

# A Middleware Perspective for Integrating Ginga-NCL Applications with the Internet of Things

Danne Makleyston Gomes Pereira  
Universidade Federal do Maranhão – UFMA  
Av. dos Portugueses, 1966, Bacanga  
São Luís, Maranhão 65080-805  
makleyston@gmail.com

Francisco José da Silva e Silva  
Universidade Federal do Maranhão – UFMA  
Av. dos Portugueses, 1966, Bacanga  
São Luís, Maranhão 65080-805  
fssilva@ltdi.ufma.br

Carlos de Salles Soares Neto  
Universidade Federal do Maranhão – UFMA  
Av. dos Portugueses, 1966, Bacanga  
São Luís, Maranhão 65080-805  
csalles@deinf.ufma.br

Álan Lívio Vasconcelos Guedes  
Pontifícia Universidade Católica do RJ – PUC-Rio  
Rua Marquês de São Vicente, 225, Gávea  
Rio de Janeiro, Rio de Janeiro 38097  
alanlivio@gmail.com

## ABSTRACT

IoT (Internet of Things) technologies are underway. They allow the inter-connection of physical devices (also called "smart objects") that are embedded with electronics and network connectivity allowing these objects to collect and exchange data. Smart objects can be used in residential environments turning objects of everyday life (e.g. lamps, thermostats and speakers) into digital products that can emit data about its usage, location and state, can be tracked, controlled, personalized and even upgraded remotely. On the other hand, currently TV devices have extended their traditional usage by allowing applications such as web navigation, social TV and gaming. This work proposes an approach for integrating the TV with surrounding smart objects through a middleware approach, taking as its base Ginga-NCL, the Brazilian middleware for digital TV, and M-Hub, a middleware that allows the discovery, connection, communication, and data distribution of IoT smart objects. The proposed software infrastructure is evaluated in different usage scenarios that allows: (i) the TV application to modify the surrounding environment through the use of actuators, (ii) to adapt the content being presented according to the perception of the environment provided by sensors; and (iii) to provide new modes of user interaction with the TV.

## KEYWORDS

Internet of Thing; Ginga-NCL; Immersion TV

## 1 INTRODUÇÃO

Pode-se definir a Internet das Coisas (IoT) como a interconexão de dispositivos físicos (conhecidos como *smart objects*) equipados de eletrônica, *software*, sensores, atuadores e conectividade de rede que permite aos mesmos coletarem e trocarem dados, cooperando com pessoas e o ambiente [1]. Esses objetos podem ser usados em

ambientes residenciais e permitem cenários como os de monitoramento (e.g. sensores de movimento) e controle do ambiente (e.g. lâmpadas, termostatos e *speakers*). Também em ambientes residenciais, atualmente dispositivos de TV estenderam seu tradicional consumo audiovisual e permitem aplicações como navegação web, Social TV e *gaming* [2].

Nesse contexto, este trabalho explora o desenvolvimento de aplicações que cooperam aplicações de TV com *smart objects* em ambientes residenciais. Essa cooperação pode ser explorada de diversas formas. Pode-se, por exemplo, alterar aspectos do ambiente físico com base no conteúdo em exibição na TV, aumentando-se o grau de imersão do telespectador. Pode-se ainda adaptar o conteúdo em exibição na TV de acordo com a percepção do espaço físico de apresentação (por exemplo, detectando-se a ausência temporária do telespectador ou a distância do mesmo com relação à TV). Finalmente, através do uso de *smart objects* pode-se explorar novas formas de interação do telespectador com a TV.

No entanto, o desenvolvimento de uma infraestrutura de *software* que permita este grau de cooperação entre a TV e *smart objects* não é uma tarefa trivial. Diversos desafios devem ser superados, como: a descoberta dinâmica de *smart objects* e dos serviços que eles disponibilizam, uma vez que pode-se haver diversos *smart objects* presentes no ambiente físico, como também os mesmos podem oferecer serviços distintos; o suporte a heterogeneidade dos protocolos de comunicação dos *smart objects*, visto que os *smart objects* utilizam protocolos tanto baseados em IP quanto baseadas em tecnologia de curto alcance, como Bluetooth e ZigBee; e a sincronização da mídia em reprodução com os *smart objects* presentes no ambiente físico, permitindo assim a geração de imersão ao usuário.

Para superar esses desafios, este trabalho propõe uma abordagem de integração da TV com *smart objects* baseada na perspectiva do *middleware*. Neste trabalho é proposto uma infraestrutura de *software* que tem por objetivo integrar o Ginga-NCL [12], um *middleware* declarativo para desenvolvimento de aplicações de TV terrestre e IPTV [12], com o M-Hub [13], um *middleware* voltado ao domínio da IoT que permite a descoberta dinâmica, estabelecimento de conexão, acesso e distribuição de dados de/para *smart objects*.

O restante do trabalho está organizado da seguinte maneira: a Seção 2 apresenta a solução proposta deste trabalho. A Seção 3

In: XVII Workshop de Teses de Dissertações (WTD 2017), Gramado, Brasil. Anais do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web: Workshops e Pôsteres. Porto Alegre: Sociedade Brasileira de Computação, 2017.  
© 2017 SBC – Sociedade Brasileira de Computação.  
ISBN 978-85-7669-380-2.

mostra um protótipo baseado na infraestrutura de *software* proposta e cenários de uso. A apresentação dos trabalhos relacionados é exibida na Seção 4, bem como uma comparação apresentando suas limitações. A conclusão deste trabalho está exposto na Seção 5.

## 2 SOLUÇÃO PROPOSTA

Para o desenvolvimento da infraestrutura de *software* que ofereça recursos de integração e de interoperação entre os *middlewares* Ginga e M-Hub, os requisitos funcionais e não funcionais foram elencados.

### Requisitos Funcionais

- **RF1.** A infraestrutura de *software* deve oferecer suporte a diversos protocolos de comunicação dada a heterogeneidade dos *smart objects* e suas tecnologias de comunicação;
- **RF2.** A infraestrutura de *software* deve oferecer mecanismos para a descoberta dinâmica de serviços de *smart objects*, dada a possibilidade de diferentes *smart objects* estarem presentes no ambiente físico;
- **RF3.** A infraestrutura de *software* deve permitir à TV a possibilidade de alterar aspectos do ambiente físico utilizando-se dos serviços dos *smart objects*. Por exemplo, a aplicação deve ser capaz de interferir no ambiente, fazendo uso dos *smart objects* de forma sincronizada ao conteúdo audiovisual apresentado;
- **RF4.** A infraestrutura de *software* deve permitir que a aplicação de TV se adapte com base nos dados de contexto do ambiente físico. Por exemplo, os *smart objects* podem coletar dados do ambiente para interferir no conteúdo audiovisual. Mais precisamente, sensores de movimento podem ser utilizados para informar a presença e interações dos usuários.

### Requisitos Não Funcionais

- **RNF1.** A latência da comunicação entre a TV e os *smart objects* deve ser baixa. Isso se deve a sensibilidade de sincronismos de aplicações de TV. Por exemplo, uma aplicação para a TV que acione os *smart objects* de forma sincronizada com momentos de sua apresentação.
- **RNF2.** A infraestrutura de *software* deve oferecer suporte à intermitência da conectividade dos *smart objects*. Ou seja, os *smart objects* ora podem estar ativos ou dentro da área de conexão ora podem estar desativados ou fora da área de conexão.

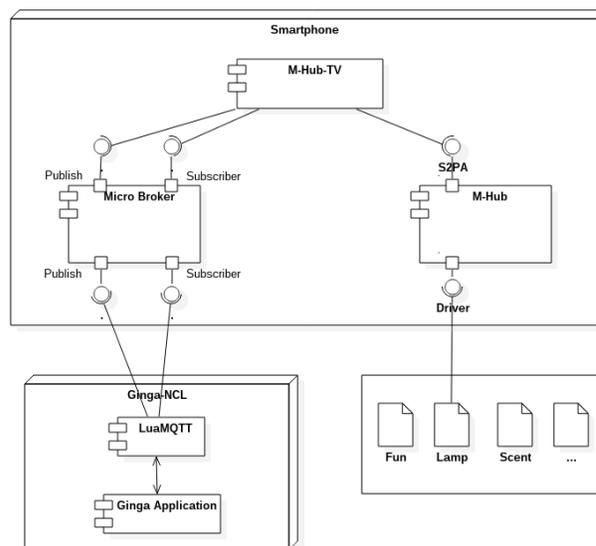
### 2.1 Infraestrutura de *software*

Neste trabalho, foi pressuposto o uso de dispositivos móveis (*e.g. smartphones e tablets*) como intermediador da comunicação entre aplicações em execução na TV e os *smart objects* presentes no ambiente físico. Esses dispositivos permitem a execução de aplicações, além de possuírem tecnologias de comunicação, como Bluetooth e wi-fi, que possibilitam uma comunicação com os *smart objects* e a TV.

Nesta infraestrutura de *software*, foi utilizado o paradigma de comunicação publicador/subscritor (*publish-subscribe*). O emprego desse modelo de comunicação permite o desacoplamento das partes que o utilizam, uma vez que os publicadores e subscritores não necessitam estabelecer uma relação de comunicação direta entre si.

Os subscritores possuem a capacidade de expressar seu interesse em um evento, enquanto que os publicadores são encarregados de registrar tais eventos. Todos os subscritores são notificados sempre que um evento, no qual possuem interesse, for registrado pelo publicador [4]. Optou-se ainda pela utilização do esquema de registros de interesses baseado em tópicos. Desta forma, os integrantes da arquitetura utilizam tópicos, que são identificados por palavras únicas, para publicar e subscrever. A utilização desse esquema permitiu a especificação de tópicos para registrar: dados sobre os *smart objects* ativos no ambiente físico, como a descrição de seus serviços; os comandos para alterações de aspectos do meio físico, encaminhados pela aplicação em execução na TV; e os dados do ambiente, captados por sensores.

Para o gerenciamento dos eventos propagados no paradigma publicador/subscritor, foi utilizado um *micro broker*. Este componente recebe os dados gerados pelo publicador e os encaminha para todos os subscritores que possuem interesse sobre esses dados. O *micro broker* utilizado na infraestrutura de *software* foi o Moquette<sup>1</sup> em virtude de ser livre de direitos autorais e leve para ser executado em um *smartphone*. Quanto ao protocolo de comunicação utilizado na troca de dados entre a aplicação em execução na TV e o *micro broker*, foi utilizado o *Message Queuing Telemetry Transport* (MQTT), por ele ser simples e leve, além de ter sido projetado para dispositivos restritos e redes de baixa largura de banda, alta latência ou não confiáveis [9]. A Figura 1 apresenta a infraestrutura de *software* proposta.



**Figure 1: Infraestrutura de *software* proposta para a troca de dados entre a TV e os *smart objects***

O M-Hub, como já mencionado, é um *middleware* voltado à IoT que possui a capacidade de descobrir e se comunicar com *smart objects* próximos, por meio de tecnologias de comunicação de curto alcance. Ele é um nó especial na rede atuando como um *gateway*

<sup>1</sup><http://andse1.github.io/moquette/>

## A Middleware Perspective for Integrating Ginga-NCL Applications with the Internet of Things

no qual os objetos se conectam para enviar os dados capturados do ambiente para a Internet [13]. O *Shot-range Sensing, Presence & Actuation* (S2PA) é um módulo funcional do M-Hub capaz de realizar varreduras periódicas no ambiente em busca de serviços de *smart objects* e estabelecer uma comunicação com os mesmos, assim é possível realizar leituras de dados dos sensores ou escrever comandos nos atuadores, por exemplo. Para a realização da troca de dados com os *smart objects* é necessário a utilização do *driver* do objeto que deverá ser fornecido pelo seu fabricante. Diante dessas características, o M-Hub foi utilizado neste projeto.

Uma aplicação voltada para o dispositivo móvel, denominada de Mobile Hub TV (M-Hub-TV), foi desenvolvida para consumir os dados e descrições de serviços que o M-Hub capta dos *smart objects* presentes no ambiente, além de realizar a publicação desses dados em tópicos no *micro broker*. O M-Hub-TV, consome também, os dados publicados, pela aplicação da TV, nos tópicos definidos no *micro broker* e os encaminha para o M-Hub poder enviá-los aos *smart objects*. Ainda, o M-Hub-TV é capaz de processar e realizar filtros sobre os dados oriundos dos sensores antes de serem publicados no *micro broker*.

O *micro broker*, o M-Hub e o M-Hub-TV estão contidos no *smartphone*. Por outro lado, no ambiente de apresentação do *middleware* Ginga, está contido outro componente da infraestrutura de *software*, um objeto Lua, denominado LuaMQTT. A aplicação NCL também faz uso desse ambiente de apresentação.

O LuaMQTT é responsável por subscrever nos tópicos definidos no *micro broker* para receber os dados dos sensores do ambiente, como também é incumbido de publicar no *micro broker* dados oriundos da aplicação Ginga-NCL. A troca de dados entre a aplicação NCL e o *micro broker* ocorre por meio do canal de retorno presente no SBTVD.

### 3 CENÁRIOS DE USO

Fazendo uso da solução proposta, foram desenvolvidos três cenários de uso. O primeiro cenário consiste em uma aplicação para alterar aspectos do ambiente físico de apresentação. O segundo permite a alteração do conteúdo em exibição mediante uma análise dos dados capturados pelos *smart objects* presentes no ambiente físico. O último permite ao usuário interagir com a aplicação em execução na TV por meio de gestos.

Para o desenvolvimento dos cenários foi utilizado a seguinte configuração de equipamentos: um notebook, um televisor, dois *smart objects* e um *smartphone*. No notebook foi executado um emulador do Ginga, versão 0.12.4, e o mesmo foi conectado ao televisor para exibir as reproduções de mídias realizadas pelo emulador. O *smartphone* utilizado possui um processador quad-core de 1,2 GHz, 1 GB de memória e seu sistema operacional é o Android 6.0. Nesse dispositivo, foi instalado a aplicação M-Hub-TV, o *micro broker* e o M-Hub. Os *smart objects* foram desenvolvidos utilizando a plataforma de prototipagem eletrônica Arduino<sup>2</sup> e componentes eletrônicos. Um dos *smart objects* possui um *dimmer* para gerenciar a intensidade da iluminação de uma lâmpada e um módulo de comunicação Bluetooth, enquanto que o outro objeto possui um sensor de presença e também um módulo de comunicação Bluetooth.

<sup>2</sup>www.arduino.cc

WebMedia'2017: Workshops e Pôsteres, WTD, Gramado, Brasil

A implementação dos casos de uso compartilham o mesmo fragmento do código NCL ilustrado pela Listagem 1. Este código utiliza três elementos <media>: um vídeo (linhas 1-6), uma imagem (linha 7) e o *script* LuaMQTT (linhas 8-12).

```

1 <media id="movie" src="movie.mp4" descriptor="
  descriptorMovie">
2   <area id="a0" begin="21s"/>
3   <area id="a1" begin="34s"/>
4   <area id="a2" begin="36s"/>
5   <area id="a3" begin="40s"/>
6 </media>
7 <media id="pause" src="pause.png" descriptor="
  descriptorPause"/>
8 <media id="LuaMQTT" src="LuaMQTT.lua">
9   <area id="a_out" label="a_out"/>
10  <area id="a_in" label="a_in"/>
11  <property name="p_value"/>
12 </media>

```

Listagem 1: Fragmento do código NCL

#### 3.1 Cenário de Uso 1: Alteração dos aspectos do ambiente físico por meio da aplicação NCL

Para a realização do primeiro caso de uso, foi adicionado as áreas no elemento <media> do vídeo (linhas 2-5) que representam os instantes em que a aplicação deve enviar comandos para o acionamento da lâmpada. O elemento <media> do objeto Lua também possui duas áreas definidas e uma propriedade (linhas 9-11). A área cujo label é "a\_in" e a propriedade "p\_value" (linhas 10-11) são interligadas por meio do elemento <link> com as áreas do elemento <media> do vídeo. Sempre que uma área do vídeo é inicializada (início definido pelo atributo begin), é enviado um parâmetro em estrutura de chave-valor, informando o nome do serviço a ser utilizado e um valor para a utilização desse serviço (e. g. lamp#25%), para a propriedade "p\_value" e um comando de start à área "a\_in". O objeto Lua recebe o parâmetro e o publica em um tópico no *micro broker*. A aplicação M-Hub-TV recebe os dados publicados nesse tópico e os envia para o M-Hub. Por fim, o M-Hub, por meio do S2PA, encaminha os dados para o *smart object* responsável por controlar a luminosidade, que atuará no ambiente. O resultado do cenário de uso 1 é apresentado na Figura 2.

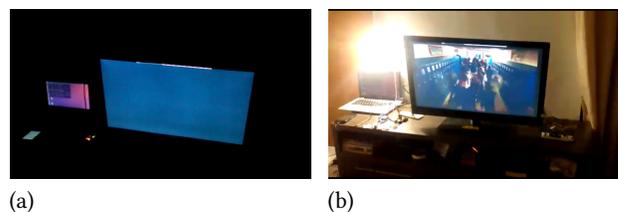


Figure 2: Alteração dos aspectos do ambiente físico por meio da aplicação Ginga-NCL

A imagem (a) da Figura 2 apresenta o resultado após a aplicação NCL enviar comandos para alterar a intensidade da lâmpada para zero (sem iluminação). Por outro lado, a imagem (b) da Figura 2 apresenta o resultado após a aplicação NCL enviar comandos para aumentar a intensidade da lâmpada. Assim, este caso de uso

demonstra a possibilidade da aplicação NCL alterar os aspectos do ambiente físico com base no conteúdo apresentado.

### 3.2 Cenário de Uso 2: Alteração no conteúdo em exibição após análise do contexto do ambiente físico

Neste cenário de uso, o *smart object*, capaz de detectar movimentos no ambiente, foi utilizado. O sensor de presença do *smart object* foi fixado na TV para uma maior precisão dos dados com relação a mesma. O M-Hub realiza uma varredura no ambiente em busca de serviços de *smart objects*. Após detectar o *smart object*, o mesmo se conecta e realiza uma leitura dos dados obtidos pelo sensor de presença. Após a captura desses dados, o M-Hub os encaminha para o M-Hub-TV, que por sua vez os processam com base em regras preestabelecidas, como, por exemplo, se o usuário não se movimentar por um determinado período, então deve-se publicar comandos em um tópico no *micro broker* para interromper a reprodução do vídeo. Por outro lado, o objeto Lua da aplicação NCL se inscreve no mesmo tópico em que o M-Hub-TV publica os dados do sensor. Assim, quando houver uma publicação de comandos para interrupção do vídeo, o objeto Lua gera um evento com ação *pause* na área cujo label é "a\_out" (linha 9). O "a\_out" é interligado por meio de um elemento <link> à mídia de imagem e mídia de vídeo utilizando o conector *onPauseStartPause*. A exibição da imagem é inicializada e uma interrupção no vídeo é realizada após o disparo deste evento. Se o sensor detectar sinais de movimentos no ambiente e a exibição do vídeo estiver no estado *pause*, então um evento *resume* é gerado, pelo objeto Lua, na área "a\_out". Com este evento, a imagem é finalizada e o vídeo retoma ao ponto interrompido. A imagem 3 apresenta o resultado deste cenário de uso.

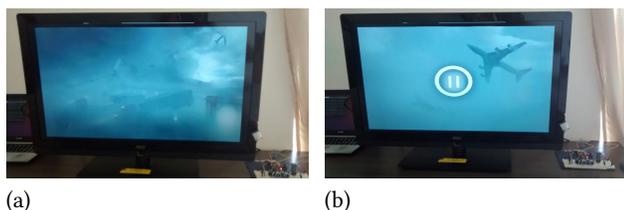


Figure 3: Alteração do conteúdo após análise do contexto do ambiente físico

A imagem (a) da Figura 3 apresenta a execução do conteúdo sem interrupção por parte da aplicação, uma vez que foi detectado movimentos no ambiente. Por outro lado, a imagem (b) da Figura 3 mostra o instante em que a reprodução do vídeo é interrompida, devido não ter sido detectado algum movimento no ambiente físico de apresentação. Assim, este caso de uso demonstrou a possibilidade do conteúdo em reprodução na TV ser modificado com base no contexto físico de apresentação.

### 3.3 Cenário de Uso 3: Interação do usuário com a aplicação em execução na TV por meio de gestos

Neste cenário de uso, o usuário utilizou o *smartphone* para enviar comandos gerados por gestos à aplicação em execução na TV. O

gesto definido neste cenário de uso, foi a movimentação suave e alternada do *smartphone* gerada por um aceno com a mão do usuário. O M-Hub monitora os dados dos sensores presentes no dispositivo e ao movimentar o *smartphone*, o mesmo captura os dados do acelerômetro e os envia para o M-Hub-TV. Por sua vez, o M-Hub-TV analisa os dados e verifica sua compatibilidade com o gesto definido. Posteriormente, caso seja detectado o movimento descrito, é publicado em um tópico do *micro broker* um comando para interrupção do conteúdo exibido na TV.

Por outro lado, o objeto Lua, que está subscrito no mesmo tópico em que o M-Hub-TV publica os dados, recebe a informação de interrupção do conteúdo apresentado e dispara um evento na área "a\_out" (linha 9). As ligações entre os objetos de mídia do vídeo, da imagem e do objeto Lua são as mesmas dispostas no cenário de uso 2. Assim, a mídia de imagem é inicializada, enquanto a mídia do vídeo é interrompida. Caso o usuário refaça o gesto com o *smartphone*, o conteúdo de vídeo volta a ser executado e a imagem é finalizada.

A Figura 3 apresenta o mesmo resultado alcançado por este cenário de uso. Assim, foi demonstrado por este cenário de uso a possibilidade do conteúdo em reprodução na TV ser modificado com base na interação do usuário.

### 3.4 Discussão

Através dos Cenários de Uso descritos, exploramos como a infraestrutura de *softwares* proposta para a integração dos *middlewares* Ginga e M-Hub permite atingir os requisitos funcionais e não funcionais especificados na seção 2. O cenário de uso 1 permitiu explorar a descoberta dinâmica de *smart objects* e os serviços por eles disponibilizados (RF2), bem como a intervenção no espaço físico de apresentação através de comandos enviados pela aplicação NCL em execução na TV para o atuador de luminosidade do ambiente (RF3). Adicionalmente, o cenário de uso 2 demonstrou como que a percepção do ambiente físico, obtida através de sensores, pode ser exportada para o Ginga de forma a influenciar a exibição de conteúdo na TV (RF4). No exemplo ilustrado, um sensor de presença foi utilizado para suspender e reiniciar a apresentação de um vídeo automaticamente de acordo com a presença ou não do telespectador. Finalmente, o cenário de uso 3 demonstra como a infraestrutura de *software* desenvolvida pode ser utilizada para prover novos meios de interação do usuário com a TV (RF4). No exemplo implementado, um mecanismo de gestos baseado na interpretação dos dados do sensor de acelerômetro do *smartphone* foi utilizado para suspender e reiniciar a apresentação de um vídeo na TV. Através do serviço S2PA presente no M-Hub, que abstrai a tecnologia de comunicação utilizada pelos *smart objects* por meio de uma interface padrão para a interação com sensores e atuadores, o RF1 é contemplado. Além das tecnologias de comunicação nativamente suportadas pelo M-Hub, a adição de outras tecnologias pode facilmente ser realizada.

Finalmente, com relação aos requisitos não funcionais, todas a comunicação entre a TV e os *smart objects* é realizada no âmbito de redes locais, que possuem baixa latência. A comunicação entre o Ginga e o M-Hub é realizada através de uma rede local wi-fi, enquanto que a comunicação entre o M-Hub e os *smart objects* varia de acordo com a tecnologia de comunicação de curto alcance utilizada pelo *smart object*.

A Middleware Perspective for Integrating Ginga-NCL Applications with the Internet of Things

## 4 TRABALHOS RELACIONADOS

Outros trabalhos também compartilham de nossa motivação de permitir a integração entre dispositivos de TV e *smart objects*. Esses trabalhos são organizados a seguir em três contextos.

### 4.1 Mecanismos de interação do usuário com a aplicação em execução na TV digital

Pedrosa et. al em [10] tem como objetivo apresentar e validar um componente, denominado *Multimodal Interaction Component - MMIC*, capaz de receber dados multimodais de diferentes dispositivos e reproduzi-los em uma TV. A comunicação do MMIC com objetos presentes no ambiente é realizado por protocolos de comunicação que não necessitam de configurações explícitas, como: IP (*ZeroConf*) e UPnP. A comunicação entre o dispositivo e a TV é realizada por meio de tabelas da linguagem Lua, que por sua vez interagem com a linguagem multimídia NCL. Foi criada uma interface multimodal (*IMultimodalInputEvent*) contendo campos que permitem o reconhecimento de dados de tinta digital, acelerômetro, voz, imagem, etc. para receber os eventos originados de diversos aparelhos, como *smartphones*, *tablets* e captadores de áudios e imagens. Por fim, um protocolo em formato XML foi desenvolvido para realizar a comunicação entre o MMIC e o Ginga.

### 4.2 Uso da TV digital como interface para o gerenciamento de objetos em ambientes de automação residencial

Nairon et. al. [15] propõem uma arquitetura, denominada de Ginga-OSGi, que permite o registro e a descoberta de serviços na rede OSGi, bem como oferece mecanismos que possibilitam uma comunicação bidirecional entre os dispositivos da rede OSGi e a TV. Desta maneira, aplicações em NCL podem controlar dispositivos presentes no ambiente. Uma rede OSGi possui um *gateway* que identifica os dispositivos presentes na rede e grava dados desses dispositivos em uma pilha de registros. A arquitetura dispõe de um módulo que gera um arquivo XML contendo as descrições dos dispositivos presentes na pilha de registros. Uma aplicação na TV realiza a leitura desse arquivo para apresentar seus parâmetros ao usuário, por exemplo. Quando se deseja que um determinado dispositivo realize alguma tarefa, a aplicação NCL envia parâmetros para o módulo responsável pela geração de arquivo XML. O arquivo gerado é lido pelo *gateway* OSGi que em seguida aciona o dispositivo desejado.

Dmitry et. al. [14] também apresentam cenários que ilustram a interação entre TV e redes domésticas baseadas em OSGi. Os cenários utilizam os *middlewares Multimedia Home Platform* (MHP) e *Digital Television Application Software Environment* (DASE) dos padrões europeu e norte-americano de TV digital, respectivamente. Para o trabalho, foi considerada uma arquitetura de sistema na qual as aplicações e serviços destinados à rede OSGi são entregues via radiofrequência junto ao carrossel de dados do sinal da TV digital. Desta forma, o receptor, ao receber os dados da emissora que contenham alguma solicitação de execução de serviços de dispositivos, envia-os para o *gateway* da rede OSGi, que encaminha ao respectivo dispositivo para ser executado.

WebMedia'2017: Workshops e Pôsteres, WTD, Gramado, Brasil

### 4.3 Favorecimento do processo de imersão do usuário ao conteúdo apresentado na TV digital através de *smart objects*

Saleme et. al.[11] propõem que aplicações de TV usem efeitos sensoriais definidos no padrão MPEG-V SEM [8] com marcações de tempo relativos ao vídeo. Então, o *middleware* das aplicações de TV seria encarregado de enviar comandos MPEG-V para um dispositivo Arduino através de um serviço UPnP. Em seus experimentos, o dispositivo Arduino era capaz de apresentar efeitos sensoriais que controlam uma lâmpada, e atuadores de vibração e vento.

Guedes et. al. [5] apresentam uma abordagem de interação entre dispositivos IoT em sincronismo ao conteúdo apresentado na TV brasileira. Sua proposta utiliza o ambiente declarativo Ginga-NCL, do *middleware* Ginga, para enviar comandos a um dispositivo Intel Galileo<sup>3</sup> com o kit *Seeed Studio Grove*<sup>4</sup>. O trabalho objetiva gerar uma imersão ao usuário por meio da atuação dos equipamentos eletrônicos presentes na residência do usuário. A comunicação entre o Galileo e os equipamentos eletrônicos se dá por meio de um contato físico obtido por fios. A proposta faz uso de uma programação de *scripts* em Lua que recebe eventos da linguagem NCL para acionar os objetos com base na reprodução da mídia em execução. Nessa arquitetura, os dispositivos devem ser previamente conhecidos pelo componente gerenciador (Galileo) e também devem ser especificados na linguagem NCL.

### 4.4 Discussão

Uma comparação entre os trabalhos apresentados foi realizada e um resumo da mesma é apresentada na Tabela 1. A comparação foi desenvolvida considerando como critérios: (i) a descoberta de *smart objects*, no qual é especificado qual tecnologia foi utilizada, (ii) o suporte a heterogeneidade dos *smart objects*, (iii) a possibilidade de alterar aspectos do ambiente físico utilizando os recursos dos serviços dos *smart objects*, (iv) a possibilidade de refletir alterações no conteúdo exibido com base nos dados de contexto do ambiente físico e (v) a possibilidade de interação multimodal por parte do usuário.

	(i)	(ii)	(iii)	(iv)	(v)
Pedrosa et. al. [10]	-	Sim	-	-	Sim
Nairon et. al. [15]	OSGi	-	Sim	Sim	-
Dmitry et. al. [14]	OSGi	-	Sim	Sim	-
Guedes et. al. [5]	-	-	Sim	Sim	-
Saleme et. al. [11]	UPnP	-	Sim	-	-
Proposta	Diversos	Sim	Sim	Sim	Sim

Table 1: Tabela comparativa dos trabalhos relacionados

Diante dos trabalhos apresentados, percebe-se que o trabalho [5] e [11] objetivam gerar efeitos sensoriais com intuito de permitir imersão ao usuário. Guedes et. al. [5] apresentam uma estrutura limitada aos dispositivos pré-definidos. Enquanto que Saleme et. al. [11] utilizam UPnP para descobrir *smart objects* próximos. Ambos os trabalhos não oferecem recursos para uma interação multimodal

<sup>3</sup><https://intel.com/content/www/us/en/do-it-yourself/galileo-maker-board.html>

<sup>4</sup>[https://www.seeedstudio.com/item\\_detail.html?p\\_id=1978](https://www.seeedstudio.com/item_detail.html?p_id=1978)

do usuário com a aplicação em exibição. Por outro lado, os trabalhos [14] e [15] utilizam uma rede OSGi para utilizar serviços na rede, no entanto, os *smart objects* devem permitir a execução de componente em Java, o que poderia limitar a quantidade de *smart objects* utilizáveis pela aplicação na TV. O trabalho [10] apresenta pontos interessantes quanto à utilização sem fio dos objetos para a entrada de dados na TV. No entanto, o usuário deve se conectar ao MMIC, ou seja, o mecanismo não faz a detecção e não estabelece uma comunicação com os *smart objects* automaticamente. Por outro lado, permite que diversos dispositivos com protocolos de comunicação distintos possam ser utilizados no MMIC.

Em contrapartida, nossa infraestrutura de *software* utiliza o *middleware* M-Hub para descobrir serviços de *smart objects* próximos e para realizar uma comunicação com os mesmos, fazendo uso de diversos protocolos de comunicação. Esta infraestrutura de *software*, permite ainda uma comunicação bidirecional entre a TV e os *smart objects*. Sendo possível a alteração de aspectos do espaço físico por meio de aplicações da TV, como também possibilita o recebimento de dados de sensores do ambiente para as aplicações Ginga-NCL. A utilização de sensores capazes de captar áudios e movimentos, por exemplo, permitem que o usuário interaja com a aplicação Ginga por meio de comandos de voz e gestos, respectivamente. Diversos outros sensores podem ser utilizados para captar modos de interação do usuário. Ou seja, para a aplicação NCL utilizar os dados de qualquer sensor, basta que o M-Hub-TV publique os comandos oriundos desses sensores no *micro broker*.

## 5 CONCLUSÃO

Este trabalho objetivou a integração entre aplicações Ginga-NCL e dispositivos da IoT. Foi apresentada uma infraestrutura de *softwares* que integra e interopera os *middlewares* Ginga, por meio de seu ambiente de apresentação Ginga-NCL, e M-Hub, *middleware* voltado à IoT.

A utilização dessa infraestrutura de *softwares* permite que aplicações em execução na TV possam trocar dados com *smart objects* dispostos no ambiente físico de apresentação. Desta forma, é possível controlar atuadores e receber dados de sensores. O recebimento de dados oriundos de sensores, permite ainda, que o usuário interaja de forma multimodal com a aplicação em execução na TV.

Foi implementado um protótipo e apresentado três casos de usos, nos quais seus objetivos foram: alterar aspectos do ambiente em sincronismo com o conteúdo em exibição; refletir alterações no conteúdo apresentado com base nos dados de contexto do ambiente; e permitir ao usuário a interação com a aplicação em execução na TV por meio de gestos. Com a execução dos cenários de usos, foi possível observar que o protótipo contemplou todos os requisitos listados neste trabalho.

Alguns caminhos podem ser seguidos para estender este trabalho, sendo alguns listados por nós como trabalhos futuros. Inicialmente, devido a falta de primitivas na linguagem NCL que permita a troca de dados entre aplicações em NCL e *smart objects*, pretendemos estender o trabalho de Guedes et. al. [6], incorporando o nossa infraestrutura de *softwares* às extensões apresentadas por eles. Um outro ponto a ser trabalhado posteriormente está associado ao modelo de comunicação entre a aplicação Ginga-NCL e o *middleware* M-Hub. Pretendemos adicionar ao mecanismo publicador/subscritor

orientado a tópicos o uso de uma ontologia específica da área de sensores e atuadores, como a M3 [7] e SSN [3], o que permitirá a adoção de um vocabulário comum e padrão para a especificação dos serviços e dados dos *smart objects*. Por fim, pretendemos implementar um módulo de Processamento de Eventos Complexos (*Complex Event Processing* - CEP) no M-Hub-TV. Este módulo facilitará o processamento dos dados oriundos dos sensores presentes no espaço físico, dado que as linguagens empregadas por motores CEP permitem a especificação de regras para a descoberta de padrões nos fluxos de eventos de acordo com restrições de causalidade e temporalidade. Desta forma, o mecanismo de reconhecimento de gestos, por exemplo, poderia ser baseado em regras CEP.

## REFERENCES

- [1] Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The Internet of Things: A survey. *Computer Networks* 54, 15 (2010), 2787 – 2805. <https://doi.org/10.1016/j.comnet.2010.05.010>
- [2] Pablo Cesar, Konstantinos Chorianopoulos, et al. 2009. The evolution of TV systems, content, and users toward interactivity. *Foundations and Trends® in Human-Computer Interaction* 2, 4 (2009), 279–373. <https://doi.org/10.1561/1100000008>
- [3] C. Michael Compton, Payam Barnaghi, et al. 2012. The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web* 17 (2012), 25 – 32. <https://doi.org/10.1016/j.websem.2012.05.003>
- [4] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Ker-marrec. 2003. The Many Faces of Publish/Subscribe. *ACM Comput. Surv.* 35, 2 (June 2003), 114–131. <https://doi.org/10.1145/857076.857078>
- [5] Alan LV Guedes, Marcio Cunha, Hugo Fuks, Sergio Colcher, and Simone DJ Barbosa. 2016. Using NCL to Synchronize Media Objects, Sensors and Actuators. In *XXII Simpósio Brasileiro de Sistemas Multimídia e Web (Vol. 2): Workshops e Sessão de Pôsteres*. (Teresina). Webmedia, 184–189. Porto Alegre: Sociedade Brasileira de Computação.
- [6] Alan LV. Guedes, Roberto G. de Albuquerque Azevedo, Sérgio Colcher, and Simone D.J. Barbosa. 2016. Extending NCL to Support Multiuser and Multimodal Interactions. In *Proceedings of the 22Nd Brazilian Symposium on Multimedia and the Web (Webmedia '16)*. ACM, New York, NY, USA, 39–46. <https://doi.org/10.1145/2976796.2976869>
- [7] A. Gyrard, C. Bonnet, and K. Boudaoud. 2014. Enrich machine-to-machine data with semantic web technologies for cross-domain applications. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*. 559–564. <https://doi.org/10.1109/WF-IoT.2014.6803229>
- [8] ISO/IEC. 2016. ISO/IEC 23005-3:2016 - Information technology - Media context and control - Part 3: Sensory information. (2016). [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=65396](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=65396)
- [9] MQTT. 2017. MQTT. <http://mqtt.org>. (2017). Accessed: 2017-04-16.
- [10] Diogo Pedrosa, José Augusto C. Martins, Jr., Erick L. Melo, and Cesar A. C. Teixeira. 2011. A Multimodal Interaction Component for Digital Television. In *Proceedings of the 2011 ACM Symposium on Applied Computing (SAC '11)*. ACM, New York, NY, USA, 1253–1258. <https://doi.org/10.1145/1982185.1982459>
- [11] Estêvão Bissoli Saleme and Celso Alberto Saibel Santos. PlaySEM: A Platform for Rendering MulSeMedia Compatible with MPEG-V. In *Proceedings of the 21st Brazilian Symposium on Multimedia and the Web (2015) (WebMedia '15)*. ACM, 145–148. <https://doi.org/10.1145/2820426.2820450>
- [12] L. F. Gomes Soares, M. F. Moreno, C. D. Salles Soares Neto, and M. F. Moreno. 2010. Ginga-NCL: Declarative middleware for multimedia IPTV services. *IEEE Communications Magazine* 48, 6 (June 2010), 74–81. <https://doi.org/10.1109/MCOM.2010.5473867>
- [13] L. E. Talavera, M. Endler, I. Vasconcelos, R. Vasconcelos, M. Cunha, and F. J. d. S. e. Silva. 2015. The Mobile Hub concept: Enabling applications for the Internet of Mobile Things. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. 123–128. <https://doi.org/10.1109/PERCOMW.2015.7134005>
- [14] D. Tkachenko, N. Kornet, and A. Kaplan. 2004. Convergence of iDTV and home network platforms. In *First IEEE Consumer Communications and Networking Conference, 2004. CCNC 2004*. 624–626. <https://doi.org/10.1109/CCNC.2004.1286935>
- [15] Nairon S Viana, Orlewilson B Maia, Vicente F de Lucena Júnior, and Luciano C Pinto. 2009. A convergence proposal between the Brazilian middleware for iDTV and home network platforms. In *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*. IEEE, 1–5.