

Modelagem e Desenvolvimento de Jogos Móveis Baseados em Localização

Cristiane Ferreira
Universidade Federal do
Ceará
Fortaleza - CE - Brasil
cristianeferreira@great.ufc.br

Windson Viana
Universidade Federal do
Ceará
Fortaleza - CE - Brasil
windson@great.ufc.br

Fernando Trinta
Universidade Federal do
Ceará
Fortaleza - CE - Brasil
fernandotrinta@virtual.ufc.br

ABSTRACT

A Pervasive Game is based on a combination of hybrid interfaces involving mobile devices, wireless networks, positioning systems and context sensing technologies. Location-Based Mobile Games (LBMGs) are a subclass of pervasive games. They make use of location technologies and add the position of their players in the game rules. A state of the art survey involving locative mobile games, LBMGs authoring tools, and modeling pervasive games was conducted. This research aims at presenting a language that allows the representation of LBMGs and the media used in the mechanics of these games. We want to support temporal aspects, as well as, spatial relationships. Also, the research will investigate checking mechanisms for generating LBMGs to be incorporated into LAGARTO tool.

Keywords

Location-based games; authoring tool; game modeling

1. INTRODUÇÃO

Antes dos jogos de computador aparecerem, os jogos eram projetados e jogados no mundo físico, contando com propriedades do mundo real, como objetos físicos, espaço físico, etc [9]. Os jogos evoluíram bastante ao longo das últimas duas décadas, influenciados pelo avanço tecnológico alcançado pelos componentes de hardware e software. Os jogos de computador e de consoles se tornaram uma forma de entretenimento dominante, provendo um alto nível de atratividade, com a criação de um mundo virtual imaginário e interativo, a utilização de gráficos, de sons, permitindo que o jogador fique imerso na ilusão do jogo [6].

O desenvolvimento da tecnologia móvel ampliou as plataformas digitais possíveis para execução dos jogos. Suas características peculiares (e.g., sensores embutidos, capacidade de comunicação e onipresença) permitem a viabilização dos jogos pervasivos digitais. Esse conceito emergiu durante a década passada [6] e é bastante amplo, principalmente, porque cada pesquisador tem uma interpretação própria sobre do que se trata a computação pervasiva e a computação ubíqua. Além disso, apresentam visões distintas sobre

o termo "jogos pervasivos"[12]. Jogo pervasivo se baseia em uma combinação de espaços híbridos que envolve o dispositivo móvel, redes sem fio, sistemas de posicionamento e tecnologias de sensoriamento. Os Jogos Móveis Baseados em Localização (JMBLs) são uma subclasse dos jogos pervasivos. São aqueles que fazem uso de tecnologias de localização e que agregam a posição de seus jogadores nas regras do jogo [11]. Nesses jogos, as interações ocorrem à medida que os jogadores se deslocam no mundo real. O Pokemon Go é o exemplo mais recente de sucesso dessa categoria de jogos.

O desenvolvimento de jogos digitais, por si só, já não é um processo trivial. Esse processo se torna mais complexo ao introduzir a necessidade de posicionamento físico dos jogadores nas regras do jogo [6]. Uma alternativa para isso é a utilização de ferramentas de autoria, que são softwares que oferecem um conjunto de recursos para a criação de conteúdos (e.g., texto, imagem, código fonte, movimentos, atividades lúdicas, jogos, etc). Algumas ferramentas de autoria propostas na literatura auxiliam a criação de jogos móveis baseados em localização, garantindo menores custos e descartando a necessidade de conhecimentos específicos de programação de jogos [14]. Porém, algumas dessas ferramentas apresentam limitações, como, em relação ao tipo de jogo que geram, na concepção de regras e mecânicas, também na ordenação de tarefas que devem ser realizadas pelo jogador. Outro ponto notório é a ausência de um modelo claro para a representação dos jogos.

O processo de criação e desenvolvimento de um jogo é uma atividade multidisciplinar que envolve profissionais de vários campos, artistas, *designer* de animações, de interface, de sons, desenvolvedores, etc. A necessidade de uma boa comunicação, no que diz respeito ao conceito do jogo, em uma equipe desse tipo é um ponto pertinente para um projeto com resultado positivo. Algumas abordagens de modelagem de jogos digitais utilizam UML (*Unified Modelling Language*) para esta tarefa [2] [17] [18] [15]. Outras pesquisas vão mais além e propõem DSLs (*Domain-Specific Language*) próprias [16] para a representação dos jogos. Apesar destes exemplos, alguns deles aplicados a jogos pervasivos, as ferramentas de autoria de JMBLs utilizam uma notação própria para representação do funcionamento do jogo. Não há uma uniformidade entre elas tanto na definição de modelo de JMBL como em sua representação. Em geral, as ferramentas mostram o jogo sob várias perspectivas, como, a montagem da cena de jogo, a especificação das atividades que o jogador deve realizar, o fluxo das cenas, o fluxo de atividades, utilizam gráficos para exibir o tempo e o ritmo especificados para o jogo.

In: Workshop de Teses e Dissertações (WTD), 16., 2016, Teresina. Anais do XXII Simpósio Brasileiro de Sistemas Multimídia e Web. Porto Alegre: Sociedade Brasileira de Computação, 2016. v. 2.

ISBN: 978-85-7669-332-1

©SBC – Sociedade Brasileira de Computação

A ausência de um modelo de representação claro de um JMBL e das mídias utilizadas no jogo pode prejudicar o entendimento das mecânicas por parte dos profissionais envolvidos desde os usuários das ferramentas até os seus desenvolvedores. Além disso, essa ausência pode dificultar a evolução e o suporte das ferramentas de autoria para outras plataformas. Esta pesquisa visa, portanto, conceber uma linguagem declarativa para a descrição de jogos baseados em localização inspirada em linguagens multimídia (e.g., NCL, SML) de forma a representar aspectos temporais e espaciais destes jogos. A linguagem deve permitir a representação de forma intuitiva e clara das mecânicas e das regras de JMBLs, incluindo as mídias utilizadas nas mecânicas desses jogos, com suporte para aspectos temporais, assim como para relações espaciais. A linguagem será acoplada a uma ferramenta Web de autoria de JMBLs desenvolvida na UFC: a LAGARTO [3].

2. FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

2.1 Modelagem e Representação de Jogos Digitais

A modelagem é feita com o objetivo de unificar as visões de *designers*, desenvolvedores e jogadores sobre um mesmo jogo.

A linguagem UML tem sido utilizada em vários domínios, incluindo o domínio de jogos computacionais. UML tem sido aplicada no campo de *design* de jogo, utilizando diagramas como casos de uso e classes. [2] argumenta a utilização de diagramas UML para documentar o *design* do software do jogo. [17] também utilizaram diagramas de casos de uso para representar a interação do jogador com o jogo. [18] utilizaram diagramas de componentes para representar as dependências entre os arquivos do código, diagramas de atividades para o fluxo de execução do software e diagramas de sequência para a interação entre os objetos do software. Já os autores de [15] usaram diagramas de classes e diagramas de transição de estado, para representar a estrutura do jogo e seu comportamento, e diagramas de objetos e diagramas de sequência, para analisar o comportamento dos objetos do jogo em tempo de execução.

Uma DSL¹ Uma (*Domain-Specific Language*) para especificação de jogos digitais foi proposta por [16]. Ela considera o contexto social, a estrutura e um conjunto de regras de videogames. A proposta consiste em uma especificação para *designers* de jogos poderem expressar a jogabilidade de maneira clara e precisa. Essa abordagem é multi dimensional e representa diferentes visões dos pontos de interesse do jogo. Os autores esperavam que a DSL suportasse aspectos como interface gráfica, *design* de áudio, IA (Inteligência Artificial), narrativa do jogo, *design* de fases, etc. Assim, o trabalho seria um primeiro passo na metodologia de desenvolvimento de jogos dirigido por modelos. Foram propostos três metamodelos: O diagrama de contexto social, que define o número de jogadores e de equipes do jogo; o diagrama de estrutura, que define os elementos, os atributos e os eventos do jogo; e o conjunto de regras, que define o comportamento do jogo,

¹DSLs são linguagens adaptadas a um domínio de aplicação específico. Elas oferecem ganhos substanciais em expressividade e facilidade de uso em comparação com linguagens de programação de uso geral em seu domínio de aplicação [10].

especificando o *gameplay* através de regras declarativas precisamente definidas. Dessa forma, todas as características do *gameplay* estão incluídas na perspectiva correspondente.

2.2 Jogos Móveis Baseados em Localização

Os Jogos Móveis Baseados em Localização (JMBLs, ou do inglês, *Location-Based Games, LBGs*) são uma subclasse dos jogos pervasivos. São aqueles que fazem uso de tecnologias de localização e que agregam a posição de seus jogadores nas regras do jogo [11]. Essas regras induzem a necessidade de movimentação a certas localizações no ambiente do mundo real, que pode ser uma posição absoluta, isto é, coordenadas GPS, ou uma posição relativa à posição atual ou a outro jogador [7]. Esses jogos usam tecnologias *location-aware* como parte de sua interface [5] e dependem de tecnologias digitais portáteis, como telefones celulares, receptores GPS, PDAs, *notebooks*, assim como de uma série de redes e conexões, como *Wi-Fi*, *Wi-Max*, *bluetooth*, RFID, GPRS, 3G, 4G, para fazer a interface entre o espaço físico e digital [4] [11]. Eles promovem uma "percepção dupla" de espaço, permitindo acesso simultâneo a ambos os espaços físico e digital. Nesses jogos, os jogadores portam dispositivos móveis equipados com sensores de localização e se movem em ambientes reais, visitando locais de interesse do jogo. Dessa forma, as interações ocorrem à medida que os jogadores se deslocam no mundo real.

2.2.1 Classificação de JMBLs

Segundo [13], considerando o nível de dependência de localização, jogos com as características citadas podem ser divididos em três categorias: jogos móveis, jogos sensíveis à localização e jogos sensíveis espacialmente.

Nos jogos móveis, os eventos acontecem quando os jogadores estão próximos localmente, co-localizados. Nesse caso, não se faz necessário o uso de GPS, apenas o sensor de proximidade e a comunicação local são suficientes. Os jogos são ditos sensíveis à localização quando são influenciados de alguma forma pela localização do jogador. Nessa categoria, eventos podem ser disparados quando um usuário visita determinados lugares. A terceira categoria, os jogos sensíveis espacialmente, integra mais fortemente o jogo e o ambiente do mundo real. Os eventos ocorrem quando o jogador estiver em um determinado contexto espacial, por exemplo, quando o jogador entra em uma igreja.

2.2.2 Padrões de Jogo de JMBLs

Para usar localização em jogos, vários padrões de jogos ou combinação deles podem ser utilizados. Os JMBLs apresentam alguns padrões: Procurar e Encontrar (*Search-and-Find*), Seguir o Caminho (*Follow-the-Path*), Perseguir e Capturar (*Chase-and-Catch*) e Variar a Distância (*Change-of-Distance*) [8]. A Figura 1 representa de maneira resumida esses padrões.

Procurar e Encontrar é um padrão popular no qual o jogador deve encontrar uma certa geolocalização. Esse local é sempre um ponto fixo no jogo. Apenas uma localização aproximada pode ser dada ao jogador, assim ele deve pesquisar sobre o local de destino. O jogador pode também ser guiado pelo jogo utilizando algum sistema de navegação. O padrão Seguir o Caminho é similar ao anterior. Nesse, o jogador também deve chegar a um destino, entretanto, o foco não está no destino, mas no modo com que o jogador o encontra. Para encontrá-lo, o jogador deve percorrer uma rota

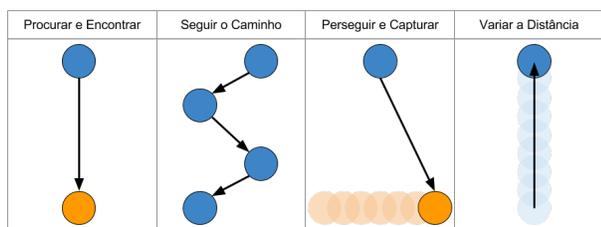


Figura 1: Resumo gráfico de alguns padrões de JMBL. Fonte: Adaptado de [8].

predefinida e será penalizado se não o fizer. Nessa rota, o jogador deve encontrar e visitar vários pontos consecutivos.

O padrão Perseguir e Capturar também é popular para JMBLs. Em jogos com esse padrão, o jogador deve caçar um objeto em movimento no mundo do jogo. Esse objeto pode ser outro jogador ou um objeto apenas virtual e o jogo pode ser *single* ou *multiplayer*. Nesse padrão o destino está frequentemente mudando. O padrão Variar a Distância é menos utilizado. Nesse padrão, o destino e a direção não importam tanto, mas sim a movimentação. O objetivo do jogo é se aproximar ou se afastar da localização dada.

2.3 LAGARTO

A LAGARTO (A LocAction based Games AuthoRing Tool) é uma ferramenta de autoria para construção de JMBLs, capaz de lidar com realidade aumentada, apresentando imagens, vídeos, sons e objetos 3D. Em se tratando do componente editor de jogos, é uma aplicação web que permite que os projetistas possam definir o local do jogo, suas missões e mecânicas, possíveis relações entre as missões e seu fluxo. O editor define ainda possíveis grupos de jogadores e as missões vinculadas a tais grupos. A ferramenta possui uma série de janelas e utiliza uma notação visual que permite ao usuário definir como missões e mecânicas são compostas.

Como características positivas da ferramenta tem-se: Geração de jogos sem programação; Suporte a dependências entre as missões; notação visual para definição das missões e mecânicas; permite definição de grupos de jogadores e as missões vinculadas a tais grupos; possui suporte para utilização de realidade aumentada.

Como pontos negativos da ferramenta podem ser destacados: Não possui aspectos temporais; suporta apenas jogos online; não possui algoritmos de checagem das especificações dos jogos; não possui suporte a relações espaciais.

3. CONTRIBUIÇÕES ESPERADAS

O foco desta pesquisa é a implementação de uma linguagem que permita representar as regras e as mecânicas de JMBLs de forma clara e descritiva. A modelagem incluirá também especificações sobre as mídias utilizadas nas mecânicas desses jogos. A linguagem proposta para a modelagem é denominada LEGaL (Location based mobile Games Language) e será baseada na estrutura e em alguns elementos da linguagem multimídia NCL (*Nested Context Language*) [1]. Em geral, a NCL tem foco em aplicações para TV Digital e permite a organização tanto espacial quanto temporal dos elementos hipermídia. A LEGaL se baseia no sincronismo temporal com mídias utilizado pela NCL e na estrutura usada pela linguagem para relacionar elementos. A linguagem proposta permite a construção de documentos que

trazem definições de como as missões do jogo são compostas e relacionadas entre si.

A proposta visa também à integração da LEGaL à ferramenta já existente LAGARTO (A Location based Games AuthoRing Tool), implementada por pesquisadores do GREat, da UFC, para a construção dos JMBLs. Para isso, serão implementados um gerador, um validador e um interpretador da linguagem, que serão integrados à ferramenta.

4. ESTADO ATUAL DO TRABALHO

4.1 Idealização da Linguagem

A linguagem é do tipo declarativa, baseada na estrutura e nos conceitos sobre mídia da linguagem NCL. A disposição dos arquivos se assemelha à estrutura da NCL e a linguagem se utiliza de alguns de seus *tokens*. O documento da linguagem proposta pode ser traduzido em um grafo aninhado direcionado, utilizando o conceito de nós e arestas para descrever os JMBLs. O grafo representa as missões a serem realizadas no jogo, os relacionamentos entre as missões e exibe o fluxo do jogo, desde o seu início até a sua finalização.

Para composição desse grafo que modela o jogo, poderão ser utilizados dois tipos de nós: nós de mídia, ou de conteúdo, que especificam a mídia a ser utilizada, e nós de missão, ou de contexto ou ainda de composição, que caracterizam o aninhamento de nós e representam as missões a serem realizadas no jogo.

As arestas do grafo representam as relações entre os nós, caracterizando o fluxo do jogo. Na linguagem proposta, a relação de sincronismo de tempo entre os nós será definida pelos conectores e elos. Os conectores definirão a relação entre um ou mais nós de origem, os nós que irão disparar uma ação, e um ou mais nós de destino, os nós que serão afetados pela ação. Quando a ação for disparada, o elo será ativado. Em geral, o elo associa os nós através dos conectores, estabelecem o sincronismo de tempo entre os nós. Ele associa os papéis, definidos pelos conectores, às mídias. Na definição do elo, são indicados quais nós farão os papéis determinados pelo conector, ou seja, quais nós serão de origem e de destino. O comportamento dos elos será definido pelos conectores.

Outros dois elementos poderão ser utilizados nos relacionamentos entre os nós: as portas e as âncoras. As portas servirão para definir onde o jogo terá início e para indicar qual o próximo nó interno a ser executado, seja nó de mídia ou de missão. As âncoras deverão indicar uma parte da mídia para ser ativadora de um elo, ou seja, um segmento de mídia de um nó para ser a origem ou destino de elo.

Para definir como as mídias serão exibidas no jogo serão utilizados os descritores, que definem parâmetros como volume da mídia, duração, transparência, visibilidade, dentre outros. Os descritores também associam a mídia a uma localização já definida. Uma mídia pode ser associada a mais de um descritor, isso permite definir diferentes apresentações para a mesma mídia.

Os localizadores tem a função de definir as relações espaciais do jogo. Eles especificam os detalhes das geolocalizações onde se encontrarão as missões do jogo e a restrição espacial da distância mínima necessária até a localidade para que o jogador possa realizar a missão.

Resumidamente, como elementos da linguagem proposta, para modelagem dos JMBLs, tem-se os nós de contexto re-

presentando as missões do jogo, os nós de mídia que são as mídias a serem visualizadas na cena de jogo, os conectores representando as ligações e relacionamentos entre os nós, especificando a sincronia de tempo, com causa e consequência, e definindo como os elos são ativados e o que eles disparam, os descritores, que definem como as mídias são apresentadas, e os localizadores, que especificam as geolocalizações utilizadas no jogo.

4.2 Tokens da Linguagem

O documento da LEGaL é escrito em XML e precisa ter um cabeçalho de definição do documento, utilizando a tag `<jmbl>`, um cabeçalho do programa, `<head>`, um corpo do programa, `<body>`, e o encerramento do documento, definido por `</jmbl>`. As definições das bases de localizações, de descritores e de conectores são feitas no cabeçalho documento. Os nós, as mídias e os elos são definidos no corpo do documento.

Alguns *tokens* foram baseados nos *tokens* da linguagem NCL, dessa forma, os conectores são definidos com a tag `<connector>`, os descritores definidos por `<descriptor>`, os nós de mídia são descritos através da tag `<media>` e podem ser especificados utilizando parâmetros como `id`, `src`, `type`, `descriptor`, `locator`. Os nós de missão são descritos por `<mission>` e são identificados no código por seu `id`. Os elos são definidos pelo elemento `<link>` e é descrito através dos parâmetros `id`, que identifica o elo no código, e `xconnector`, que indica o conector utilizado.

Um elemento localizador representado pela tag `<locator>` especifica as geolocalizações de cada mecânica do jogo. Ele é definido no cabeçalho do documento e é descrito por parâmetros como `id`, que indica o identificador ao qual será referenciada a geolocalização, `lat`, que indica a latitude, e `lon`, que indica a longitude referente àquela geolocalização.

São definidos, pelos conectores, papéis de relacionamento (condição e ação) entre nós de mídia e entre nós de missão. Os papéis de condição para ambos os tipos de nós poderão ser: `onBegin`, `onEnd`, `onPause`, `onResume`, `onAbort`. Os papéis de ação, aplicados a nós de mídia, poderão ser: `start`, `stop`, `pause`, `resume`, `abort`, `set`. Os papéis de ação para nós de missão poderão ser: `enable`, `desable`. O valor `enable` utilizado para desbloqueio de um nó de missão, quando esta possui pré-requisitos a serem cumpridos, e o `desable` pode ser utilizado para um caso em que a missão pode ser realizada mais de uma vez. A Tabela 1 traz um breve resumo sobre os principais elementos da linguagem proposta.

Os tipos de mídias suportados pela linguagem são texto, imagem, áudio, vídeo e objeto 3D e obedecem à norma MIME (*Multipurpose Internet Mail Extensions*) que é norma utilizada na Internet para o formato das mensagens de correio eletrônico. Os tipos de conteúdo são especificados no nó de mídia pelo atributo `type` que indica o `type` e o `subtype` do conteúdo da mídia no seguinte formato: `type = "<type> / <subtype>"`. A Tabela 2 mostra os tipos de mídias suportados pela linguagem proposta.

Primeiramente devem ser especificadas as localidades, que são as geolocalizações onde se encontrarão as mídias que compõem as missões. Em seguida, devem ser definidos os descritores, com detalhes de como a mídia será apresentada. Após isso, conectores e elos são definidos e, em seguida, as mídias devem ser especificadas. O passo seguinte é a definição das portas para a composição do fluxo de missões. A

Listagem 1 exemplifica a estrutura básica do documento da LEGaL.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <jmbl>
3   <head>
4     <!-- definicao dos localizadores -->
5     <locatorBase ... > ... </locatorBase>
6
7     <!-- definicao dos descritores -->
8     <descriptorBase ... > ... </descriptorBase>
9
10    <!-- definicao dos conectores -->
11    <connectorBase ... > ... </connectorBase>
12  </head>
13  <body>
14    <!-- especificacao da porta inicial -->
15    <port id="pStart" component="ncMain"
16      interface="iStart" />
17
18    <!-- especificacao das missoes -->
19    <mission ... >
20      ...
21      <!-- especificacao das midias -->
22      <media ... > ... </media>
23      ...
24    </mission>
25
26    <!-- especificacao de outros elementos -->
27    ...
28  </body>
</jmbl>

```

Listagem 1: Estrutura básica do código da linguagem proposta.

5. DESCRIÇÃO E AVALIAÇÃO DOS RESULTADOS

Como teste da linguagem, foi modelado um jogo com três missões, em que cada missão é composta por apenas uma mídia, respectivamente, um vídeo, um texto e uma imagem. O jogo foi modelado com localizações pertencentes ao Campus do Pici da UFC, em Fortaleza. O jogo segue os padrões *Search-and-Find* e *Follow-the-Path*, ele possui uma rota predefinida que deve ser executada pelo jogador até chegar a uma geolocalidade objetivo, ao final do jogo. Para a modelagem do jogo, foram criados os seguintes elementos:

1. As localizações, onde serão realizadas cada missão;
2. Os descritores, que determinam como as mídias serão exibidas;
3. Os nós de missão compostos pelos nós de mídia;
4. Os nós de mídia com as mídias a serem exibidas;
5. A porta de entrada, que define por qual nó o jogo se inicia;
6. As portas de comunicação dos nós, usadas na definição do fluxo do jogo;
7. Os conectores que definem os papéis dos nós de origem e destino;
8. Os elos que associam os papéis aos nós.

Para essa modelagem, é necessário ter as mídias no diretório especificado no documento. A Listagem 2 mostra o código que modela o jogo utilizando a linguagem proposta.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <jmbl id="gameExample" >

```

Tabela 1: Principais elementos e atributos da linguagem proposta.

| Categoria | Elemento | Token | Atributos |
|----------------|-----------------------|----------------|---|
| Localização | Base de Localizadores | locatorBase | id |
| | Localizador | locator | id, lat, lon, distance |
| Conteúdo | Nó de Missão | mission | id, mandatory, requirements |
| | Nó de Mídia | media | id, type, scr, descriptor |
| Relacionamento | Base de Conectores | connectorBase | id |
| | Conector | connector | id, condition, action |
| | Porta | port | id, component |
| | Elo | link | id, xconnector |
| Apresentação | Vínculo | bind | component, interface, role |
| | Base de Descritores | descriptorBase | id |
| | Descritor | descriptor | id, locator, duration, opacity, volume, fontSize, fontColor, fontWeight |

Tabela 2: Tipos de Mídias para a linguagem proposta.

| Tipo/Subtipo da Mídia | Extensão dos Arquivos |
|-----------------------|-----------------------|
| text/plain | txt |
| image/bmp | bmp |
| image/png | png |
| image/jpeg | jpg, jpeg |
| image/3d | obj, mtl |
| audio/mp3 | mp3 |
| video/mpeg | mpeg, mpg |

```

3 <head>
4   <locatorBase>
5     <locator id="lcDptoComp" lat="-3.7462006
6       " lon="-38.5747877" />
7     <locator id="lcB1951" lat="-3.746218"
8       lon="-38.575476" />
9     <locator id="lcBiblioFisica" lat="
10       -3.746218" lon="-38.575476" />
11   </locatorBase>
12   <descriptorBase>
13     <descriptor id="dDptoComp" locator="
14       lcDptoComp" />
15     <descriptor id="dB1951" locator="lcB1951
16       " />
17     <descriptor id="dBiblioFisica" locator="
18       lcBiblioFisica" />
19   </descriptorBase>
20   <connectorBase>
21     <conector id="onEndEnable" condition="
22       onEnd" action="enable" />
23   </connectorBase>
24 </head>
25 <body>
26   <!-- starting port -->
27   <port id="pStart" component="msVideo"
28     interface="pVideo" />
29
30   <!-- missions -->
31   <!-- first mission -->
32   <mission id="msVideo">
33     <port id="pVideo" component="mdVideo" />
34     <media id="mdVideo" type="video/mpeg"
35       scr="media/video1.mpg" descriptor="
36       dDptoComp" />
37   </mission>
38
39   <!-- second mission -->
40   <mission id="msText">
41     <port id="pText" component="mdText" />
42   </mission>
43
44   <!-- third mission -->
45   <mission id="msImage">
46     <port id="pImage" component="mdImage" />
47     <media id="mdImage" type="image/png" scr
48       ="media/image1.png" descriptor="
49       dBiblioFisica" />
50   </mission>
51
52   <!-- links -->
53   <!-- the end of video1 must enable the
54     execution of text1 -->
55   <link id="link1" xconnector="onEndEnable">
56     <bind component="msVideo" interface="
57       pVideo" role="onEnd" />
58     <bind component="msText" interface="
59       pText" role="Enable" />
60   </link>
61
62   <!-- the end of text1 must enable the
63     execution of image1 -->
64   <link id="link2" xconnector="onEndEnable">
65     <bind component="msText" interface="
66       pText" role="onEnd" />
67     <bind component="msImage" interface="
68       pImage" role="Enable" />
69   </link>
70 </body>
71 </jmb1>

```

Listagem 2: Código da modelagem de um jogo.

Cada elemento é identificado no documento por seu **id**. Na modelagem, as missões são, em ordem, msVideo, msText e msImagem e suas respectivas portas são pVideo, pText e

pImagem. As portas dão acesso ao conteúdo de cada nó de missão. Na linha 43, o elo link1 define que, após a realização da missão msVideo, a missão msText está disponível para realização. Na linha 49, o elo link2 define que a realização da missão msText habilita a realização da missão msImage.

Os elementos da linguagem utilizados nessa modelagem podem ser representados em um esquema visual ilustrado pela Figura 2. A imagem representa os principais elementos e como estão relacionados na definição do jogo. Dentre os elementos, são representados os nós de missão, os nós de mídias, as interfaces de comunicação dos elementos, os elos e os conectores.

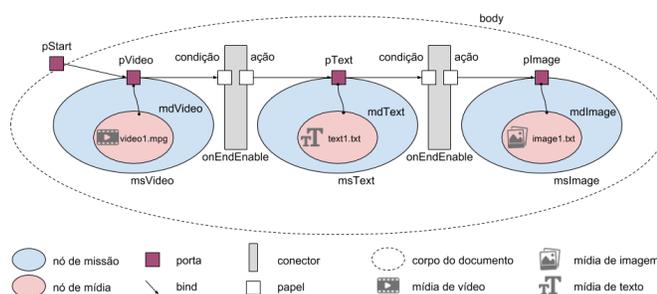


Figura 2: Visão estrutural da modelagem do jogo. Fonte: Elaborado pelo autor.

6. CONCLUSÃO E TRABALHOS FUTUROS

Este artigo apresentou um levantamento bibliográfico realizado focado nos temas de jogos móveis baseados em localização e desenvolvimento e modelagem de jogos pervasivos. Foram explicados alguns desafios enfrentados no desenvolvimento e representação desses jogos e como algumas abordagens de modelagem encontradas na literatura podem ser utilizadas na resolução de tais desafios. Foram abordadas algumas ferramentas de autoria de JMBLs propostas na literatura, realizando uma breve comparação entre algumas de suas características. Foi proposta uma linguagem declarativa, baseada na linguagem NCL, que deve permitir a representação de forma descritiva e clara das mecânicas e das regras de JMBLs, incluindo as mídias utilizadas nas mecânicas desses jogos, com suporte para aspectos temporais.

A linguagem será acoplada à LAGARTO, uma ferramenta Web de autoria de JMBLs desenvolvida na UFC. Também será implementado um gerador, um validador e um interpretador da linguagem que serão integrados à ferramenta LAGARTO. A validação da linguagem proposta será feita utilizando XML Schema. Outro trabalho futuro é dar suporte à implementação de mecanismos de checagem na ferramenta que gera JMBLs, para evitar possíveis inconsistências na modelagem do jogo ou nas mecânicas e regras criadas pelo autor do jogo.

7. REFERENCES

- [1] M. J. Antonacci, D. Muchaluat-Saade, R. Rodrigues, and L. F. G. Soares. Ncl: Uma linguagem declarativa para especificação de documentos hiperídia na web. *VI Simpósio Brasileiro de Sistemas Multimídia e Hiperídia-SBMídia2000, Natal, Rio Grande do Norte*, 2000.
- [2] E. Bethke. *Game development and production*. Wordware Publishing, Inc., 2003.
- [3] W. V. L. F. S. Carleandro De O. Nolêto, Fernando Trinta and M. Lima. Uma ferramenta de autoria web para edição de jogos móveis baseados em localização. 2015.
- [4] M. De Lange. From always on to always there: Locative media as playful technologies. *A. de Souza e Silva & DM Sutko (eds.), Digital cityscapes: merging digital and urban playspaces*, pages 55–70, 2009.
- [5] A. d. S. e Silva. *Digital cityscapes: Merging digital and urban playspaces*, volume 57. Peter Lang, 2009.
- [6] G. Hong. *Concepts and Modelling Techniques for Pervasive and Social Games*. PhD thesis, Norwegian University of Science and Technology, 2015.
- [7] P. Kiefer, S. Matyas, and C. Schlieder. Systematically exploring the design space of location-based games. In *Pervasive 2006 Workshop Proceedings, Poster presented at PerGames2006*, volume 7, pages 183–190, 2006.
- [8] L. Lehmann. Location-based mobile games. 2012.
- [9] C. Magerkurth, A. D. Cheok, R. L. Mandryk, and T. Nilsen. Pervasive games: bringing computer entertainment back to the real world. *Computers in Entertainment (CIE)*, 3(3):4–4, 2005.
- [10] M. Mernik, J. Heering, and A. M. Sloane. When and how to develop domain-specific languages. *ACM computing surveys (CSUR)*, 37(4):316–344, 2005.
- [11] A. Mont’Alverne. Jogos móveis locativos: uma proposta de classificação. *Contemporânea*, 10(1), 2012.
- [12] M. Montola. Exploring the edge of the magic circle: Defining pervasive games. In *Proceedings of DAC*, volume 1966, page 103, 2005.
- [13] D. Nicklas, C. Pfisterer, and B. Mitschang. Towards location-based games. In *Proceedings of the international conference on applications and development of computer games in the 21st century: ADCOG*, volume 21, pages 61–67, 2001.
- [14] I. M. Pinto, S. C. Botelho, R. C. de Souza, T. Sonogo, R. C. Goulart, and R. L. Campos. Plataforma saberlândia: Integrando robótica e multimídia no desenvolvimento de jogos educacionais. In *VII Brazilian Symposium on Computer Games and Digital Entertainment November, 10-12, 2008 Belo Horizonte-MG-BRAZIL*, page 177. Citeseer, 2008.
- [15] E. M. Reyno and J. Á. C. Cubel. Model driven game development: 2d platform game prototyping. In *GAMEON*, pages 5–7. Citeseer, 2008.
- [16] E. M. Reyno and J. Á. C. Cubel. A platform-independent model for videogame gameplay specification. In *Breaking New Ground: Innovation in Games, Play, Practice and Theory: Proceedings of the 2009 Digital Games Research Association Conference*, 2009.
- [17] A. C. Siang and G. R. K. Rao. Designing interactivity in computer games: A uml approach. *International Journal of Intelligent Games & Simulation*, 3(2), 2004.
- [18] R. von Bitter Rucker and R. Rucker. *Software Engineering and Computer Games*. Addison-Wesley, 2003.