Abordagem semântica para o desenvolvimento de sistemas de suporte a ambientes inteligentes inclusivos

Felipe Morine Magami[1], José J. Pérez-Álcazar[1], Fábio Nakano[1], Andre L. M. Silva[2], João S. de Oliveira Neto[3], Sergio T. Kofuji[4]

[1]Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
São Paulo – SP – Brazil

[2]Departamento de Sistemas de Informação – Universidade Federal de Sergipe (UFS)
Itabaiana – SE – Brazil

[3]Centro de Ciências Exatas e Tecnológicas – Universidade Federal do Recôncavo da Bahia (UFRB)
Cruz das Almas – BA – Brazil

[4]Lab. de Sistemas Integráveis – Escola Politécnica – Universidade de São Paulo (USP)
São Paulo – SP – Brazil

{felipe.magami, jperez, fabionakano, andrelumesi, j.neto, kofuji}@usp.br

Abstract. *There are hundreds of millions of visually impaired people around the world, individuals that in a daily basis face difficulties related to their mobility. The progress of mobile technologies brings more opportunities to create applications to aid the visually impaired. This paper proposes an approach based on semantics that organizes, relates and assign meaning to information of the urban space, in order to provide content tailored to people with disabilities. In order to validate this proposal, we have created an indoor navigation system for shopping mall to support the visually impaired.*

Resumo. *Há centenas de milhões de deficientes visuais no mundo, indivíduos que enfrentam diariamente dificuldades relacionadas à sua mobilidade. O avanço das tecnologias móveis traz mais oportunidades para a construção de aplicativos para apoio ao deficiente visual. Esse trabalho propõe uma abordagem baseada em semântica que organiza, relaciona e atribui significado a informações do espaço urbano, com intuito de prover conteúdo adaptado às pessoas com deficiência. Para validação dessa proposta [1], criamos um sistema de navegação em* shopping center *para apoio a usuários com deficiência visual.*

1. Introdução

Cerca de 15% da população mundial possui algum tipo de deficiência sensorial, motora ou cognitiva [World Bank 2016]. Ao longo do século XX, a luta destes indivíduos e de associações possibilitou a ampla discussão das condições de vida a que eram submetidos e, como consequência, uma série de direitos foram adquiridos e políticas públicas voltadas para as pessoas com deficiência (PcD) foram definidas.

[1]Agradecimentos especiais a HUAWEI pelo apoio dado a este projeto de pesquisa no âmbito do convênio HUAWEI-USP.

O mercado e os grupos de pesquisas deveriam se sensibilizar mais com as questões relacionadas à acessibilidade dado que todos nós experimentaremos algum tipo de deficiência ao longo da vida, ou até mesmo várias deficiências coexistentes ao envelhecermos [World Health Organization et al. 2011]. As PcD continuam a ser discriminadas e seu acesso ao ambiente profissional e aos serviços de saúde, educação e lazer ainda é limitado. Para aumentar a presença de PcD na sociedade em geral e diminuir o isolamento dessas pessoas, surgiram as tecnologias assistivas [Federici and Scherer 2012]: produtos e serviços que ajudam a mitigar a limitação da PcD e/ou as barreiras impostas pelo ambiente: bengalas, cadeiras de roda motorizadas, aparelhos auditivos, pisos táteis ao longo de calçadas, teclados de computadores adaptados, leitores de tela de computador, entre outros.

Apesar dos avanços trazidos pelo conjunto de tecnologias assistivas disponíveis, alguns cenários ainda são fonte de apreensão e incerteza para as PcD, como o espaço urbano. É na cidade que acontece a maior parte das interações humanas da sociedade contemporânea: relações econômicas, sociais, políticas, humanas, etc. Mesmo que as PcD tenham conseguido o direito e os meios para frequentar a universidade ou exercer uma atividade profissional, o deslocamento de seus domicílios até a sala de aula ou até a empresa é uma verdadeira *via crucis* [de Oliveira Neto and Kofuji 2016]: calçadas tomadas por entulho dificultando a locomoção, alertas sonoros que não são percebidos por surdos, textos em placas de direção e rotas que não são acessíveis a cegos, entre outros exemplos. Infelizmente, as cidades são projetadas e construídas para os cidadãos não-portadores de deficiência em detrimento de uma parcela significativa da população.

Por outro lado, graças aos avanços da Computação Pervasiva, a cidade passou a ser vista como um grande sistema de sistemas [Nam and Pardo 2011], onde a tecnologia tem ajudado instituições governamentais e privadas a oferecem melhores serviços aos cidadãos. Surgiram assim as Cidades Inteligentes: iniciativas que buscam de uma maneira inovadora otimizar o acesso aos serviços de transporte, saúde, educação, segurança e comunicação visando proporcionar uma melhor qualidade de vida aos moradores [Alawadhi et al. 2006].

Este trabalho tem como objetivo oferecer à comunidade de desenvolvedores de soluções para Cidades Inteligentes uma abordagem apoiada em semântica que organiza as informações que estão disponíveis no espaço urbano, a fim de criar relações entre estas informações, dotar de sentido essa informação e prover as PcD de um conteúdo adaptado às suas habilidades. Esta abordagem provê às PcD uma forma diferente de navegar pela cidade e de interagir com equipamentos urbanos – placas, prédios, semáforos, paradas de ônibus etc. Para validar essa proposta, foi construída uma aplicação que apoia PcD a se locomoverem e chegarem a uma determinada localização dentro de um *shopping center*.

O restante deste artigo é organizado da seguinte forma: na Seção 2 explicaremos sobre os fundamentos e tecnologias que nos nortearam para a conclusão do trabalho. A Seção 3 coloca em evidência obras com temas similares e soluções para problemas relacionados ao nosso. A solução desenvolvida, bem como sua avaliação, se encontram nas Seções 4 e 5, respectivamente. E finalmente, nossas considerações finais, na Seção 6.

2. Fundamentação teórica

A web semântica estende a web tradicional de forma que a informação disponível seja acompanhada por significado. A semântica incluída permite que a informação possa ser processada e interpretada computacionalmente. Dessa forma, máquinas desempenham um papel mais ativo no consumo de dados e não apenas na transmissão de informação para consumo humano [Fensel et al. 2011]. Em nossa proposta, o suporte semântico foi dado pelo emprego de WSML.

WSML (Web Service Modeling Language) [Fensel et al. 2011] é uma linguagem formal para descrever serviços web semânticos baseada em formalismos lógicos, tais como lógica de descrições, lógica de primeira ordem e lógica de programação. No âmbito deste trabalho, fizemos uso dos seguintes elementos WSML: ontologias, serviços semânticos e metas.

A Figura 1 apresenta um exemplo de ontologia definida em WSML. As linhas 1 e 2 definem, respectivamente, a variante WSML adotada e o *namespace* da ontologia. Em seguida, na linha 3, é definida a ontologia *SeresVivos*. As linhas 6 a 10 são utilizadas para a definição de classes. Em WSML, uma classe é definida através da palavra reservada *concept*. A classe Humano é composta pelos atributos nome do tipo *_string* e idade do tipo *_integer* (linhas 7 e 8). Na linha 10, é apresentada a definição da classe *Crianca*, que é um subclasse de *Humano*. Por ser uma subclasse, *Crianca* herda todos os atributos de *Humano*. Instâncias de classe são definidas através da palavra reservada *instance*. Exemplos de definição de instância são apresentadas nas linha 17 a 23. Nelas, são definidas a instância *Paulo* e *George*, ambas do tipo *Humano*.

Figura 1. Exemplo de ontologia em WSML

```
1  wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-rule"
2  namespace { _"http://br.usp.poli.pad/exemplosontologias#"}
3
4  ontology SeresVivos
5
6  concept Humano
7      nome ofType _string
8      idade ofType _integer
9
10  concept Crianca subConceptOf Humano
11
12  axiom EhCrianca
13     definedBy
14        ?x[idade hasValue ?idade] and ?idade < 12
15           implies ?x memberOf Crianca.
16
17 instance Paulo memberOf Humano
18     nome hasValue "Paulo Silva"
19     idade hasValue 34
20
21 instance George memberOf Humano
22     nome hasValue "George Carlos"
23     idade hasValue 8
```

O suporte à inferência é feita por meio de axiomas (linhas 12 a 15). O *axiom EhCrianca* define que caso uma instância *?x* apresente o atributo idade com valor menor que 12, logo *?x* é do tipo *Crianca*. Por conta desse axioma, podemos inferir que a instância *George* é do tipo *Crianca*, pois apresenta o atributo *idade* com valor 8.

A web semântica também traz o conceito de serviços semânticos. Diferente dos serviços habituais, serviços semânticos apresentam anotações semânticas que informam,

de maneira bastante precisa, sua funcionalidade. Adicionalmente, manipulam ontologias de forma a dar maior precisão ao processo de busca, recuperação, representação, extração, interpretação e manutenção da informação [Breitman et al. 2007].

Em WSML, um serviço semântico é definido através da palavra reservada *webService*. A funcionalidade do serviço é apresentada através de suas precondições, suposições, pós condições e efeitos. Para elucidar melhor esses conceitos, apresentamos a Figura 2 que define o serviço semântico *RegistroDeNascimento*.

De maneira análoga à ontologia, o cabeçalho de um serviço deve acompanhar a variante WSML e o seu *namespace* (linhas 1 e 2). É possível atribuir um rótulo a um *namespace*. Isso é feito na linha 3, onde é atribuído o rótulo *eo* para o *namespace* da ontologia *SeresVivos*. Na linha 4 é definido o *webService RegistroDeNascimento*. Em seguida, na linha 5, é realizada a importação da ontologia *SeresVivos*. Dessa forma, as classes e instâncias definidas na Ontologia *SeresVivos* tornam-se acessíveis ao *webService RegistroDeNascimento*. A descrição da funcionalidade do serviço é feita por meio da *capability rndCapability* (linha 7). Nesse campo, são apresentadas as precondições, suposições, pós condições e efeitos do serviço, definidas através das palavras reservadas *precondition*, *assumption*, *postcondition* e *effect*, respectivamente.

Precondition e *assumption* descrevem o estado antes da execução do *webService*. *Precondition* descreve condições sobre a entrada do serviço. No serviço *RegistroDeNascimento*, a *precondition* define que a instancia *?crianca* deve apresentar o atributo *localNascimento* com valor igual a *Brasil* ou a instância *?crianca* deve apresentar no atributo *pai* uma instância *?pai* cujo atributo *cidadania* possui valor igual a *Brasil* (linhas 9 a 11). Já *assumption* descreve condições sobre o estado do mundo não relacionadas aos dados de entrada. A *assumption* do serviço *RegistroDeNascimento* atribui temporalidade, sinalizando que o serviço deve observar a hora atual (*?hora*) e somente funcionar caso *?hora* represente horário comercial (linhas 13 a 15). *PostCondition* descreve a relação entre a

Figura 2. Exemplo de Serviço Semântico em WSML

```
1  wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-rule"
2  namespace { _"http://br.usp.poli.pad/exemplos#",
3              eo _"http://br.usp.poli.pad/exemplosontologias#"}
4
5  webService RegistroDeNascimento
6    importsOntology {eo#SeresVivos}
7    capability rdnCapability
8      sharedVariables ?crianca
9      precondition
10       definedBy
11         ?crianca memberOf Crianca and ?crianca[localNascimento hasValue Brasil]
12         or (?crianca[pai hasValue ?pai] and ?pai[cidadania hasValue Brasil] ).
13     assumption
14       definedBy
15         ?hora memberOf Tempo and ehHorarioComercial(?hora).
16     postcondition
17       definedBy
18         ?crianca memberOf Crianca and ?crianca[cidadania hasValue Brasil].
19     effect
20       definedBy
21         ?conta memberOf Conta and ?conta[debitos hasValue 100 ].
```

entrada e a saída. Na Figura 2, vemos que a instância *?crianca* passará a ter o atributo *cidadania* com valor igual a *Brasil* (linhas 16 a 18) caso a *precondition* e *assumption* sejam atendidas. Por último, *effect* descreve mudanças causadas pelo serviço não relacionada

aos dados de entrada e saída. Em nosso exemplo, é gerado uma instância de *Conta* cujo atributo *debitos* apresenta valor igual a 100 (linhas 19 a 21).

WSML também dá suporte à modelagem de metas através da palavra reservada *goal*. *Goal* representa os objetivos dos usuários com relação ao serviço web utilizado. A *goal* é composta pelos mesmos elementos que descrevem um *webService*, mudando apenas pelo emprego da palavra reservada *goal* no lugar da palavra reservada *webService*. A diferença entre esses elementos consiste do ponto de vista da descrição. Enquanto um *webService* adota o ponto de vista do serviço, *goal* segue o ponto de vista do solicitante do serviço.

3. Trabalhos relacionados

As ferramentas de navegação, definição de rotas ou *wayfinding* são de extrema importância para as PcD pois tais ferramentas podem aumentar o grau de independência e de autonomia destas pessoas. Como todo cidadão, é navegando pelo espaço urbano que as PcD percebem e se apropriam do espaço que é compartilhado com os seu concidadãos e se sentem parte integrante da comunidade que está ao seu redor. As obras da literatura científica relacionadas a *wayfinding* podem ser agrupadas segundo alguns critérios, tais como:

- roteamento *indoor* (Chang et al., 2010) ou *outdoor* [Sheehan et al. 2006];
- deficiência/limitação em questão: visual [Gaunet and Briffault 2005], cognitiva [Chang et al. 2008], auditiva [de Paolis and Guerini 2015], velhice [Falco et al. 2010], motora [Duvall et al. 2016], etc.;
- tecnologia utilizada, tais como: RFID [Willis and Helal 2006], *wearables* [Ross and Blasch 2000], robôs [Kulyukin et al. 2006], entre outras.

Poucos são os trabalhos que propõem o uso de semântica em soluções de *wayfinding*. Nossa proposta é uma solução baseada em semântica voltada para navegação e roteamento *indoor* e *outdoor* em espaços públicos para PcD com limitações sensoriais e motoras, podendo seu uso ser extrapolado também para outros indivíduos que por alguma razão temporária ou permanente não conseguem interpretar os símbolos e textos presentes na paisagem urbana — como, por exemplo, idosos, indivíduos não-letrados, e turistas que não falam/leem/entendem o idioma local. As tecnologias empregadas são aquelas utilizadas comumente em projetos de Cidades Inteligentes: sensores, Internet das Coisas, Computação em Nuvem, e dispositivos móveis.

4. Solução desenvolvida

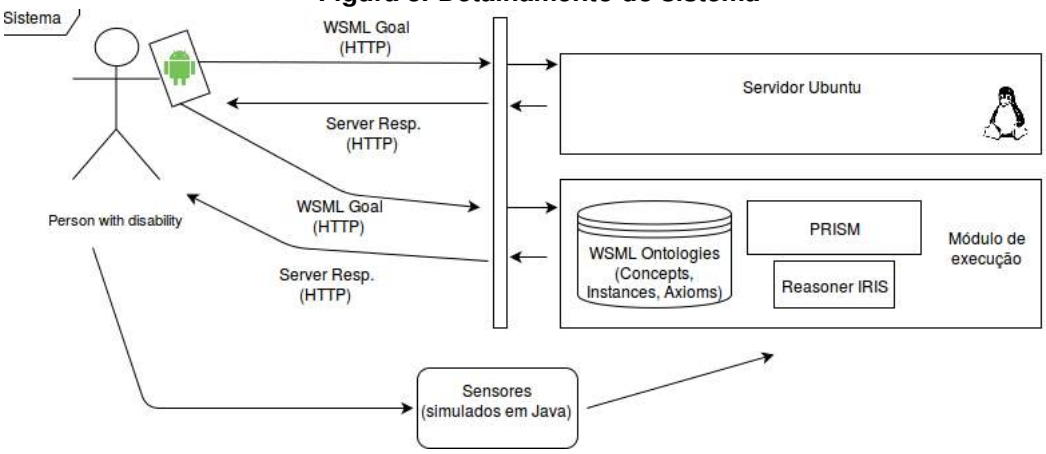
Para validar o modelo proposto e confirmar a sua viabilidade desenvolvemos um protótipo tendo como cenário um sistema de auxílio a pessoas com deficiência visual na navegação em um shopping center aplicando a nossa abordagem semântica. O protótipo descreve os vários serviços oferecidos por um shopping, bem como os estabelecimentos encontrados no centro comercial. A Figura 3 detalha a solução desenvolvida[2]. A interação com o usuário é feita por meio de um aplicativo Android que se comunica com um servidor RESTful, utilizando requisições HTTP que encapsulam pacotes JSON (ilustrados pelas setas). Para utilizar um serviço, a aplicação envia um *goal* WSML para o servidor Ubuntu

[2]Documentos em WSML, bem como diagramas detalhando o sistema, estão disponíveis em: `https://github.com/minato21/shopping_ontology`

que descobre um serviço que atende às necessidades impostas na meta (*goal*). Como resposta, o servidor envia à aplicação Android a URL do serviço. Em seguida, a aplicação faz uma chamada à URL para que o serviço seja executado no módulo de execução. O módulo de execução apresenta um conjunto de ontologias WSML. O processo de consulta é feito através do IRIS [Bishop and Fischer 2008].

O cálculo de rotas é feito dinamicamente por uma ferramenta de checagem de modelos probabilísticos chamada PRISM [Kwiatkowska et al. 2011]. Ele se baseia na localização do usuário e nas informações presentes nas ontologias de lugares e rotas. A localização do usuário é obtida através de sensores que notificam o servidor sobre a localização atual do usuário assim que o usuário passa por ele. Em nosso protótipo, sensores foram desenvolvidos como serviços web, de forma a dar suporte a diferentes tipos de hardware. Em nossos experimentos, adotamos a modalidade simulada, na qual a ação de detecção e notificação dos servidores é feita puramente em software.

Figura 3. Detalhamento do sistema



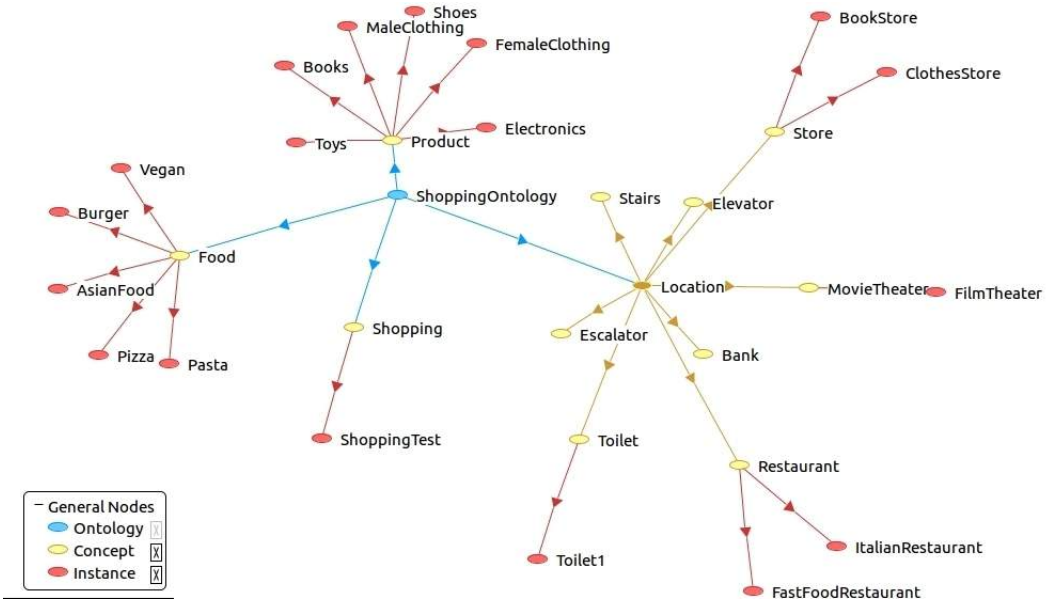
Para o protótipo, definimos que o sistema deve atender às seguintes funcionalidades: buscar localidade, procurando mais informações sobre a localidade desejada; buscar rota, tal que ela seja calculada pelo servidor entre o ponto atual do usuário e uma localidade selecionada; curtir localidade, explicitando sua preferência pela localidade selecionada; e recomendar localidade, em que, caso o usuário esteja próximo a uma localidade curtida, o sistema o avisa.

4.1. Descrição do shopping em ontologias

A Figura 4 mostra uma visão gráfica da ontologia que representa o shopping utilizado em nosso estudo de caso, produzida pela ferramenta Web Semantic Modeling Toolkit. Para nossa ontologia de shopping, definimos que cada localidade tem um sensor correspondente. A cadeia de sensores foi modelada em uma ontologia de grafo, sendo as localidades representadas por nós e o caminho entre as localidades, por arestas.

O shopping apresenta seis instâncias de localidade. São elas: **Toilet1**, um banheiro; **ItalianRestaurant**, um restaurante que oferece pizzas, massas italianas e também comida vegetariana; **FilmTheater**, um cinema; **FastFoodRestaurant**, uma hamburgueria; **ClothesStore**, uma loja de roupas; e **BookStore**, uma livraria.

Figura 4. Representação gráfica da ontologia *ShoppingOntology*



A semântica relativa à funcionalidade dos serviços foi modelada na ontologia *ShoppingOntology*, que define a semântica dos dados presentes nas pré e pós condições dos *webServices* e *goals* WSML. No caso de não existir uma instância que contemple as precondições de uma *goal*, gera-se dinamicamente uma instância. Essa instância é útil para que o serviço de busca consiga realizar o casamento entre a *goal* e o *webService*.

4.2. Aplicativo Android

Para que o usuário consiga utilizar os serviços disponíveis no servidor, desenvolvemos um aplicativo Android. Pensando nas limitações das pessoas com deficiência visual, este aplicativo permite que usuário interaja com o sistema a partir de comandos de voz. Uma vez que esse comando, relacionado às funcionalidades descritas anteriormente, seja compreendido, o sistema cria uma *goal* descrita em WSML e a envia para o servidor de descoberta de serviços. Caso o servidor descubra algum serviço que contemple a especificação da *goal*, ele envia a URL do serviço. O aplicativo Android irá, então, reenviar a *goal* para esse endereço. A resposta desse serviço é interpretada e informada ao usuário utilizando um sintetizador de voz.

O aplicativo também é responsável por comunicar ao usuário as informações recebidas do servidor. Tais informações podem ser referentes à localização do usuário, à recomendação de localidades, caso ele se encontre próximo de localidades definidas como curtidas, às informações sobre localidades relevantes a uma busca, ou, no caso de uma rota, podem fornecer instruções que o usuário deve seguir para chegar ao destino desejado. Todas essas informações são notificadas ao usuário utilizando o sintetizador de voz.

4.3. Serviço de descoberta

O algoritmo de busca baseia-se no casamento de funcionalidade entre a meta do usuário (*goal*) e a descrição do serviço semântico (*webService*). Como apresentado na Seção 2, em WSML, *goal* e *webService* são descritas por meio de suas *precondition*, *assumption*,

postcondition e *effect*. Em nossa implementação, usamos a abordagem clássica que adota o conceito de *subsumption* [Paolucci *et al*, 2002]. Nessa abordagem, o casamento entre serviços e metas é realizado através da observação da compatibilidade semântica de suas respectivas definições.

Para o *ranking*, bem como para definição do grau de compatibilidade do serviço, adotamos um sistema de 4 bits, conforme ilustrado na Tabela 1. O bit mais à esquerda refere-se a *precondition* e o mais a direita ao *effect*. O valor 1 é atribuído aos bits cuja descrição da meta e do serviço são equivalentes. Dessa forma, serviços que apresentam compatibilidade total com a meta apresentam maiores valores. O usuário também pode estabelecer um grau de compatibilidade mínimo. Para isso, o usuário deve passar para o serviço de busca um valor no intervalo de [1, 15]. Tal valor sinaliza quais descrições de funcionalidade devem ser obrigatoriamente compatíveis com a meta.

Tabela 1. Mapeamento entre bit e o elemento de descrição do webService

preconditon	*assumption*	*postcondition*	*effect*
bit 3	bit 2	bit 1	bit 0

4.4. Módulo de execução

Este módulo disponibiliza serviços que podem ser descobertos. Cada serviço é isolado um dos outros, uma vez que, em um cenário real, é possível que os serviços disponíveis estejam hospedados em servidores diferentes. O módulo atual abrange três funcionalidades básicas — buscar localidade, curtir localidade e buscar rota — e, mais dois outros serviços de apoio: serviço de autenticação e serviço de sensoriamento.

- **Serviço de autenticação:** O serviço de autenticação é necessário para guardar informações sobre o usuário, no sistema atual, o endereço de IP do aparelho móvel e a porta de escuta do aplicativo. Estas informações são utilizadas para notificar o aplicativo sobre sua posição atual.
- **Serviço de sensor:** Este é o único serviço cuja requisição não parte do aplicativo. O sensor envia para o servidor o identificador do sensor (isto é, o identificador do nó pelo qual o usuário acabou de passar), e, utilizando as informações de IP e porta recebidas do aplicativo pelo serviço de Login, o serviço de sensor repassa essa informação do nó onde o usuário está localizado para o aplicativo móvel.
- **Serviço de busca:** O serviço de busca opera da seguinte forma: primeiramente, o serviço carrega as ontologias relativas ao sistema (i.e., o shopping modelado em WSML). Em seguida, ele analisa a goal recebida com uma instância que contém atributos que serão utilizados para encontrar a localidade desejada. Depois, ele verifica se a instância contida na *goal* WML é equivalente à definida na pré-condição do WSML WebService. Se for, com as informações da instância contida na *goal* e utilizando o *reasoner* IRIS, uma busca é feita na ontologia de shopping sobre as localidades relevantes. Finalmente, a resposta dessa busca é tratada e retornada ao aplicativo. As informações contidas no pacote JSON de resposta para cada localidade encontrada são: o nome da localidade; a descrição; o nó; e se o local foi curtido.
- **Serviço de localidades curtidas:** Este serviço é semelhante ao serviço de busca, com algumas particularidades: o serviço não só busca a ontologia que deve ser

curtida ou não utilizando o reasoner IRIS, mas também salva a ontologia atualizada no seu arquivo de origem. Por fim, o serviço responde ao aplicativo as informações atualizadas da localidade alterada.

- **Serviço de rota:** Este serviço recebe, do pacote JSON vindo do aplicativo Android, uma instância com informação sobre dois nós, um inicial e outro final. Similar aos serviços de busca e de localidades curtidas, ele deve validar a ontologia contendo a instância que recebeu, juntamente com a *goal* WSML. Com as duas informações, o serviço utiliza o software PRISM para o cálculo da rota. Para isso, modelamos o layout do nosso shopping em um processo de decisão de Markov, com as localidades representadas por estados e transições representando os caminhos entre as localidades, com os custos relacionados às distâncias desses caminhos. Quando o cálculo é finalizado, o serviço envia para o aplicativo um conjunto de pares de String, cada par contendo a descrição da rota que o usuário deficiente deve seguir, e o próximo nó por onde ele deverá passar caso siga as instruções corretamente, sendo que o último par contém o nó final. O serviço apresenta suporte a definição de novas rotas, caso o usuário não siga a rota estabelecida.

4.5. Sensores de localidade

Os sensores de localidade correspondem a uma classe de serviços que enviam ao módulo de execução a informação do nó, isto é, seu identificador. São ativos no momento em que o deficiente passa por ele.

5. Avaliação realizada

Para avaliação, verificamos o uso do sistema utilizando o aplicativo Android para interagir com o serviço de descoberta e o módulo de execução no servidor Ubuntu. Simulamos o comportamento de um usuário cego, bem como a atuação dos sensores de localidade[3].

O servidor de descoberta funcionou como esperado, retornando corretamente os serviços relativos às goals enviadas. O módulo de execução dos serviços também funcionou sem problemas. Para o serviço de rotas, fizemos uma simulação de 100 requisições HTTP para um shopping com seis localidades. A média de tempo de execução entre o envio da requisição e da resposta do servidor foi de 1875 ms em um servidor Apache Tomcat 8.5 executando em um Linux Ubuntu, em uma máquina Intel i7 com 8 GB de memória.

Finalizado o uso, concluímos que, com o protótipo desenvolvido, a produção de um sistema aplicado a uma situação real, seguindo a metodologia adotada neste trabalho, é viável.

6. Considerações finais

Esse trabalho teve como objetivo apresentar uma abordagem semântica para o desenvolvimento de sistemas voltados a pessoas com deficiência visual. Atualmente, a abordagem proposta vem sendo aplicada no desenvolvimento de ambientes inclusivos para a Escola de Artes, Ciências e Humanidades da Universidade de São Paulo. O sistema em desenvolvimento adota todos os recursos apresentados nessa proposta, com a adição de recursos

[3]Um vídeo de demonstração do sistema está disponível em: `https://youtu.be/Z5p-3x3fMd8`

mais robustos para interface com o usuário e sensoriamento. Para interface do usuário, o sistema apresenta uma luva, responsável por abrigar o dispositivo móvel, bem como enviar *feedbacks* ao usuário. No âmbito de sensoriamento, optou-se pela adoção de *QR Code* espalhados pelo campus, que são lidos e processados pelo aplicativo. Como trabalhos futuros, pretende-se implantar o suporte multi-idiomas. Adicionalmente, tornar mais robusta as ontologias relativas ao contexto, de forma a prover um serviço mais adaptável ao perfil do usuário.

Referências

Alawadhi, S. et al. (2006). Building understanding of smart city initiatives. *International Conference on Electronic Government.*, pages 40–53.

Bishop, B. and Fischer, F. (2008). Iris-integrated rule inference system. In *International Workshop on Advancing Reasoning on the Web: Scalability and Commonsense (ARea 2008)*.

Breitman, K., Casanova, M. A., and Truszkowski, W. (2007). *Semantic web: concepts, technologies and applications*. Springer Science & Business Media.

Chang, Y.-J., Tsai, S.-K., and Wang, T.-Y. (2008). A context aware handheld wayfinding system for individuals with cognitive impairments. *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility.*, pages 27–34.

de Oliveira Neto, J. S. and Kofuji, S. T. (2016). Inclusive smart city: An exploratory study. *International Conference on Universal Access in Human-Computer Interaction.*

de Paolis, R. and Guerini, S. (2015). Wayfinding design and accessibility. *GSTF Journal of Engineering Technology*, 3(2).

Duvall, J. A., Pearlman, J. L., and Karimi, H. A. (2016). Development of route accessibility index to support wayfinding for people with disabilities. *Smart City 360.*, pages 104–112.

Falco, J. M. et al. (2010). Indoor navigation multi-agent system for the elderly and people with disabilities. *Trends in practical applications of agents and multiagent systems.*, pages 437–442.

Federici, S. and Scherer, M. (2012). *Assistive Technology Assessment Handbook.* CRC Press.

Fensel, D., Facca, F. M., Simperl, E., and Toma, I. (2011). *Semantic web services*. Springer Science & Business Media.

Gaunet, F. and Briffault, X. (2005). Exploring the functional specifications of a localized wayfinding verbal aid for blind pedestrians: Simple and structured urban areas. *Proceedings of the ACM 2011 conference on Computer supported cooperative work.*, 20(3):267–314.

Kulyukin, V. et al. (2006). Robot-assisted wayfinding for the visually impaired in structured indoor environments. *Autonomous Robots*, 21(1):29–41.

Kwiatkowska, M., Norman, G., and Parker, D. (2011). Prism 4.0: Verification of probabilistic real-time systems. In *International Conference on Computer Aided Verification*, pages 585–591. Springer.

Nam, T. and Pardo, T. A. (2011). Conceptualizing smart city with dimensions of technology, people, and institutions. *Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times.*, pages 282–291.

Ross, D. A. and Blasch, B. B. (2000). Wearable interfaces for orientation and wayfinding. *Proceedings of the fourth international ACM conference on Assistive technologies.*, pages 193–200.

Sheehan, B., Burton, E., and Mitchell, L. (2006). Outdoor wayfinding in dementia. *Dementia*, 5(2):271–281.

Willis, S. and Helal, S. (2006). Rfid information grid for blind navigation and wayfinding. In *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, pages 34–37. IEEE.

World Bank (2016). Disability overview. `http://www.worldbank.org/en/topic/disability/overview`. Acesso em 05 de março de 2017.

World Health Organization et al. (2011). *World report on disability.* World Health Organization.