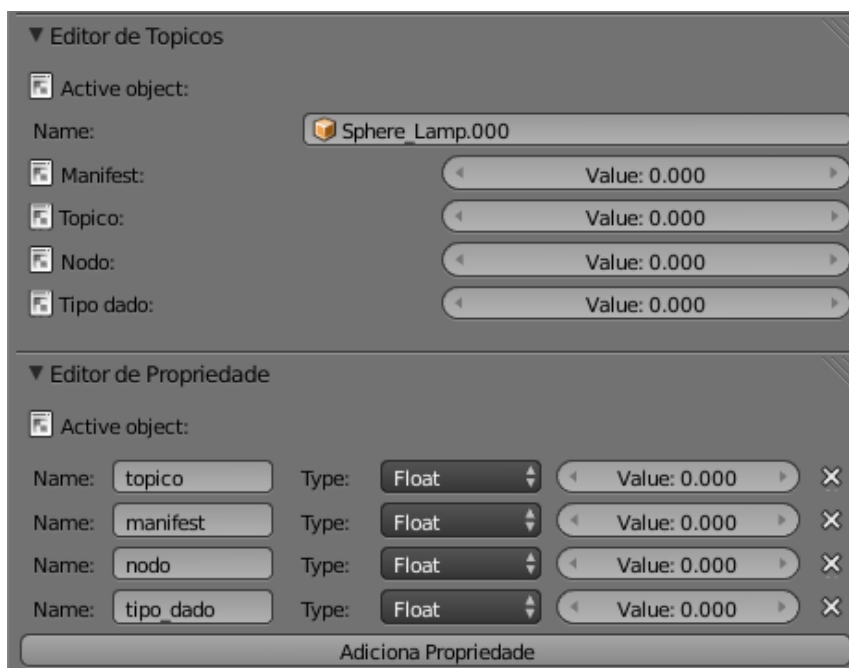


Figura 4.6: Editor de propriedades do Objeto

serviço por exemplo. A interface do editor Toogle, permite a configuração manual de seus estados, adicionando outros atributos não previstos no projeto, cada estado possui três

campos que podem ser usados na sua definição . Figura 4.6.

4.4 Toogle Navegador

O módulo Toogle Navegador permite a navegação por hiperambientes de forma a acessar as informações circulantes em um CPS. Novamente em analogia a hipertextos, este módulo seria o navegador responsável pelo acesso as informações presentes neste arquivo. Portanto, o Toogle Navegador é capaz de monitorar e modificar os diferentes estados associados as entidades virtuais e *smart objects* do sistema.

Interface de Visualização Avançada A verificação e validação do CPS atualmente é propiciada por uma interface gráfica 3D com recursos de estereoscopia e multi-projeção para ambiente do tipo CAVE Filho [2009]. Esta interface possibilita a visualização dos *smartobjects* e informações providas por *Proxs Digitais* do Toogle e representa graficamente de forma tridimensional e imersiva. Esta conta com recursos multi-projeção e estereoscópicos. Assim, o Toogle navegador possibilita o acesso e visualização das informações relacionadas ao hiperambiente que foram criadas no Toogle Editor. Ele é um visualizador de cenários tridimensional, onde é possível ver a representação gráfica do hiperambiente, envolvendo suas entidades e estados. A interface do navegador é a mesma do editor, já descrita anteriormente. Esta interface possibilita a visualização dos *smartobjects* e informações providas por *Proxy's Digitais* do Toogle e representadas graficamente de forma tridimensional e imersiva.

Simulador Físico O Toogle navegador conta também com um simulador físico que permite avaliar a evolução temporal de determinadas grandezas associadas a *smart objects*. Este é responsável pelas simulações físicas, permitindo realismo e alto desempenho, que contribui para o CPS quanto ao aspecto de *validação* das informações na forma de simulação física, e como *modelo estocástico* de análise. A simulação é executada em *offline*, isto é, sem sincronia com o hardware. Tais características podem atender aos requisitos de projeto de um CPS, abordados no capítulo 2. O sistema utiliza a biblioteca de código aberto Bullet, com renderização em tempo real através do OpenGL [Albanowski, 2013]. Mecanismos de detecção de colisão e dinâmica de corpos rígidos estão presentes na ferra-

menta. O simulador físico permite simulações de objetos que possam cair, rolar e colidir com outros objetos de uma maneira bem realística. Aspectos de iluminação de cenas usam técnicas de Pixel Shaders, utilizados em plataformas gráficas de alto desempenho com resultados mais realísticos. Vale resaltar a importância da simulação física, como sendo um elemento chave na representação e integração do mundo físico com o cibernético.

Plugin para Internet O advento do Html5 [e Mark, 2010] possibilitou o desenvolvimento de ferramentas de visualização 2D/3D para navegadores web sem a necessidade de utilização de plugins para visualização 2D/3D. Desta forma seria possível a visualização de hiperambientes em tempo real acessados de forma remota pelo usuário via navegador na internet.

Neste trabalho uma proposta de inicial de arquitetura é apresentada na figura 4.7, onde percebe-se a sua integração com a plataforma Toogle. Nesta arquitetura há dois elementos principais, o *Módulo Servidor* e o *Módulo Cliente*. No Módulo Servidor tem-se o middleware ROS que é integrado ao *Servidor Web Socket* através da biblioteca Java script *ROS Brigade*. O middleware responsável pela comunicação entre os módulos servidor e cliente é o *Web Socket*, sendo este um recurso disponibilizado pelo novo Html5. No *Módulo Cliente*, o navegador web com suas bibliotecas Java Script, web socket é o responsável pela comunicação, como explicado anteriormente, a seguir tem-se o protocolo EEML que fornece a estrutura coerente para representar as informações recebidas repassando-as para as bibliotecas responsáveis pela renderização do cenário e suas interações.

O navegador web proposto aqui neste trabalho para hiperambientes é composto por cinco módulos principais, que trabalham de forma independente do módulo Toogle Editor / Navegador, visualizados na figura 4.8, são eles:

- **WebSocket** é o responsável pela comunicação de dados entre o navegador web e o servidor da plataforma Toogle. Constitui-se em uma tecnologia para comunicação bidirecional por canais full-duplex sobre um único soquete *Transmission Control Protocol* (TCP). É projetado para ser executado em navegadores e servidores web que suportem o HTML5¹, mas pode ser usado por qualquer cliente ou servidor de aplicativos. A API *WebSocket2* está sendo padronizada pelo W3C³ e o protocolo WebSocket está sendo padronizado pelo IETF⁴ [Bijin Chen e Zhiqi Xu, 2011];

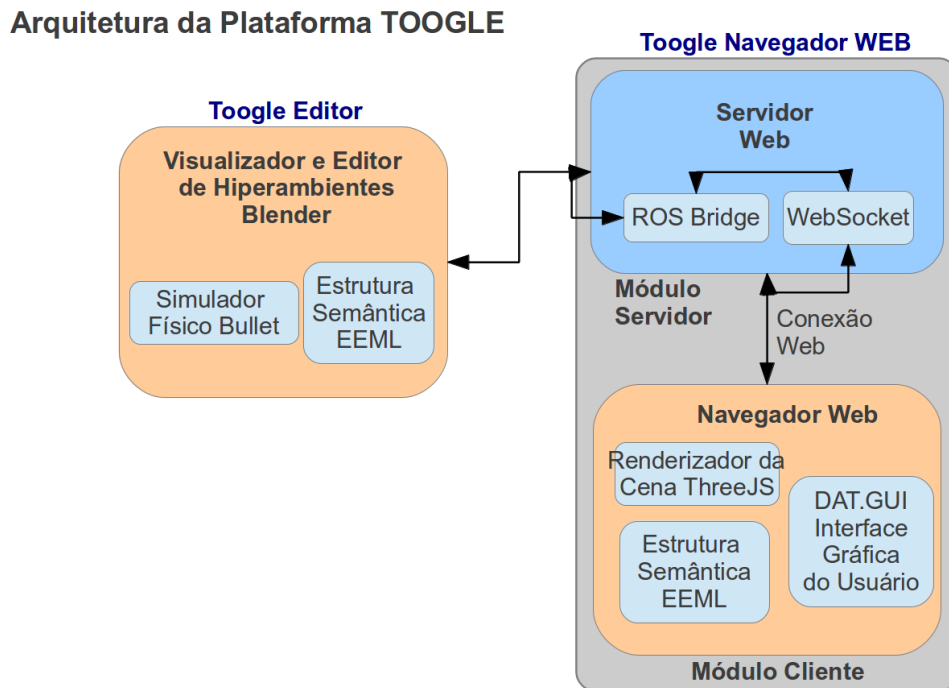


Figura 4.7: Arquitetura proposta do Navegador web para Hiperambientes

- **Scene Render ThreeJS** é um framework *JavaScript* biblioteca / API utilizada para criar e exibir animação gráfica 3D em um browser. *ThreeJS* scripts podem ser usados em conjunto com o elemento HTML5 canvas element, SVG ou WebGL. O *ThreeJS* possui um editor 3D on-line open source, com recursos básicos de modelagem 3D que é disponibilizado junto com o pacote *ThreeJSs* [Dirksen.js, 2013];
- **GUI Dat.gui** é responsável por prover interação do usuário com o navegador web para hiperambientes, sendo uma biblioteca de controle leve, inspirada no design de interface do *Sistema Operacional Android*, que permite a fácil manipulação de variáveis e funções através de uma Interface Gráfica para Usuário (GUI).[dat, 2013]
- **Physics Engine Physijs** é responsável pela simulação física dos elementos do cenário, traz uma interface simples para o framework *ThreeJS*. Physijs utiliza o mesmo conceito de usabilidade do *ThreeJS*, o que torna sua implementação totalmente compatível junto a este, entretanto o Physijs e a Bullet executam suas simulações de

forma independente, cada um em seu módulo da Toogle. [plugin for Three.js, 2013]

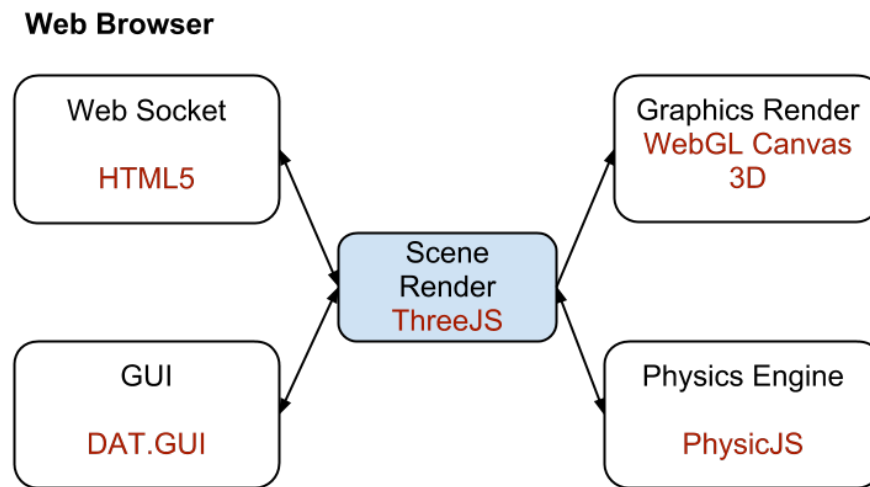


Figura 4.8: Arquitetura dos componentes para o Navegador web de Hiperambientes

Com a conexão estabelecida entre o navegador web e o servidor, o módulo EEML passa a receber os dados e estrutura-los de forma coerente, dando acesso aos estados e serviços dos *smartobjects* e *Entidade Digitais*. O módulo renderizador de cena *ThreeJS*, acessa esses dados e interpreta as saídas de acordo com suas características, análogo ao um navegador de web padrão, que interpreta um código html e exibe suas informações de acordo com o seu tipo, como por exemplo um campo contendo uma informação do tipo imagem será exibida no navegador como tal. O módulo *Dat.gui*, tem a função de visualizar dados recebidos dos estados das entidades e inferir novos dados através de sua interface. Por último tem-se o módulo de simulação física, que provê simulações físicas e pode fornecer a garantia de certificação de comportamentos físicos de entidades físicas no mundo real. Estas funcionalidades permitem o acesso às informações de diferentes naturezas, seu monitoramento e demais tratamentos possíveis a partir de um hiperambiente descrito em EEML.

4.5 Considerações Finais

Neste capítulo foi apresentada a plataforma de desenvolvimento para CPS, onde foram descritos os seus elementos e suas relações com o modelo conceitual de CPS. Destaca-se ainda a utilização do conceito de hiperambientes como elemento central do sistema.

No próximo capítulo será abordada a metodologia empregada para a implementação de hiperambientes com a plataforma Toogle.

Capítulo 5

IMPLEMENTANDO HIPERAMBIENTES COM A TOOGLE

Este capítulo visa apresentar um exemplo de uso da ferramenta Toogle para criar um hiperambiente. O capítulo explora as definições apresentadas no modelo conceitual e descreve os passos necessários para implementar, criar, acessar e editar um Sistema Físico Cibernético. A Toogle e seus módulos são empregados e um passa-a-passo visando esclarecer ao leitor pontos importantes do trabalho.

O Sistema Físico Cibernético apresentado como exemplo no Capítulo 3 é recuperado nesta seção, de forma a permitir que se apresente todos os passos de desenvolvimento de um CPS utilizando o hiperambiente e a ferramenta Toogle. O ambiente utilizado é mesmo apresentado na figura 3.3.

5.1 Passo 1: Adicionando dispositivos - sensores e atuadores

No modelo conceitual proposto para CPS é descrito o elemento *Dispositivo*, responsável por integrar dispositivos sensores e atuadores à arquitetura. O módulo Toogle Dispositivo faz uso do conceito *Dispositivo* do modelo conceitual para adicionar e integrar dispositivos físicos mantendo estes conectados ao sistema Toogle, disponibilizando seus recursos para o middleware ROS, que por sua vez cria os proxys digitais para serem utilizados pelo sistema como serviços de hardware. Uma descrição breve do procedimento de adição de dispositivo pela interface gráfica do usuário, já foi descrita na seção 4.1.

5.2 Passo 2: Adicionando serviços e recursos

Descrito como uma das cinco características de um *smartobject*: "uso de hardware complexo ou serviços de software fornecidas pelo *smartobject* associado", é tratado aqui sobre o serviços de software, tendo em vista que os serviços de hardware foram descritos no passo 1. O exemplo, utiliza o objeto Camera do ambiente exemplo, que pode assumir duas formas de classificação, pode ser tanto um *smartobject* como pode ser um Serviço ou Recurso de um outro *smartobject*, isto dependerá de como se quer ou necessita-se que o objeto Câmera se comporte com relação ao ambiente. Neste caso específico é classificado como *smartobject*.

O *smartobject* Câmera, captura imagens do ambiente físico, mas não há qualquer processo computacional inteligente analisando as imagens, supondo que o objetivo deste *smartobject* tenha a função de detectar presença ou movimento, tem-se a necessidade de um agente inteligente, que pode ser um software encarregado de analisar as imagens e devolver a informação para o *smartobject*, se houve ou não, detecção de movimento

ou presença. Para que este software forneça tal serviço, ele precisa estar conectado ao sistema como uma *Entidade Digital*, que tem seu *Proxy Digital* e categorizado no *Serviço de Resolução*. Este contém uma lista de todos os recursos e serviços disponíveis no sistema, com o serviço estando selecionado pelo usuário, ou pelo agente de software, este é associado ao *smartobject* requerente deste serviço. Como demonstrado na figura 5.1, tal software ou serviço, deve ser desenvolvido ou adaptado dentro dos requisitos do Sistema Toogle.

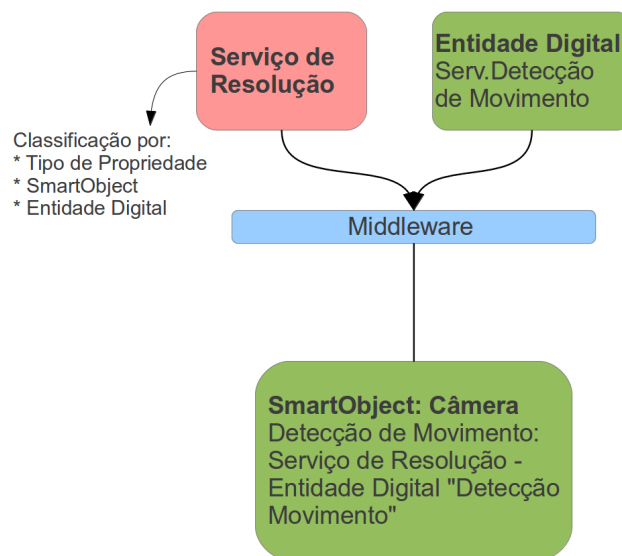


Figura 5.1: Arquitetura de serviços e recursos

Nos elementos apresentados na figura 5.1, destaca-se o Serviço de Resolução "*Serv.Resolution*", que é um serviço responsável por categorizar os Recursos disponíveis no sistema. Um dos recursos categorizados é o de Deteção de Movimento, que não é um recurso local do *smartobject* Câmera, este pode ser associado a este, como para vários *smartobject* simultaneamente. O Serviço de Resolução não é obrigatório no sistema, mas ele ajuda no gerenciamento dos recursos disponíveis nos sistema.

5.3 Passo 3: Modelando um ambiente físico

O Toogle Editor permite a criação de ambientes virtuais interativos, que podem ser usados como representações e simulações de ambientes físicos, conforme figura 5.2. As simulações

podem atualizar os estados dos *smartobject*, realizando comportamentos de controle, percepção e atuação inteligentes, representação física geométrica e simulação física acurada, disponibilizado pelo motor de física *Bullet*. Tais características podem vir a atender a alguns requisitos do Modelo Conceitual de CPS, como: garantia e certificação, através dos recursos de simulação e comportamento do Módulo de Simulação Física da Toogle. Todos os objetos criados, são armazenados no próprio arquivo de edição, formando uma biblioteca de representação gráfica para usos futuros.

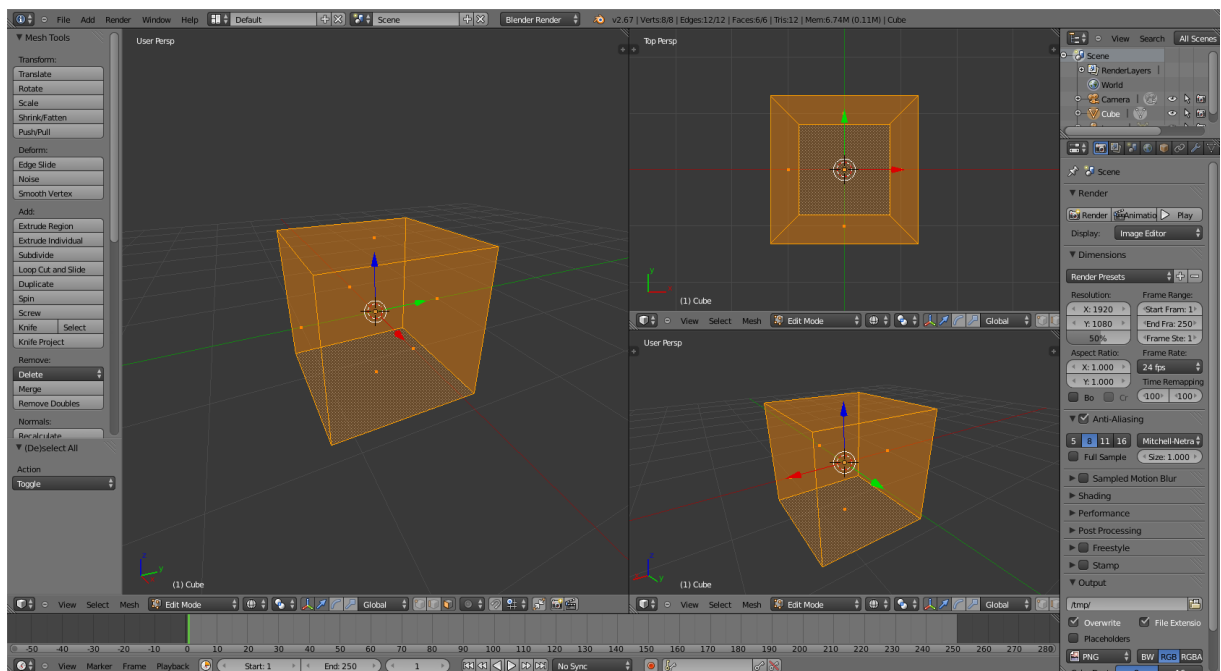


Figura 5.2: Exemplo de objeto modelado em 3D

5.4 Passo 4: Procedimentos básicos do Sistema Toogle

5.4.1 O Editor Toogle

Após ser criado um cenário virtual com a ferramenta de modelagem 3D da Toogle, tem-se início a configuração dos elementos virtuais, que irão representar as diversas entidades da arquitetura, como *smartobjects*, *Entidades Físicas e Digitais*, *Recursos* e *Usuários*. Cada objeto de representação gráfica pode ser associado ao um proxy digital, ou ser um elemento não discriminado no ambiente físico, mas mesmo assim passível de influenciar

os elementos com proxy digital associado.

Com o cenário proposto modelado, o usuário pode rodar a aplicação de hiperambientes dentro do próprio editor, permitindo testes rápidos, novos ajustes e configurações no cenário. Na figura 5.3. é demonstrado um exemplo de aplicação em execução, composto por um veículo que é *smartobject* com display de seus estados de orientação, um piso e uma mesa que são *Entidades Digitais* e no canto superior esquerdo um display com o monitor de recursos computacionais do sistemas.

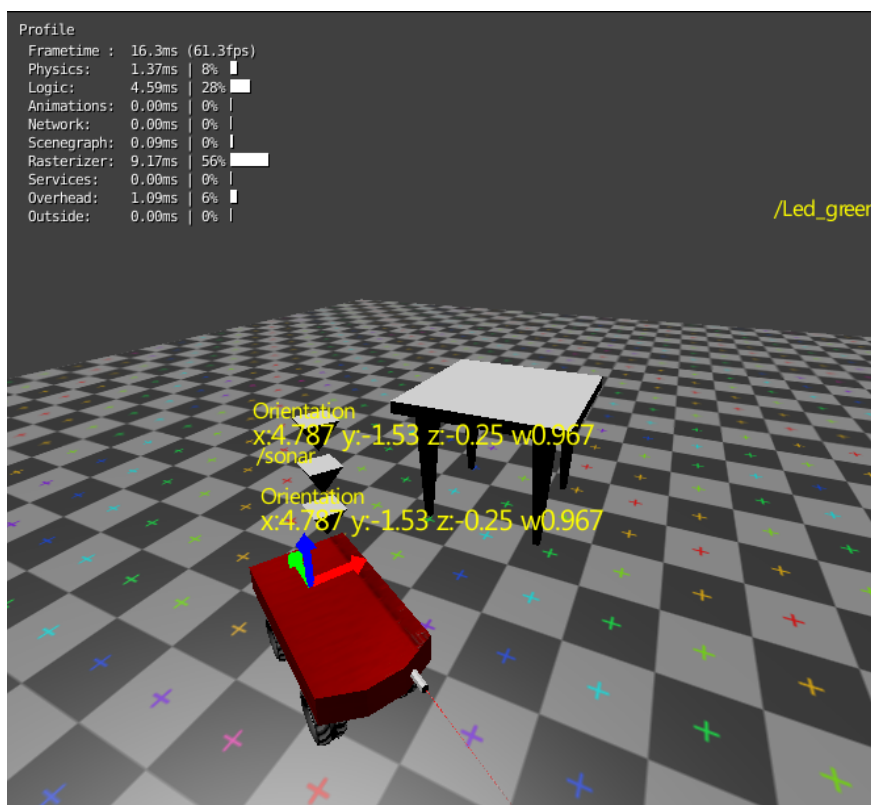


Figura 5.3: Exemplo do Hiperambiente em execução pelo Editor/Navegador Toogle

5.4.2 O Editor Toogle Persistência de Dados

O hiperambiente pode ser armazenado em dois formatos distintos, o primeiro guarda o arquivo no formato blend, que é o formato nativo do Blender 3D, este guarda todas as configurações do cenário, incluindo, scripts, imagens, objetos 3D, animações entre outros, possibilitando no futuro novas edições do cenário. O segundo formato é o formato

XML no esquema EEML, este guarda o Hiperambiente propriamente dito, todo o significado semântico dos elementos com proxy do cenário. O EEML gerado pelo Editor Toogle/Blender3D, pode ser lido por qualquer outro Navegador de Hiperambientes, como o Navegador Web para Hiperambientes proposto nesta dissertação, que terá suas próprias interpretações do EEML, independente do Editor Toogle, isto significa, para que o Hiperambiente tenha coerência em seu funcionamento, o desenvolvedor deverá fazer uma implementação que corresponda aos requisitos mínimos propostos pela arquitetura do sistema para CPS. Com a leitura do EEML, tornar-se possível a geração de um cenário que irá representar através do Hiperambiente: *smartobjects*, *Entidades Físicas e Digitais*, com representações gráficas 2D/3D. Análogo a uma página de internet, para que a entidade possa ser representado de forma coerente, cada tipo de objeto tem um plugin associado a este. Na figura 5.4, mostra um exemplo de fluxo de dados, de como é realizado o processo de persistência de dados das informações do hiperambiente e do cenário, começando pelo editor Toogle Editor que acessa os arquivos do tipo blend e EEML, sendo este último acessado posteriormente pelo navegador de Hiperambientes.

5.5 Considerações Finais

Neste capítulo foi apresentado um exemplo de desenvolvimento de um CPS utilizando a ferramenta Toogle. Um hiperambiente contendo diferentes tipos de entidades e recurso foi passo-à-passo descrito. No próximo capítulo serão realizados alguns testes os quais permitirão analisar e validar a proposta.

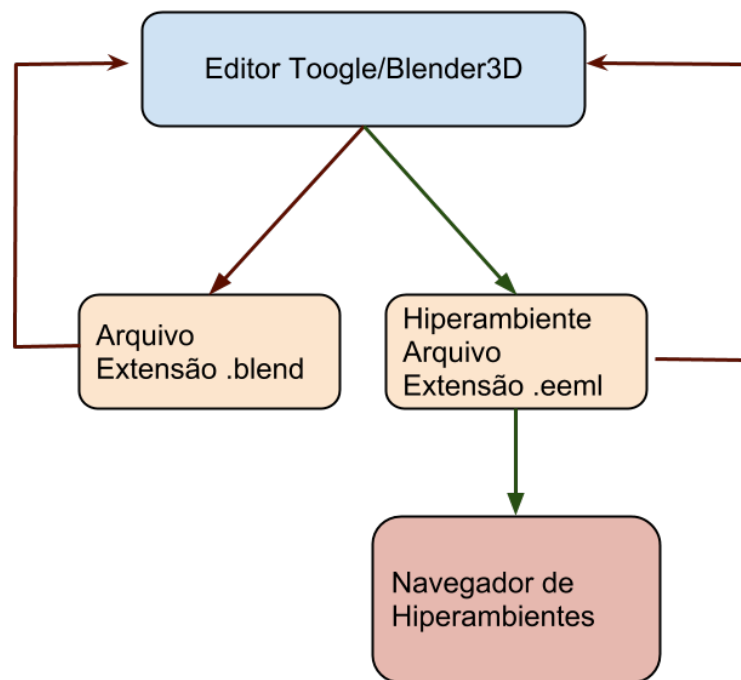


Figura 5.4: Exemplo do Fluxo de dados gerados para reutilização posterior

Capítulo 6

TESTES E RESULTADOS

Neste capítulo serão abordados alguns testes realizados com a plataforma Toogle, empregando o Modelo Conceitual de CPS descrito nesta dissertação. Para tal, foram utilizados dois estudos de caso, o primeiro teste trata de um CPS para um Sistema de Rastreamento de Insumos, que utiliza como estudo de caso o Estaleiro Rio Grande, situado na cidade de Rio Grande. Após as seções de testes será apresentada a análise dos resultados obtidos. O segundo teste consiste em um protótipo de plataforma robótica desenvolvida para a inspeção de dutos subterrâneos, onde foi implementado um CPS para teleoperação, controle e análise do protótipo.

6.1 Caso de Estudo: Estaleiro

Um dos grandes problemas observados nos estaleiros brasileiros é a falta de tecnologias de auxílio à localização, no pátio de armazenamento, de chapas metálicas e de pequenas peças cortadas e soldadas. A partir do sensoriamento de ambientes através de RFID [Want, 2006] é possível, por exemplo, realizar o controle de inventários em plantas produtivas. As tecnologias RFID são utilizadas para uma larga variedade de aplicações, desde rastreamento de grandes equipamentos, como veículos, até autenticação de produtos, bens ou mercadorias. Consistem em tecnologias que envolvem comunicação entre leitores e etiquetas, através de rádio frequência. O propósito de uma etiqueta *TAG* RFID é anexar fisicamente dados sobre um objeto. Todas as TAGs possuem duas capacidades básicas, a primeira é a de poder se anexada a um item de alguma forma e a segunda é a de transmitir as informações nela contida por alguma frequência de rádio. O Leitor é um dispositivo que captura e processa os dados das etiquetas. Alguns leitores possuem ainda a capacidade de gravar dados em uma etiqueta. Este ainda é responsável pela interface com o computador. O caso de estudo constitui-se na construção de um protótipo para a simulação da interface dos equipamentos de rastreamento no estaleiro. O protótipo foi desenvolvido em hiperambiente conforme etapas descritas a seguir:

6.1.1 Etapa 1: Modelo 3D do Estaleiro Rio Grande

Na primeira etapa do experimento, utilizou-se o sistema Toogle Editor para criar o modelo 3D do Estaleiro Rio Grande Figura 6.1.

6.1.2 Etapa 2: Maquete do Estaleiro Rio Grande

Como segunda etapa do desenvolvimento do protótipo, foi criada uma maquete do Estaleiro Rio Grande, construída a partir do modelo 3D da etapa 1. Esta maquete representa o estaleiro em uma escala de 1:750, conforme ilustrado na figura 6.2.

Nesta maquete foram colocadas três TAGs dos dispositivos RFIDs, a fim de representar o rastreamento de possíveis objetos e insumos dentro do estaleiro. Cabe chamar a atenção aqui que, devido as restrições de equipamentos, foi necessário uma inversão na forma de utilização do sistema RFID. O funcionamento comum desses sistemas implica na utilização

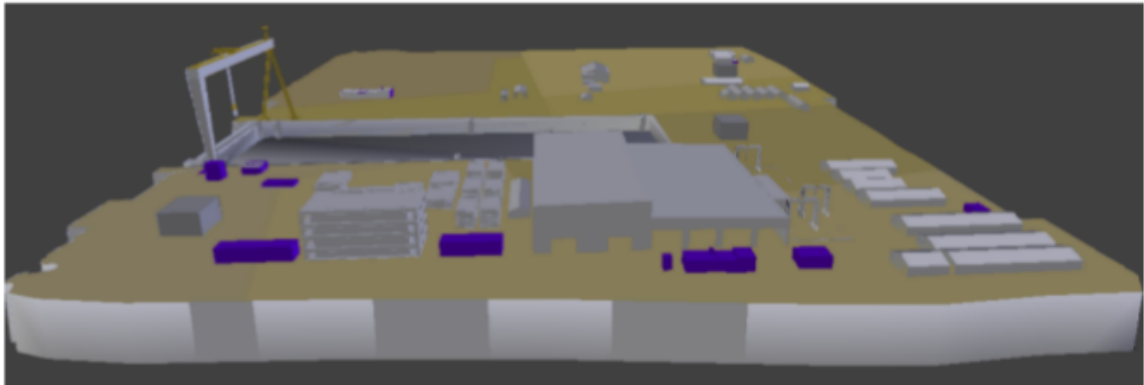


Figura 6.1: Modelo 3D do Estaleiro Rio Grande



Figura 6.2: Maquete 3D do Estaleiro Rio Grande

de leitores RFIDs estáticos distribuídos pelo cenário, enquanto as TAGs, fixadas aos objetos, se movimentam e podem ser rastreadas. Para tal experimento, a equipe tinha a disposição apenas um leitor e um conjunto de TAGs pelo cenário (maquete), tornando-as fixas e movimentou-se o leitor, simulando assim o rastreamento dos possíveis objetos e insumos do estaleiro.

Os equipamentos utilizados no experimento podem ser vistos na Figura 6.3.



Figura 6.3: Leitor e Etiquetas RFIDs

1. Leitor: L-RX201;
2. Tags: TAG DOMINO L-TG100;

6.1.3 Etapa 3: Adicionando os dispositivos ao cenário

A próxima etapa constitui-se de prover a comunicação dos dados do leitor RFID para a plataforma Toogle. Para isso foi utilizado o módulo Toogle dispositivo o qual cria a interface responsável e possibilita a configuração dos parâmetros do dispositivo, enviando para o leitor o comando para iniciar a leitura das TAGs para comunicação. No painel do editor gráfico da Toogle é adicionado um objeto 3D que representa o leitor RFID, resultando em um *smartobject*, já que este conta com uma *Entidade Física* e um *Dispositivo-leitor* RFID associado por um proxy digital e fornece um serviço de geolocalização das TAGs lidas. A TAG detectada também é um *smartobject* e associado a este através de um serviço de resolução, um proxy digital que é o serviço de identificação de TAGs com o nome "*tags-Lidas*" do *smartobject*-Leitor RFID, no qual possui uma função de posicionamento da TAG lida para sua posição em tempo real. A Figura 6.4 mostra o resultado dos aspectos descritos anteriormente na forma da estrutura semântica em EEML do Modelo Conceitual aplicada no teste.

A Figura 6.4 mostra a arquitetura desenvolvida no teste, destacando as conexões do

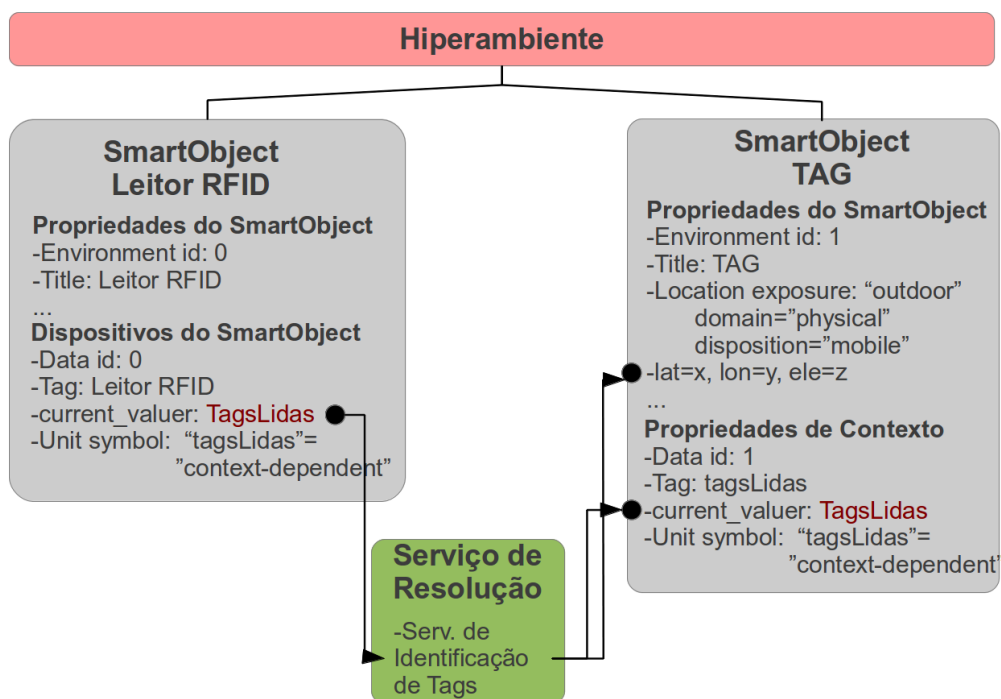


Figura 6.4: Estrutura Semântica do Modelo Conceitual aplicada no teste

leitor RFID e sua TAGs.

A TAG detectada pelo leitor envia os dados detectados podendo ser, por exemplo, intensidade do sinal, identificação (ID) da TAG, ID do leitor, no caso do experimento é publicado apenas o ID da TAG detectada. A figura 6.6 mostra no seu lado esquerdo o Serviço de Identificação de TAGs, listando os códigos das TAGs detectadas pelo *Dispositivo*-leitor RFID, no lado direito da mesma figura apresenta as propriedades do *smartobject*-Leitor RFID.

6.2 Resultados do primeiro teste

Para sua validação foi desenvolvido um Estaleiro Virtual a partir da planta real do Estaleiro Rio Grande ERG1, situado no Polo Naval RS. O Estaleiro Virtual ERG1 foi utilizado para rastreamento online de insumos/smartobjects e recursos através de TAGs de RFID. Através do leitor RFID foi possível a leitura e identificação da TAG em tempo real, sendo

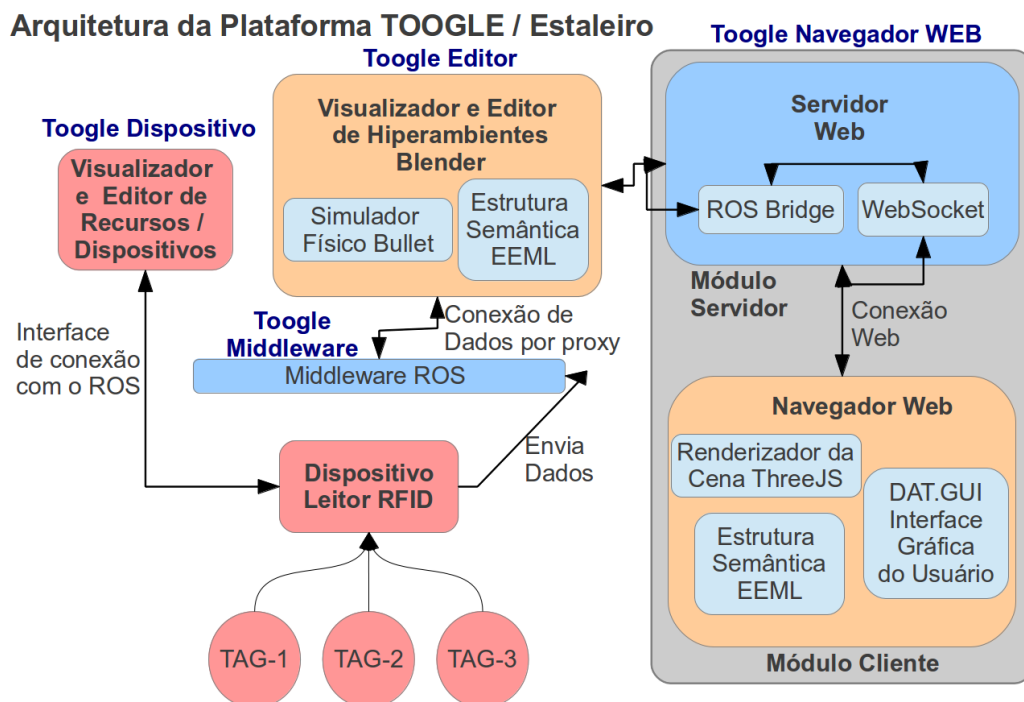


Figura 6.5: Arquitetura da Plataforma Toogle para o Hiperambiente Estaleiro

visualizado por um display no Toogle Navegador e por um modelo de representação gráfica de insumo/smartobject no ambiente 3D do Toogle Navegador. A plataforma desenvolvida permitiu a criação, edição e acesso à ambientes virtuais distribuídos que representam o conjunto de agentes fornecidos pela telepercepção. O ambiente desenvolvido foi implementado pelo modelo conceitual de Hiperambientes, este apresentou uma semântica e integração coerente entre os *smartobjects*, *Dispositivos* e *Entidade Físicas e Digitais* da arquitetura.

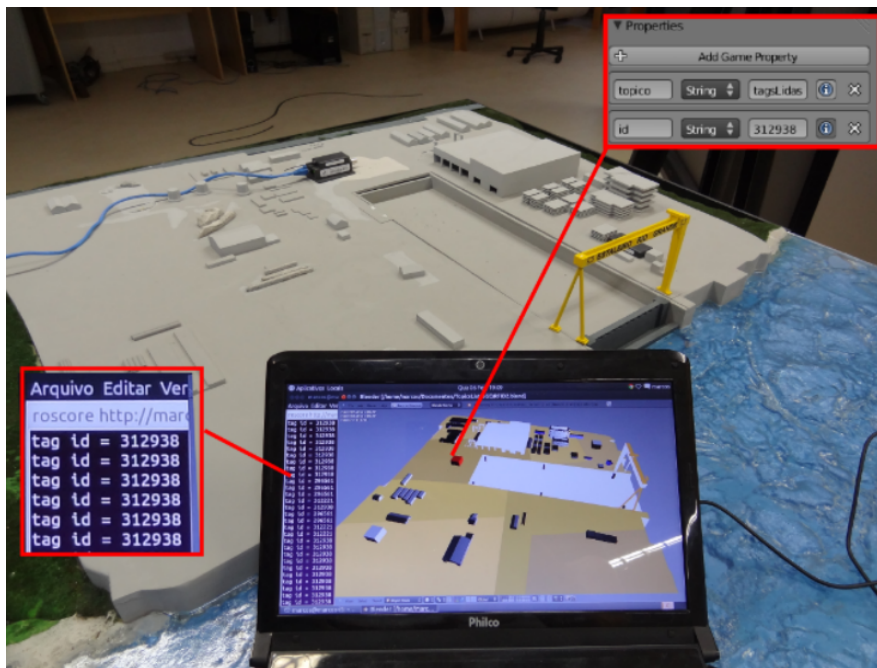


Figura 6.6: Toogle-Navegador com informações do tópico tagsLidas e propriedades do objeto

6.3 Plataforma Robotica Tatubot

Este teste apresenta a implementação de uma plataforma robótica, de forma a validar o uso do conceito de hiperambientes em um sistema CPS para uma missão de inspeção subterrânea e exploração. O experimento envolve usuários geograficamente distribuídos, contendo redes de sensores, atuadores e o robô Tatubot. Tal robô é equipado com câmera (Tritech Typhoon Colour Underwater Video Camera), sonar Tritech MiniKing e um conjunto de sensores (altímetro e acelerômetros). Através do uso da câmera o usuário pode visualizar o ambiente onde se encontra o robô e este pode se auto-localizar com a utilização de software de localização por análise de imagens. Os demais sensores são utilizados como complemento para a telemetria.

A inspeção Subterrânea: A inspeção da distribuição de energia apresenta uma série de desafios. Dentre as várias formas de distribuição, cabos de alimentação com diferentes conformações estão confinados no interior de dutos subterrâneos. O ambiente apresenta dificuldade de acesso, restringindo a realização de tarefas de inspeção e manutenção. A

detecção de falhas pontuais torna-se difícil uma vez que tais malhas envolvem quilômetros de cabos e dutos. Sistemas robóticos, veículos autônomos subterrâneos (autonomous underground vehicles) - AUVs começam a ser utilizados de forma a permitir a inspeção contínua do ambiente [Tongzhu Zhang, 2010]. Propõe-se um grid híbrido, formado por sensores estáticos e o robô de inspeção, este em conjunto com humanos na superfície, realizam de forma distribuída e colaborativa a inspeção subterrânea. A missão consiste em utilizar o Tatubot dentro da tubulação de cabos, para que o mesmo inspecione esse ambiente, passando por diversas situações que esse ambiente possa apresentar e seu movimento é feito com a utilização de rodas laterais apoiadas nas paredes. A figura 6.7, vista em um corte transversal do duto, demonstra as situações em que os cabos elétricos estão posicionados de forma irregular.

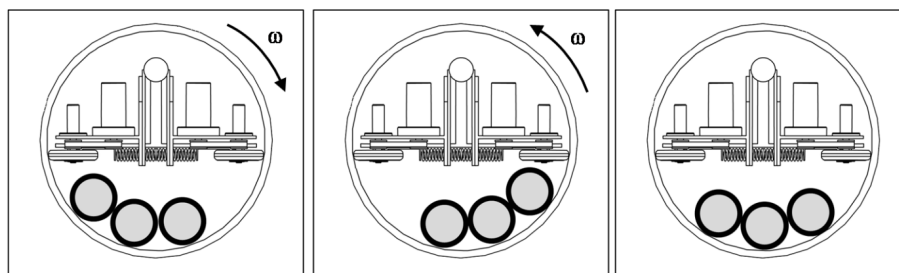


Figura 6.7: Ilustração da disposição dos cabos, duto e do Tatubot.

A plataforma Toogle foi utilizada para integrar a rede de sensores e os elementos do cenário. O hiperambiente foi composto por vários *smartobjects*, que por sua vez tem associados a si *Dispositivos*, ver Figura 6.8. Foi utilizada a plataforma open-hardware Arduino para os dispositivos sensores e atuadores que compõem o hiperambiente. Cada *Dispositivo* é conectado ao middleware ROS, para isto, foi utilizado o Toogle Dispositivo para criação e configuração dos sensores e atuadores, conforme a figura 6.9 e um hiperambiente *Tatubot* foi criado via editor.

Com o Toogle Editor foi possível criar este Hiperambiente o qual agrega as informações associadas aos *smartobjects*, e sua representação virtual, as quais são publicadas/recuperadas em suas propriedades e serviços. O navegador permite a sua visualização através de estruturas 3D, aplicadas ao Módulo Simulador Físico, descrito na seção 5.5. A estrutura semântica em EEML do Hiperambiente é demonstrada na figura 6.10.

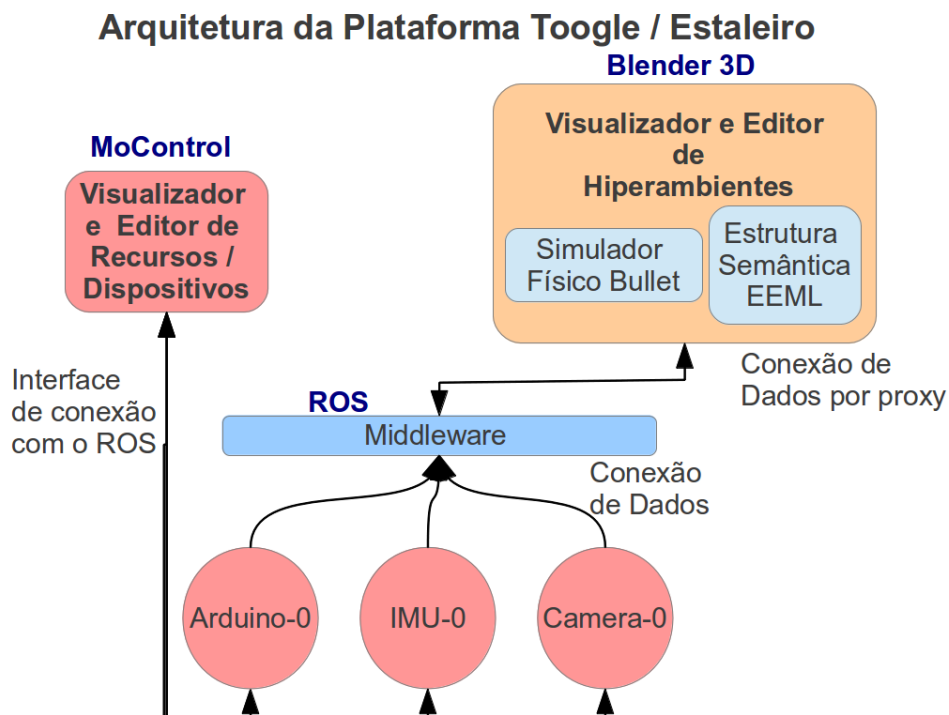


Figura 6.8: Arquitetura Toogle Tatubot

O modelo virtual do Tatubot mostrado na figura 6.11, foi criado a partir do modelo real mostrado na figura 6.12, com suas especificações: mecânicas, geométricas e de comportamento físico.

Resultado da Missão de Inspeção Usando o Hiperambiente: O navegador Toogle permitiu o controle e visualização virtual da missão real, explorando o conceito de recuperação/publicação de informações na rede. O visualizador e simulador embutidos no navegador permitiram representar as informações relativas aos sensores instalados no robô, através de displays de visualização na tela do navegador e o seu deslocamento pelo cenário com sua atualização em tempo real, bem como a simulação do comportamento físico imposto por obstáculos colocados em seu trajeto. Este atendeu a metodologia de projeto com suporte de validação e verificação, previstos pelo mapa conceitual de CPS

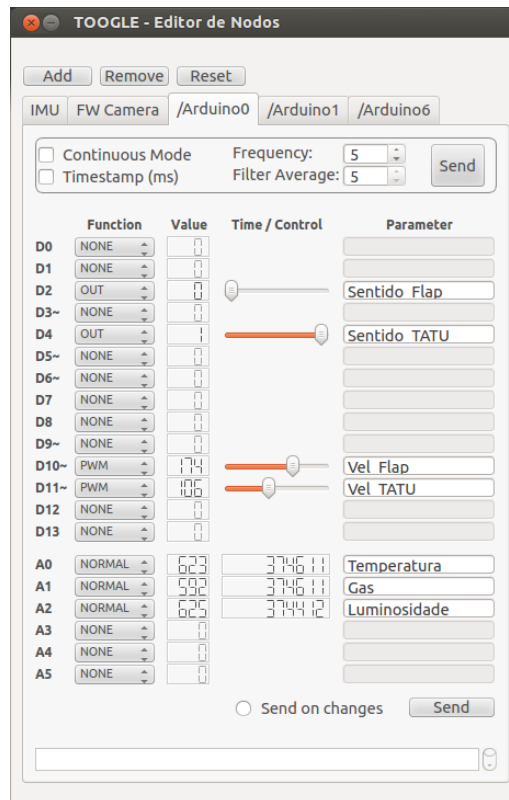


Figura 6.9: Editor/Visualizador de dispositivos físicos Toogle

da Figura 2.2, através de simulação física provido pelo módulo. Também foi possível a teleoperação remota do robô através do envio de comandos de *Usuário*, utilizando *Serviços e Recursos* dos *smartobjects*.

O navegador Toogle disponibilizou recurso de visualização imersivo em um dispositivo CAVE onde se demonstra a missão realizada, utilizando a arquitetura descrita anteriormente, ver figura 6.13.

Resultados e objetivos alcançados Como forma de se verificar que os resultados dos testes atenderam aos objetivos do projeto, se faz necessário recuperar os requisitos de projeto de CPS, descritos na seção 1 em Sistemas Físicos-Cibernéticos e comparar com os resultados, que a seguir são apresentados.

- **Edição e Navegação:** Através da arquitetura e interface gráfica do sistema Toogle, foi possível visualizar, acessar e modificar as informações das entidades, nos dois experimentos.

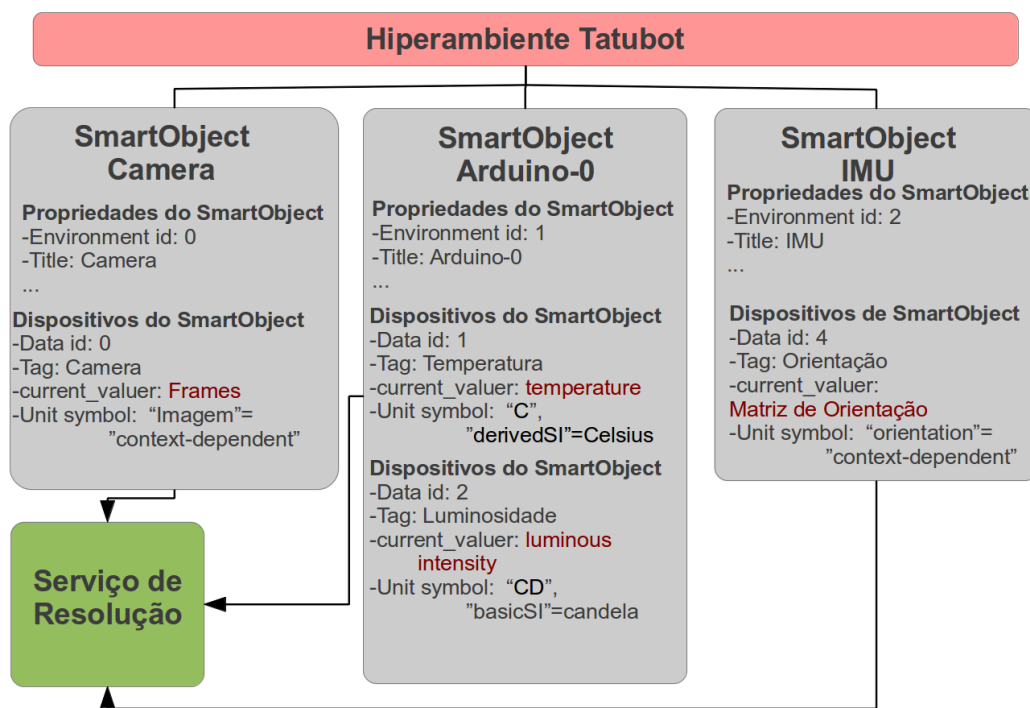


Figura 6.10: Semântica Toogle Tatubot

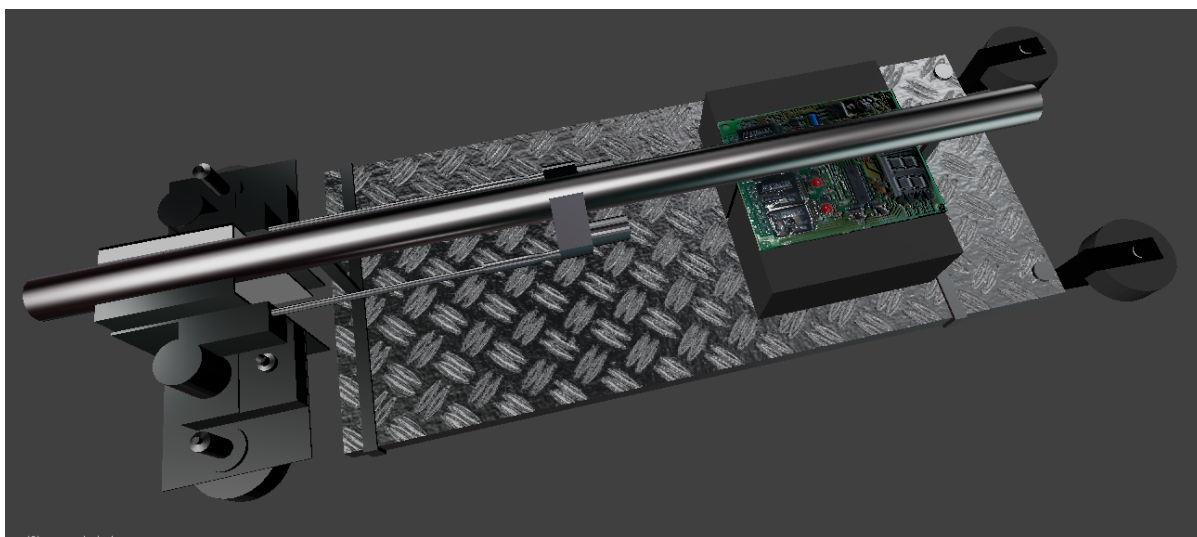


Figura 6.11: Representação gráfica 3D do Tatubot

- **Abstração semântica e modularidade:** A plataforma mostrou ser capaz de implementar as diferentes entidades previstas no modelo conceitual de um CPS, além

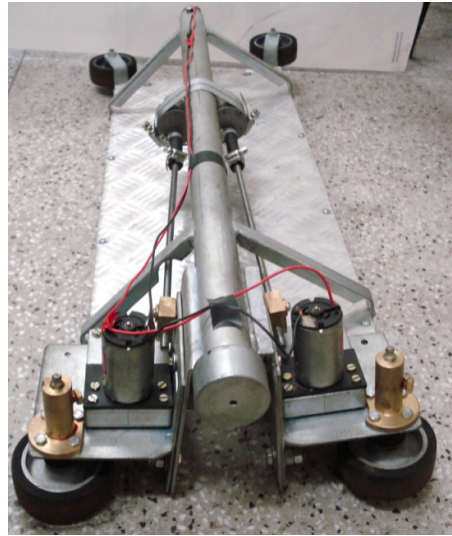


Figura 6.12: Plataforma robotica real Tatubot

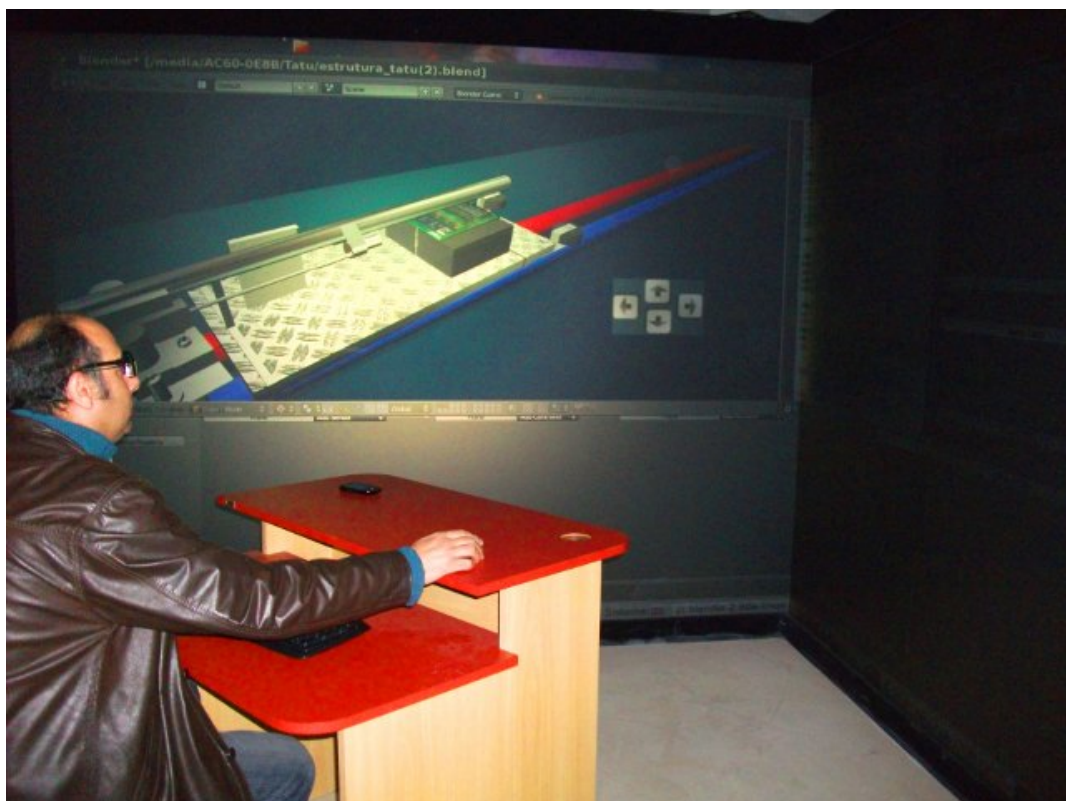


Figura 6.13: Visualização do Sistema Toogle na CAVE

de permitir tratar de forma modular e com semântica adequada as relações entre seus recursos conforme previsto no padrão hiperambiente. Isto pode ser observado

no experimento de rastreamento de insumos do Estaleiro Rio Grande, o módulo Toogle Dispositivos, possibilitaram a leitura de insumos por TAGs RFID. Através do Toogle Middleware o Editor e Navegador pode buscar as informações estruturadas pela semântica das Tags e utilizar as *Entidades Digitais* com seus serviços de controle sua localização no grid.

- **Heterogenidade:** A plataforma Toogle foi capaz de dar suporte ao grid de sensores e atuadores heterogêneos, isto foi demonstrado através do experimento do Tatubot, onde encontram-se vários dispositivos diferentes, e mesmo assim pode se obter, modificar e enviar informações de forma integrada através do middleware ROS.
- **Verificação, Validação e Simulação:** O simulador dinâmico Bullet foi utilizado para validar e certificar o comportamento das entidades físicas do CPS através de simulação física offline, sem sincronia com o robot real. Este previu comportamentos dinâmicos do mundo real, através da estrutura virtual do Tatubot e de sua interação no cenário virtual.

6.4 Considerações Finais

Os experimentos realizados através dos Casos de Estudos do Estaleiro Rio Grande e da Plataforma de Robótica Tatubot, demonstraram a aplicação do Modelo Conceitual para CPS descrito nesta dissertação, através do uso da ferramenta de Edição e Navegação de Hiperambientes Toogle, integrando os diversos dispositivos e coisas nestes cenários. Os requisitos de projeto associados a, edição e navegação, abstração semântica e modularidade, heterogenidade, verificação, validação e simulação, foram verificados e satisfeitos. A proposta permitiu desenvolver CPSs através do modelo conceitual proposto e pela ferramenta de edição e navegação de hiperambientes Toogle, integrar dispositivos heterogêneos, tratando suas informações e comportamentos, construir cenários virtuais e simulações aplicadas no mundo real de forma integrada. No próximo capítulo serão apresentadas as conclusões do trabalho e perspectivas de trabalhos futuros.

Capítulo 7

CONCLUSÃO

Finalizando o trabalho, este capítulo retoma os principais aspectos do trabalho desenvolvido, seguindo com uma discussão das principais contribuições alcançadas e, posteriormente, por sugestões de trabalhos futuros.

7.1 Resumo do Trabalho

Neste trabalho foram estudadas metodologias e ferramentas para o desenvolvimento de ambientes mistos, envolvendo entidades físicas e virtuais. O trabalho abrangeu o estudo de diferentes áreas envolvidas, focalizando-se em sistemas físicos e cibernéticos e suas aplicações.

Foi proposto um modelo conceitual para descrição de CPS o qual foi utilizado para embasar o conceito de hiperambientes. O modelo conceitual proveu o formalismo necessário para a descrição dos elementos e suas interações em um CPS. O hiperambiente resultante comporta-se como um padrão de descrição de CPS, utilizando o protocolo EEML para estruturar e transcrever as informações relacionadas, fornecendo a abstração semântica necessária ao tratamento da heterogeneidade e complexidade das relações existentes.

Uma plataforma para o desenvolvimento de CPS a partir do conceito de hiperambientes foi apresentada. A plataforma Toogle é composta de quatro módulos principais. *Toogle Dispositivo*: fornece um nível de abstração capaz de gerenciar os dispositivos de um CPS. O módulo permite a criação desta entidade no sistema, bem como a sua configuração. *Toogle Middleware*: As informações associadas as entidades físicas no hiperambiente poderão ser acessadas e modificadas através dos recursos e serviços. O toogle middleware integra os diferentes *smartobjects* permitindo a efetiva construção e navegação de hiperambientes. *Toogle Editor*: possibilita construir e editar hiperambientes, permitindo o desenvolvimento de sistemas CPS, conforme modelo conceitual apresentado. O Toogle editor permite a criação, a edição, a ligação e a remoção das informações associadas a diferentes entidades virtuais e *smartobjects* presentes em um determinado hiperambiente. *Toogle Navegador*: permite a navegação por hiperambientes de forma a acessar as informações circulantes em um CPS. Novamente em analogia a hipertextos, este módulo seria o navegador responsável pelo acesso as informações presentes neste arquivo.

A plataforma proposta atende os requisitos de projeto associados a CPS, sendo eles: edição e navegação, abstração semântica e modularidade, heterogenidade, verificação, validação e simulação.

Trabalhos Futuros Do ponto de vista conceitual e modelagem, muitos são os aspectos em aberto relacionados a CPS e que não foram tratados neste trabalho. Questões

importantes relacionadas a modelos para verificação que considerem os aspectos discretos e contínuos do tempo não fizeram parte do escopo deste trabalho. Vislumbra-se a possibilidade de extensão de Redes de Petry como formalismo para tratar tais questões.

Como trabalho futuro, relacionado a plataforma Toogle a curto termo se faz necessário finalizar o *Navegador WEB para CPS*, utilizando os novos recursos disponíveis no HTML5, demonstrados nesta dissertação, o que possibilitaria efetivamente uma interface de acesso a CPSs em dispositivos moveis tais como, smartophones e tablets, agregando ao sistema mobilidade e compatibilidade com várias plataformas, sem a necessidade de desenvolvimentos específicos para cada uma. A finalização do módulo de persistência de dados, para o editor e navegador Toogle bem como sua utilização para o Navegador WEB para CPS, com a aplicação do protocolo EEMML.

Também diferentes aplicações do Toogle estão atualmente sendo já desenvolvidas pelo nosso grupo. Destas pode-se citar:

- O Ambiente de Educação não Restritiva, associado ao Projeto PROEXT-MEC;
- Missões de Inspeção Subaquática utilizando ROV, CT-AQUA;
- Projeto SIMULAEDIF e SISCON, simulação de Estaleiros, FINEP-CT-AQUA

à utilização da plataforma nestas aplicações está conduzindo a detecção de uma série de melhorias e possíveis extensões do sistema.

Bibliografia

- [dat, 2013] (2013). acessado em 2013, <http://code.google.com/p/dat-gui/>.
- [Akella, 2013] Akella, R. (2013). Verification of information flow security in cyber-physical systems.
- [Albanowski, 2013] Albanowski, K. (2013). The industry's foundation for high performance graphics. World Wide Web, <http://www.opengl.org/>. Acessado em: jun 2013.
- [Atzori, 2010] Atzori, L., I. A. e. M. G. (2010). The internet of things: A survey. *comput. netw..* pages 54:2787–2805.
- [Baheti, 2011] Baheti, Radhakisan e Gill, H. (2011). Cyber-physical systems. *The Impact of Control Technology*, pages 161–166.
- [Bianco, 2006] Bianco, Lucio e Dell Olmo, P. e. G. S. (2006). Scheduling models for air traffic control in terminal areas. *Journal of Scheduling*, 9(3):223–253.
- [Bijin Chen e Zhiqi Xu, 2011] Bijin Chen e Zhiqi Xu, School of Computer Science e Engineering, S. C. o. U. G. P. C. (2011). A framework for browser-based multiplayer online games using WebGL and WebSocket.
- [Blender Foundation, 2012] Blender Foundation, C. (2012). Blender 3d tool. World Wide Web, <http://www.blender.org>. Acessado em: out 2012.
- [Botts, 2007] Botts, Mike, R. e. A. (2007). OpenGIS sensor model language (sensorml) implementation specification. *OpenGIS Implementation Specification OGC*, pages 07–000.

- [Bullet, 2012] Bullet (2012). Erwin coumans. World Wide Web, <http://www.bullet.org>. Acessado em: out 2012.
- [Buyya, 2009] Buyya, Rajkumar e Ranjan, R. e. C. R. N. (2009). Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. In *High Performance Computing & Simulation, 2009. HPCS'09. International Conference on*, pages 1–11. IEEE.
- [Carvalho, 2011] Carvalho, J. T. (2011). Hiperambientes: modelo conceitual e middleware para a integração de ambientes tecnológicos. - Universidade Federal do Rio Grande - FURG.
- [Clarke, 2011] Clarke, Edmund M e Zuliani, P. (2011). Statistical model checking for cyber-physical systems. In *Automated Technology for Verification and Analysis*, pages 1–12. Springer.
- [Classes, 2013] Classes, I. F. (2013). International alliance for interoperability.
- [Compton, 2012] Compton, Michael, B. P. e. B. L. e. G.-C. R. e. C. O. e. C. S. e. G. J. e. H. M. e. H. C. e. H. A. e. o. (2012). The ssn ontology of the w3c semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17:25–32.
- [des Poids et Mesures, 2006] des Poids et Mesures, B. I. (2006). The international system of units (si). ISBN 92-822-2213-6.
- [Dillon, 2010] Dillon, T. S., Z. H. W. C. S. J. e. C. E. (2010). (2011), web-of-things framework for cyber physical systems. concurrency computat. pages Pract. Exper., 23: 905?923. doi: 10.1002/cpe.1629.
- [Dirksen.js, 2013] Dirksen.js, J. (2013). Javascript 3d library, 2012. acessado em 2013, <http://threjs.org>.
- [Donassolo, 2010] Donassolo, Bruno e Casanova, H. e. L. A. e. V. P. (2010). Fast and scalable simulation of volunteer computing systems using simgrid. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, pages 605–612. ACM.

- [Du, 2011] Du, Ding-Zhu e Ko, K.-I. (2011). *Theory of computational complexity*, volume 58. Wiley. com.
- [Dunkels, 2008] Dunkels, A. e Vasseur, J. P. (September 2008). Ip for smart objects, internet protocol for smart objects (ipso) alliance, white paper 1.
- [e Joseph A. Paradiso, 2009] e Joseph A. Paradiso, J. L. (2009). Dual Reality: Merging the Real and Virtual. In *Proceedings of the First International ICST Conference on Facets of Virtual Environments (FaVE)*.
- [e Lee Edward A e Vincentelli A Sangiovanni, 2012] e Lee Edward A e Vincentelli A Sangiovanni, D. P. (2012). Modeling cyber-physical systems. *Proceedings of the IEEE*, 100(1):13–28.
- [e Mark, 2010] e Mark, P. (2010). *HTML5: Up and running*. O'Reilly.
- [e Michael Zimmer e Yooseong Kim e Hokeun Kim e Jian Cai e Aviral Shrivastava e Stephen A. Edwards e Michael Zimmer e Yooseong Kim e Hokeun Kim e Jian Cai e Aviral Shrivastava e Stephen A. Edwards e Edward A. Lee, D. B. (2013). Precision timed infrastructure: Design challenges.
- [Elmroth, 2010] Elmroth, Erik e Hernández, F. e. T. J. (2010). Three fundamental dimensions of scientific workflow interoperability: Model of computation, language, and execution environment. *Future Generation Computer Systems*, 26(2):245–256.
- [Finkenzeller, 2003] Finkenzeller, K. (2003). *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley & Sons, Inc., New York, NY, USA, 2 edition.
- [Forum, 2011] Forum, R. G. (Maio 2011). Casagras.
- [Guerra, 2008] Guerra, Rodrigo da Silva e Boedecker, J. e. M. N. e. Y. S. e. H. Y. e. Y. K. e. N. M. e. A. M. (2008). Introducing physical visualization sub-league. In Visser, Ubbo e Ribeiro, F. e. O. T. e. D. F., editor, *RoboCup 2007: Robot Soccer World Cup XI*, volume 5001 of *Lecture Notes in Computer Science*, pages 496–503. Springer Berlin / Heidelberg. 10.1007/978-3-540-68847-1-53.

- [Guinard, 2011] Guinard, Dominique e Trifa, V. e. M. F. e. W. E. (2011). From the internet of things to the web of things: Resource-oriented architecture and best practices. pages 97–129.
- [Haque, 2013] Haque, O. P. (2013). Online at <http://www.pachube.com>.
- [Ji Eun Kim, 2008] Ji Eun Kim, D. M. (2008). Generic framework for design, modeling and simulation of cyber physical systems. page Volume 5 Issue 1.
- [Joseph A. Paradiso, 2009] Joseph A. Paradiso, J. A. L. (2009). Guest editors' introduction: Cross-reality environments. *IEEE Pervasive Computing*, 8:14–15.
- [Kaiyu Wan K.L., 2010] Kaiyu Wan K.L., M. e. D. H. (2010). Towards a unified framework for cyber-physical systems (cps). pages 292–295.
- [Kortuem, 2009] Kortuem, G., K. F. F. D. S. (2009). V. (2009) smart objects as building blocks for the internet of things, internet computing, iee journal of. pages Vol.14, pp44–51.
- [Language, 2013] Language, E. E. M. (2013). Acessado em: out 2013.
- [Lee, 2008] Lee, E. A. (2008). Cps foundations - published in 2008 11th iee international symposium. pages 363–369.
- [Lee, 2013] Lee, E. A. (2013). - department of eecs uc berkeley - presentation: Cyber-physical systems:a rehash or a new intellectual challenge? Invited Talk in the Distinguished Speaker Series, sponsored by the IEEE Council on Electronic Design Automation (CEDA) held at the Design Automation Conference (DAC), Austin, Texas.
- [Li, 2010] Li, Jiaming e Poulton, G. e. J. G. (2010). Coordination of distributed energy resource agents. *Applied Artificial Intelligence*, 24(5):351–380.
- [Lifton, 2007] Lifton, J. (2007). *Dual Reality: An Emerging Medium*. Ph.D. Dissertation, Massachusetts Institute of Technology, Department of Media Arts and Sciences.
- [Lim, 2005] Lim, Hock Beng, T. Y. M. M. P. W. W. F. S. S. e. o. (2005). Sensor grid: integration of wireless sensor networks and the grid. In *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, pages 91–99. IEEE.

- [Lin, 2008] Lin, Shan e He, T. e. S. J. A. (2008). Cps-ip: cyber physical systems interconnection protocol. *SIGBED Review*, 5(1):22.
- [Lévy, 2010] Lévy, P. (2010). Towards a unified framework for cyber-physical systems (cps). acessado em 21/jun/2013 url:<http://hyperlogos.info/tiki-index.php?page=Levy>.
- [M P Papazoglou, 2005] M P Papazoglou, P Traverso, S. D. F. L. e. B. J. K. (2005). Service-oriented computing: A research roadmap. page Volume 5 Issue 1.
- [Maiden, 2005] Maiden, Neil e Robertson, S. (2005). Integrating creativity into requirements processes: Experiences with an air traffic management system. In *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*, pages 105–114. IEEE.
- [Milgram, 1994] Milgram, Paul e Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, E77-D(12).
- [Mirco Musolesi, 2008] Mirco Musolesi, e Emiliano Miluzzo, e. N. D. L. e. S. B. E. e. A. T. C. (2008). The Second Life of a Sensor: Integrating Real-world Experience in Virtual Worlds using Mobile Phones. In *Proceedings of the Fifth ACM Workshop on Embedded Networked Sensors (HotEmNets 2008)*, Charlottesville, Virginia, USA. ACM Press.
- [Neuhaus, 2009] Neuhaus, Holger e Compton, M. (2009). The semantic sensor network ontology. In *AGILE Workshop on Challenges in Geospatial Data Harmonisation, Hannover, Germany*, pages 1–33.
- [Núñez, 2013] Núñez, A. F. (2013). Um framework para prototipação de nodos físicos em hiperambientes - estudos de caso em robótica móvel. *C3-FURG*.
- [plugin for Three.js, 2013] plugin for Three.js, P. (2013). acessado em 2013, <http://chandlerprall.github.io/Physijs/>.
- [Raab, 2011] Raab, AF e Ferdowsi, M. e. K. E. e. U. I. G. e. S.-K. S. e. P. P. e. A. E. e. C. L. e. J. N. e. H. N. e. o. (2011). Virtual power plant control concepts with electric vehicles. In *Intelligent System Application to Power Systems (ISAP), 2011 16th International Conference on*, pages 1–6. IEEE.

- [Schilit, 1994] Schilit, B. e Adams, N. e. W. R. (1994). Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on*, pages 85–90.
- [Sheth, 2008] Sheth, A. ; Kno.e.sis Center, W. S. U. D. O. . H. C. . S. S. (2008). Semantic sensor web. pages 78 – 83.
- [Shi, 2011] Shi, J. (2011). Sch of phys and electron sci, shanxi datong univ, datong, china jiafu wan, hehua yan, hui suo. a survey of cyber-physical systems. pages 1–6. E-ISBN :978-1-4577-1008-7.
- [Tan, 2008] Tan, Ying e Goddard, S. e. P. L. C. (2008). A prototype architecture for cyber-physical systems. page 26.
- [Teodora Sanislav,] Teodora Sanislav, L. M.
- [Tongzhu Zhang, 2010] Tongzhu Zhang, Xueping Wang, J. C. X. L. . P. C. (2010). [29] automotive recycling information management based on the internet of things and rfid technology. pages 620–622.
- [uID Center, 2011] uID Center (Apr 2011). <http://www.uidcenter.org/>.
- [Wang, 2010] Wang, F.-Y. (2010). The emergence of intelligent enterprises: From cps to cpss. *Intelligent Systems, IEEE*, 25(4):85–88.
- [Want, 2006] Want, R. (2006). An introduction to rfid technology. *Pervasive Computing, IEEE*, 5(1):25–33.
- [Weiser, 1991] Weiser, M. (1991). The computer for the twenty-first century. *Scientific American*, 165(3):94–104.
- [Wing, 2008] Wing, J. M. (2008). Cyber-physical systems research charge. *Presentation at Cyber-Physical Systems Summit, St. Louis, MO*.
- [Xively, 2013] Xively (2013). The internet of things is open for business. acessado em Fevereiro de 2013.

- [Yinghui Huang, 2010] Yinghui Huang, e. G. L. (2010). Descriptive models for internet of things. In *Intelligent Control and Information Processing (ICICIP), 2010 International Conference on*, pages 483 –486.
- [Zhang, 2013] Zhang, Liguó e Fallah, Y. P. e. J. R. (2013). Cyber-physical systems: Computation, communication, and control.
- [Zhu, 2013] Zhu, Quanyan e Bushnell, L. e. B. T. (2013). Resilient distributed control of multi-agent cyber-physical systems. In *Control of Cyber-Physical Systems*, pages 301–316. Springer.
- [Ziemer, 1998] Ziemer, Rodger E e Tranter, W. H. e. F. D. R. (1998). *Signals and systems: continuous and discrete*, volume 4. Prentice Hall.
- [Zouganeli, 2009] Zouganeli, E. e Svinnet, I. (2009). Connected objects and the internet of things x2014; a paradigm shift. pages 1–4.