

## 4

### Serviço de Inferência de Localização

O Serviço para Inferência de Localização (*LIS - Location Inference Service*) tem como objetivo a disponibilização de informações sobre localização de dispositivos móveis para aplicações ou outros serviços baseados em localização.

A inferência da localização é realizada utilizando informações sobre o sinal de redes IEEE 802.11, principalmente a intensidade. Escolhemos como estratégia realizar o cálculo da inferência em um servidor central, ao invés de no cliente. Nosso objetivo com isso é poupar recursos dos clientes móveis, além de utilizar uma infra-estrutura já disponível na MoCA.

Nas seções seguintes explicaremos em detalhes nossa estratégia para a localização de dispositivos móveis. Começaremos abordando a infra-estrutura necessária ao funcionamento do LIS, Seção 4.1. Os conceitos de regiões simbólicas e hierarquias de regiões são apresentados na Seção 4.2.2. Na Seção seguinte, 4.3, descreveremos a arquitetura do sistema, descrevendo seus componentes internos. A interface para consultas disponibilizada pelo serviço é mostrada na Seção 4.4 Por fim, na Seção 4.5, faremos algumas considerações finais sobre o serviço.

#### 4.1

##### Infra-Estrutura de Apoio

O LIS utiliza como infra-estrutura de apoio a arquitetura MoCA (*Mobile Collaboration Architecture*) [31, 32].

A MoCA consiste em um conjunto de serviços básicos para atender às necessidades de aplicações colaborativas para dispositivos móveis, de APIs e um arcabouço para facilitar a integração dessas aplicações com tais serviços. As APIs e os serviços básicos foram projetados para serem genéricos e flexíveis,

assim como úteis para diferentes tipos de aplicações colaborativas cientes de contexto.

Os serviços da arquitetura MoCA utilizados pelo LIS são o *Monitor* e o *CIS* (*Context Information Service*). O *Monitor* é um *daemon* executando em cada dispositivo móvel com a responsabilidade de coletar dados relacionados ao ambiente computacional do dispositivo móvel e enviar estes dados para o *CIS* executando em um (ou mais) nó(s) da rede fixa. Os dados coletados pelo *Monitor* e necessários para a operação do LIS incluem a conectividade do dispositivo, o ponto de acesso corrente, e principalmente a lista de todos os pontos de acesso audíveis pelo dispositivo móvel e suas intensidades de sinal.

Para realizar essa coleta o monitor utiliza uma função do protocolo IEEE 802.11 conhecido como *scanning*. Através dessa função um dispositivo envia um *broadcast* que será “ouvido” por todos os pontos de acesso em seu raio de cobertura. Ao receber a solicitação, o ponto de acesso responde com informações específicas do ambiente/rede, como por exemplo, a intensidade do sinal observado pelo ponto de acesso em relação ao dispositivo executando o *Monitor*.

Além desses dados o *Monitor* ainda coleta informações sobre a disponibilidade de recursos no dispositivo, como uso de CPU, memória e energia e alguns parâmetros de rede como endereço *MAC*, IP e máscara de rede.

O *CIS* é um serviço distribuído onde cada servidor *CIS* recebe e processa informações do ambiente computacional de *monitores* executando em dispositivos móveis sob a sua responsabilidade. A associação de um *Monitor* a um servidor *CIS* é feita a partir do seu endereço *MAC*, sendo esse o identificador único de um dispositivo no sistema. O *CIS* também é capaz de receber subscrições de outros serviços (ou aplicações), notificando tais clientes se uma mudança no estado do dispositivo é do interesse do cliente.

Apesar de ter sido projetado para a MoCA, em princípio, o LIS pode funcionar em qualquer outro ambiente no qual tenha acesso à informação (periodicamente atualizada) sobre a intensidade de sinal de rádio-frequência de todos os pontos de acesso alcançáveis por um dispositivo móvel.

## 4.2

### Regiões Simbólicas

O LIS provê informações sobre a localização de dispositivos através do conceito de “Regiões Simbólicas”. “Regiões Simbólicas” são nomes dados a regiões geográficas e/ou lógicas bem definidas, como salas, corredores, prédios, ruas, quadras, cruzamentos, andares, departamentos, secretarias, etc, em ambientes fechados ou abertos que podem ser de interesse para aplicações baseadas em localização. Aplicações que utilizam os serviços do LIS estão interessadas principalmente na entrada ou saída de dispositivos de “Regiões Simbólicas”.

“Ponto de Referência”, como dito antes, é um local bem definido onde são coletados amostras dos sinais de cada ponto de acesso no ambiente onde se deseja realizar a localização dos dispositivos móveis. Geralmente, pontos de referências são amostrados em regiões de interesse, de forma que se possa detectar quando um dispositivo está em uma determinada região.

#### 4.2.1

##### Mapeamento entre Pontos de Referência e Regiões Simbólicas

Para o LIS inferir a localização de um dispositivo móvel, é necessário definir uma forma de mapeamento entre “Pontos de Referência” e “Regiões Simbólicas”. Esse mapeamento é necessário para transformar o resultado da inferência, um ponto de referência, em uma região simbólica. A definição dos pontos de referência, definição das regiões simbólicas e coleta dos padrões de sinais para cada ponto de referência são tarefas executadas durante a implantação do sistema.

O mapeamento de pontos de referência para regiões simbólicas pode ser realizado de duas formas, que chamamos de “Mapeamento Geométrico” e “Mapeamento Simbólico”.

No “Mapeamento Geométrico” é necessário dispor de um mapa (planta baixa) da região geográfica no qual se deseja detectar a localização de um dispositivo, região de interesse. Nesse mapeamento as regiões simbólicas são definidas como coordenadas, por exemplo, para uma região em forma de retângulo podemos anotar os pontos superior esquerdo e inferior direito. Essa marcação pode ser realizada graficamente com a ajuda do mapa. Nesse

mapeamento, são escolhidos “Pontos de Referência” com suas respectivas coordenadas. De posse das coordenadas de cada ponto de referência e das regiões simbólicas é possível determinar quais pontos de referências estão relacionados a cada região simbólica.

Para regiões de interesse onde não está disponível um mapa, o LIS oferece o “Mapeamento Simbólico”. Nesse mapeamento, como não existem informações de coordenadas, regiões são anotadas apenas simbolicamente. Ou seja, regiões são definidas apenas por um nome simbólico. “Pontos de Referência” são amostrados da mesma forma como no “Mapeamento Geométrico”. No entanto, em vez de associar uma coordenada a cada ponto de referência, associa-se o nome de uma região simbólica a cada ponto de referência. Ou seja, no “Mapeamento Simbólico” pontos de referência ficam associados diretamente a uma região, e não a uma coordenada, como no “Mapeamento Geométrico”.

Quando se usa o “Mapeamento Geométrico” é possível redefinir regiões simbólicas sem a necessidade de re-amostrar os pontos de referência. Isso ocorre porque nesse mapeamento existem coordenadas tanto para regiões como para os pontos de referências. Assim, é necessário, na redefinição das regiões, somente re-agrupar os pontos de referência por região de acordo com suas coordenadas. Outra vantagem do “Mapeamento Geométrico” é uma inferência mais precisa.

Apesar das vantagens apresentadas pelo “Mapeamento Geométrico”, o “Mapeamento Simbólico” é mais fácil de ser executado, já que não é necessário dispor de um mapa e não existe a preocupação com o tratamento de coordenadas, permitindo uma implantação rápida e em regiões pouco conhecidas.

#### 4.2.2

#### **Regiões Simbólicas Hierárquicas**

Para facilitar o uso do LIS por aplicações criamos o conceito de regiões simbólicas aninhadas. O aninhamento de regiões é feito de forma que regiões simbólicas “maiores” englobem regiões simbólicas “menores”, criando dessa forma um conceito de topologia (hierarquia) de uma região simbólica.

Uma estrutura em árvore é a forma natural de representar a composição de uma região simbólica em suas sub-regiões. Nessa árvore, nós folhas re-

presentam regiões simbólicas que nomeamos de “regiões atômicas”, enquanto nós intermediários representam regiões simbólicas compostas de outras regiões simbólicas.

Durante a implantação do LIS são definidas apenas regiões atômicas, ou seja, sem a noção de hierarquia. Aplicações que necessitem definir regiões com granularidades maiores podem usar as regiões atômicas, previamente definidas durante a implantação do LIS, para compor regiões simbólicas com maiores granularidades, formando dessa maneira uma hierarquia de regiões simbólicas.

Através do conceito de aninhamento de “Regiões simbólicas” podemos representar regiões simbólicas com diferentes granularidades. Por exemplo, tanto o campus da PUC-Rio como a sala 501 do 5<sup>o</sup> andar do RDC podem ser caracterizadas por uma região simbólica. Com isso, aplicações podem ter acesso a informações de dispositivos que estão em regiões grandes, como o campus da PUC-Rio, como em regiões bem limitadas, como a sala 501 do prédio do RDC da PUC-Rio.

Como exemplo, a hierarquia do campus da PUC-Rio pode ser pensada como mostrado na Figura 4.1. Portanto, quando falamos que um dispositivo está localizado na região simbólica *Room 501*, de acordo com a hierarquia definida na Figura 4.1, ele também está localizado nas regiões simbólicas *PUC-Rio/RDC/5th Floor*.

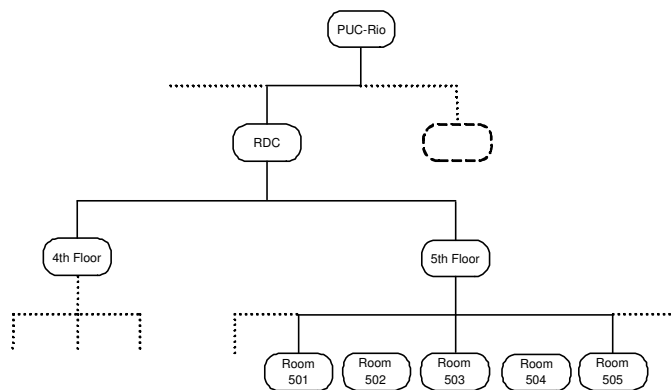


Figura 4.1: Diagrama da hierarquia de regiões da PUC-Rio.

A inferência de localização executada pelo LIS é realizada somente com base em regiões atômicas, ou seja, o LIS somente detecta a migração de dispositivos com relação a regiões atômicas. Essa restrição determina a forma como hierarquias podem ser compostas. Nós folhas de hierarquias devem ser regiões atômicas (definidas pelo LIS) ou regiões previamente definidas em outras hierarquias. Caso existam nós folhas das hierarquias que não sejam

regiões atômicas, não é possível ao LIS determinar a localização de dispositivos nessas regiões.

Em princípio, os algoritmos de localização usados no LIS independem do conceito de regiões hierárquicas. Quando são realizadas consultas sobre regiões simbólicas hierárquicas, essa consulta é traduzida para consultas sobre regiões atômicas (sub-consultas), de forma que possa ser calculada sua resposta (sub-resposta), e depois unidas às sub-respostas para formar à resposta a consulta original.

### 4.3

#### Arquitetura

A Figura 4.2 mostra a interação entre os vários elementos da infraestrutura usada pelo LIS. Inicialmente, (1) *Monitores* que executam em dispositivos móveis coletam e enviam ao *CIS* suas informações contextuais, que inclui a intensidade do sinal recebido de todos os pontos de acesso. (2) Para ser capaz de perceber a mudança de região de dispositivos móveis o LIS registra-se junto ao *CIS* para receber informações periódicas sobre a intensidade do sinal de rádio-frequência percebida por cada dispositivo móvel. A periodicidade do envio das informações é determinada pelo LIS no momento do registro junto ao *CIS*. (3) As informações recebidas pelo LIS, do *CIS*, são as intensidades de sinais de cada ponto de acesso coletadas pelos *monitores*. (4) A partir dessas informações o LIS pode inferir a localização do dispositivo e servir aplicações com as informações requisitadas.

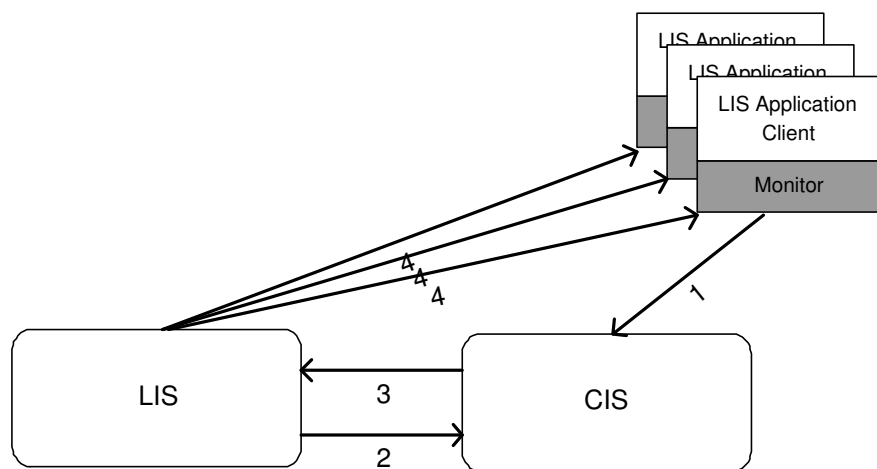


Figura 4.2: Interação entre os elementos da infra-estrutura do LIS.

O LIS provê duas formas de interação com aplicações, uma síncrona e outra assíncrona. No modo de comunicação síncrona, aplicações fazem requisições para consultar informações e bloqueiam até receber uma resposta. Essa forma de comunicação, adotada em *middlewares* tradicionais, funciona bem em ambientes onde a latência de comunicação é pequena e o canal de comunicação é confiável. Entretanto, esse modo de comunicação apresenta algumas desvantagens em cenários móveis [33].

Na forma de comunicação assíncrona, uma aplicação, ao necessitar de informações de localização de um determinado dispositivo, se inscreve no serviço de eventos do LIS para que seja notificado quando da ocorrência de um evento de localização. Os eventos de localização publicados pelo LIS são a mudança de região de dispositivos móveis e entrada ou saída de dispositivos móveis de regiões. Dessa forma aplicações não bloqueiam a espera de uma resposta. Outra vantagem nessa forma de comunicação é que aplicações tomam conhecimento tão logo a informação de localização esteja disponível.

A Figura 4.3 mostra a arquitetura do serviço de localização. Basicamente, o serviço realiza três funções principais (Inferência de localização, Processamento de consultas síncronas e Processamento de consultas assíncronas) que descreveremos a seguir.

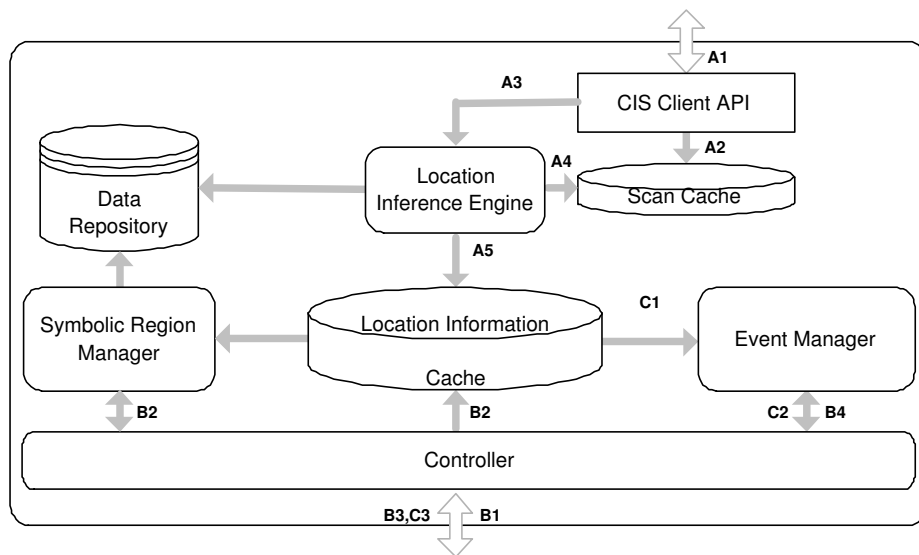


Figura 4.3: Principais componentes da arquitetura do LIS.

## **A - Inferência de localização**

Após receber as informações contextuais do *CIS* (em A1) o componente CIS Client API processa as informações, armazena a informação processada no Scan Cache (A2) e notifica o Location Inference Engine que novos dados estão disponíveis (A3). O componente Location Inference Engine executa o algoritmo de inferência (A4), baseado nas informações disponíveis nos componentes Data Repository e Scan Cache, e armazena a informação de localização no Location Cache Manager (A5).

## **B - Processamento de uma consulta síncrona**

Ao receber uma consulta síncrona (B1), o Controller analisa a requisição e determina o seu tipo. Nesse caso há duas possibilidades: uma consulta padrão sobre informações de localização (modo síncrono); ou uma requisição de registro de interesse em algum evento (modo assíncrono). No caso de uma consulta sobre informações de localização, o Controller consulta o Location Cache Manager/Symbolic Region Manager (B2) e responde a requisição (B3). No caso de um registro de interesse em eventos a requisição é repassada ao Event Manager (B4), que trata a requisição.

## **C - Processamento de uma consulta assíncrona**

O componente Location Cache Manager (C1) notifica o Event Manager sempre que é detectado que um dispositivo mudou de localização. O Event Manager, por sua vez, verifica se há algum registro de interesse relacionado a este dispositivo ou região. Caso haja tal registro, o Event Manager notifica o Controller (C2) que então se encarrega de notificar as aplicações (C3).

A seguir descreveremos os principais componentes da arquitetura, suas responsabilidades e alguns detalhes de implementação..

## **CIS Client API**

O componente CIS Client API é responsável por interagir com o *CIS* e registrar o interesse do LIS em obter informações contextuais (intensidade



de sinal) de dispositivos móveis, pelo recebimento e o tratamento de tais informações.

O *CIS* expõe uma interface de registro de interesse de informações de contexto baseado no *matching* de predicados sobre variáveis de contexto na forma de uma expressão *SQL*. O LIS registra-se junto ao *CIS* utilizando o *tag ALL\_DEVICES* e o predicado *OnLine = True*. O *tag ALL\_DEVICES* especifica o interesse em todos os dispositivos sendo monitorados pelo *CIS*. O registro junto ao *CIS* pode ainda ser feito com base em um dispositivo específico. Entretanto, ao se registrar com o *tag ALL\_DEVICES*, o LIS é capaz de realizar a descoberta espontânea de dispositivos, o que não ocorre no outro caso. O predicado *OnLine = True* especifica que o LIS está interessado em receber somente informações de dispositivos *on-line*. Desta forma, o LIS é capaz de perceber quando um determinado dispositivo foi desligado.

Além do predicado, o componente CIS Client API informa, no momento do registro junto ao *CIS*, a periodicidade na qual está interessado em receber informações sobre padrões de sinais de rádio-freqüência dos dispositivos móveis. Dessa forma, a cada período de tempo, o *CIS* envia para o LIS as informações de todos os dispositivos registrados pelo LIS.

A freqüência de envio das informações é um parâmetro da categoria da aplicação em questão, sendo um parâmetro configurável do serviço. Para aplicações que necessitam de informações atualizadas freqüentemente, como um rastreamento de pessoas em um ambiente interno, é necessário um envio com freqüência alta (ex., a cada 1 segundo). Entretanto, outras categorias de aplicações como o envio de mensagens a regiões (como realizado na aplicação *Notes in the Air* [34]), não requerem uma atualização tão freqüente da localização do usuário (bastando ser, por exemplo, a cada 10 segundos).

Todas as informações recebidas pelo componente CIS Client API são processadas e armazenadas no componente Scan Cache.

## Scan Cache

Este componente tem a responsabilidade de armazenar os últimos  $n$  padrões de sinais recebidos para cada dispositivo.

O *cache* dos últimos  $n$  padrões é realizado com o objetivo de prover uma inferência mais confiável. No momento da inferência da localização, o componente Location Inference Engine utiliza a média (ou algum outro valor

derivado destas amostras) e não apenas o último valor recebido. Isso permite que variações na intensidade do sinal comuns em sinais de rádio-frequência, principalmente em ambientes internos, mesmo enquanto o dispositivo móvel permanece estacionário, sejam atenuadas. Tais variações podem ocorrer devido a diversos fatores, como por exemplo, o trânsito de pessoas no ambiente.

É importante salientar que essa estratégia acarreta um “atraso” na detecção da mudança de localização de um dispositivo, já que amostras antigas são sempre levadas em consideração na inferência. O atraso na detecção é diretamente proporcional à periodicidade do envio das informações pelo *CIS* ( $p$ ) e ao tamanho do cache ( $n$ ). Desta forma, o tamanho do cache (com base no máximo “atraso” da detecção de localização desejado) pode ser determinado usando a seguinte relação:

$$atraso = n \times p.$$

O valor de  $n$  representa um compromisso entre precisão e confiabilidade da informação de localização, e pode ser configurado no LIS de forma dinâmica.

Este componente é implementado usando um *buffer* circular de tamanho  $n$ . Os dados são armazenados internamente em uma tabela *hash* onde a chave é o identificador do dispositivo móvel (endereço *MAC*) e o conteúdo são as últimas  $n$  amostras recebidas.

## Data Repository

O componente **Data Repository** armazena de forma persistente as informações sobre definições de regiões atômicas, pontos de referência e hierarquias, de forma persistente. Junto com os pontos de referências são armazenadas os respectivos padrões de sinais para cada ponto de acesso audível.

Outra função desse componente é a tradução de pontos de referências para regiões simbólicas. Essa tradução é realizada de duas formas distintas, dependendo do tipo de mapeamento utilizado. No caso do mapeamento geométrico o componente transforma a coordenada, de um ponto de referência, para uma região com base nas coordenadas de cada região (o componente verifica a qual região a coordenada pertence). Quando o mapeamento é simbólico o componente transforma o ponto de referência em uma região de acordo com a associação definida na ocasião do mapeamento.

## Location Inference Engine

Este componente tem a função de inferir a localização dos dispositivos móveis usando um dos algoritmos descritos no Capítulo 2. A inferência da localização é realizada com base nas informações de intensidade de sinal disponibilizadas pelos componentes *Scan Cache* e *Data Repository*.

O componente *Location Inference Engine* define uma interface, *InferenceAlgorithm*, que deve ser implementada pelos algoritmos de localização. Dessa forma, qualquer algoritmo pode ser incorporado ao LIS, desde que seja capaz de implementar tal interface.

A interface *InferenceAlgorithm* define, entre outros, um método, *InferenceDiff[] sortReferencePoints(Scan[] scans)*, para ordenar os pontos de referência de acordo com a similaridade em um conjunto de amostras. A estrutura *InferenceDiff* contém um ponto de referência e a sua distância ao conjunto de amostras passado como parâmetro. O método retorna uma lista dessa estrutura ordenada pela distância. Como exemplo, a distância para os algoritmos *Histograma* e *MNN* são, respectivamente, a probabilidade da amostra pertencer a aquele ponto de referência e o inverso da distância euclidiana.

Para implementar um novo algoritmo de inferência é necessário somente implementar esse método. Para isso deve-se ordenar os pontos de referência, segundo alguma regra, em relação ao parâmetro *Scan[] scan* que representa a seqüência atual de valores de intensidades de sinal do dispositivo que se deseja calcular a localização.

De posse destas informações o componente pode inferir a localização do dispositivo de duas formas distintas, dependendo do tipo de mapeamento utilizado. No mapeamento geométrico, onde existe informações de coordenadas dos pontos de referência e regiões, o componente seleciona os  $n$  primeiros pontos de referência, onde  $n$  é o número de vizinhos, e calcula a coordenada do ponto inferido computando a média ponderada das posições dos pontos de referência selecionados utilizando como peso a distância. No caso do mapeamento simbólico, onde não existe a informação de coordenadas, o componente simplesmente seleciona o primeiro ponto de referência da lista. O resultado da inferência, uma coordenada ou um ponto de referência, é sempre transformado em uma região simbólica através de uma consulta ao componente *Data Repository*.

Este componente alimenta a base de dados do componente *Location Cache Manager*.

## Location Cache Manager

Este componente é responsável por armazenar a informação de localização para cada dispositivo. Além dessa informação esse componente mantém uma estrutura de dados relacionando os dispositivos por região.

A implementação desse componente contém duas tabelas de dispersão (*hash*). A primeira armazena a localização (região atômica) de cada dispositivo, sendo indexada pelo seu endereço *MAC*. A segunda tabela de dispersão armazena os dispositivos localizados por região (atômica). Essa tabela é indexada pelo identificador da região e contém como valores listas de identificadores de dispositivos.

Para resolver consultas sobre regiões não-atômicas, esse componente mantém duas tabelas *hash* adicionais. A primeira mapeia cada região para uma lista de suas regiões atômicas. Com isso, consultas sobre uma região não-atômica podem ser respondidas obtendo a lista de regiões atômicas que constituem a região em questão. A segunda tabela *hash*, uma tabela inversa, mapeia cada região atômica para as regiões, de todas a hierarquias, que a contém. Essa tabela é utilizada para notificações assíncronas onde o usuário é notificado sobre mudanças de dispositivos em regiões, sendo necessária para identificar as regiões não-atômicas onde existe o interesse de clientes. As informações sobre a constituição de regiões são obtidas a partir do componente *Symbolic Region Manager*.

Quando este componente detecta que um dispositivo mudou de região atômica, o mesmo consulta na tabela *hash* todas as regiões (não atômicas) que contêm a região atômica em questão. De posse de todas as regiões, atômicas ou não, que o dispositivo tenha entrado ou saído, esse componente repassa essa informação para o componente *Event Manager* para que seja realizada a notificação de qualquer cliente do serviço (por exemplo, aplicações) que tenha previamente manifestado o interesse em qualquer dessas regiões, ou no dispositivo.

Para responder a requisições síncronas o *LIS* consulta este componente, que contém a última informação de localização, respondendo a seus clientes de forma mais rápida do que se fosse necessário inferir a localização do dispositivo no momento da requisição.

Informações são apagadas dessa cache no momento em que o *LIS* detecta que o dispositivo está *off-line*. Isso ocorre quando o *LIS* deixa de receber informações sobre o dispositivo.

## Event Manager

Este componente é responsável por notificar as aplicações interessadas (e registradas) em receber informações sobre a localização de dispositivos. Ou seja, é o componente responsável pela comunicação assíncrona entre o LIS e seus clientes. Outras funções básicas, como registro e cancelamento no interesse de eventos, também são administrados por este componente.

Há dois tipos de notificações controladas por este componente: notificações relativas a dispositivos e notificações relativas a regiões. Na notificação relativa a um dispositivo esse componente informa os clientes toda vez que um determinado dispositivo muda de localização através do envio de um evento. Para isso, a cada notificação de mudança de região de dispositivo realizada pelo componente **Location Cache Manager**, o **Event Manager** verifica se existe algum cliente interessado nessa informação e se for o caso, notifica-o.

Na notificação relativa a uma região esse componente informa os clientes toda vez que a composição de uma determinada região muda, ou seja, quando um dispositivo entra ou sai de uma determinada região. Para isso, a cada notificação de mudança da composição de uma região realizada pelo componente **Location Cache Manager**, o **Event Manager** verifica se existe algum cliente interessado nessa informação e notifica-o.

Para cada mudança de região de um dispositivo esse componente dispara dois eventos. O primeiro de saída do dispositivo de uma região, e o segundo de entrada deste dispositivo em uma nova região.

Para a implementação desse componente usamos a *API Event-based Communication Interface (ECI)* para comunicação baseada em eventos da MoCA [35]. Esta *API* provê facilidades para implementar comunicação assíncrona usando o paradigma *publish/subscribe*, onde processos podem subscrever-se em eventos relacionados a algum “assunto”/“predicado”, e/ou publicar eventos relacionados a um “assunto”. No nosso caso, o LIS é o publicador de eventos onde o “assunto” ou é o identificador de uma região, ou o identificador de um dispositivo.

## Symbolic Region Manager

Este componente expõe uma interface para definir (criar, modificar e apagar) e consultar hierarquias de regiões simbólicas. Além de consultar

hierarquias, clientes podem também se registrar para serem notificados quando uma hierarquia for adicionada, removida ou modificada.

Para evitar a consulta constante a esse componente, clientes podem se registrar para serem notificados quando hierarquias são modificadas. Dessa forma, os clientes armazenam localmente uma cópia das hierarquias nas quais os mesmos estão interessados.

## 4.4

### Interface para Consultas

Dois tipos de dados são chaves nas interfaces de consulta do LIS, regiões e hierarquias. Esses dados estão representadas pelas classes *Region* e *Hierarchy*, mostrados na Figura 4.4.

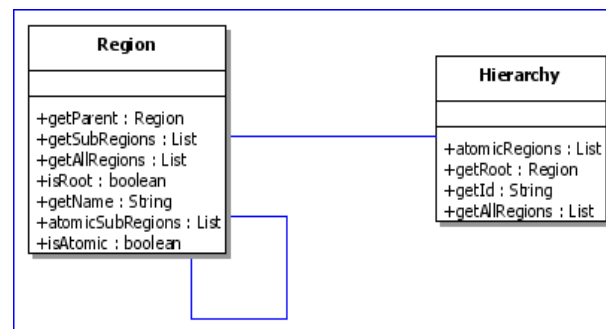


Figura 4.4: Diagrama de classes de hierarquia e região.

A classe *Region* representa diretamente o conceito de regiões simbólicas, inclusive a idéia de regiões hierárquicas. O método *getSubRegions()* retorna a lista de sub-regiões da região em questão, que por sua vez também são objetos da classe *Region*. Regiões atômicas também são representadas por essa classe e podem ser verificadas através do método *isAtomic()*. O método *getName()* retorna o nome da região. O identificador de uma região tem a seguinte forma:

$$\langle hierarchyId \rangle : / \langle region \rangle / \langle region \rangle / \langle region \rangle$$

Por exemplo, o identificador da região “Room 501” apresentada na Figura 4.1 é “Hierarquia01:/PUC-Rio/RDC/5th Floor/Room 501”, onde “Hierarquia01” é o identificador da hierarquia mostrada.

O identificador de uma região atômica é um nome simples, desde que regiões atômicas não pertençam a nenhuma hierarquia em particular e não apresentem sub-regiões.

Uma hierarquia é composta por uma região raiz. A estrutura hierárquica está contida dentro da própria região raiz. O método *getRoot()* da classe *Hierarchy* retorna essa região. Os métodos *getParent()* e *getSubRegions()* da classe *Region* nos permite navegar pela hierarquia.

Como dito antes, o LIS disponibiliza duas formas de comunicação, síncrona e assíncrona. Para a forma de comunicação síncrona existem métodos bloqueantes para consultar e manter regiões. Na forma de comunicação assíncrona todos os métodos funcionam através de funções de *callback* que serão executadas quando um determinado evento ocorrer, seja ele de consulta ou manutenção de regiões e dispositivos. Uma lista completa de toda a interface disponibilizada pelo LIS pode ser encontrada no Anexo A.

## 4.5

### Considerações Finais

Apesar de redes 802.11 estarem se tornando cada vez mais difundidas e sendo usadas em nosso dia a dia, essa tecnologia ainda predomina em ambientes internos. Entretanto, para diversas aplicações seria interessante um serviço de localização capaz de funcionar em ambientes internos e externos. Dado que a tecnologia *GPS* [15, 16] é bastante disseminada em ambientes externos, cabe aqui discutir quais mudanças seriam necessárias na arquitetura do LIS afim de que está possa acomodar também uma localização através dessa tecnologia.

Antes de qualquer coisa, naturalmente, os dispositivos móveis teriam que estar equipados, além de uma placa de rede IEEE 802.11, com um receptor *GPS*. É cada vez mais comum encontrar esse tipo de hardware em dispositivos móveis, como por exemplo, em celulares.

O LIS teve seu projeto baseado em componentes, de forma a tornar rápida e fácil a incorporação de novos requisitos com a adição de novos componentes. Por isso, incorporar a tecnologia *GPS* no LIS torna-se uma tarefa bastante simples e direta, que requer mudanças mínimas na arquitetura do LIS (mostradas na Figura 4.5).

Na Figura 4.5 o componente 802.11 Based Location representa os componentes CIS Client API, Scan Cache e Location Inference Engine da arquitetura apresentada na Figura 4.3. Nesta nova arquitetura o componente GPS Based Location seria responsável apenas por receber do dispositivo as coordenadas *GPS* e repassá-las diretamente ao componente Location Cache Manager. Uma

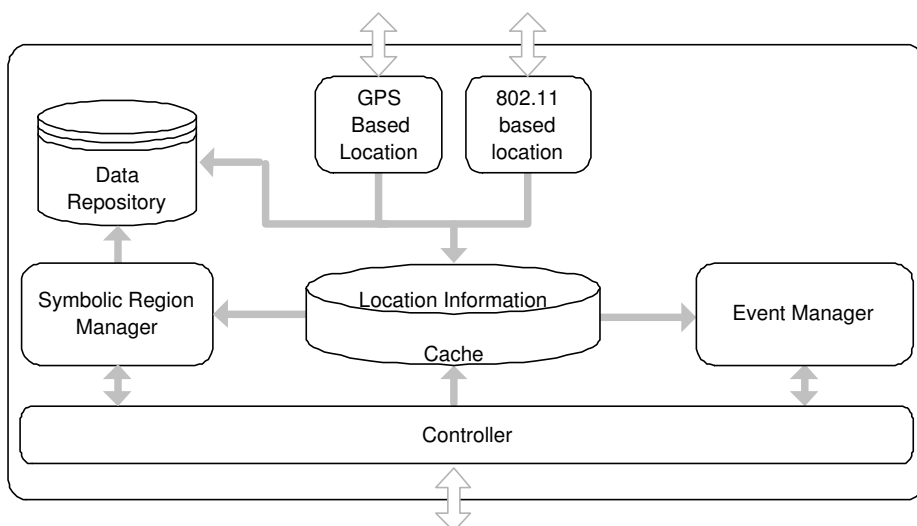


Figura 4.5: Arquitetura revisada do LIS para incorporação de *GPS*.

questão a ser resolvida é a de como traduzir a informação de localização em coordenadas a localização em termos de regiões simbólicas. No caso do mapeamento geométrico isso pode ser facilmente realizado anotando para cada região, não somente as suas dimensões e sua coordenada local (isto é, relativa a um marco referencial), mas também as suas coordenadas globais obtidas via *GPS* (latitude, longitude, altitude). Portanto, para cada prédio ou construção seria necessário conhecer pelo menos a posição *GPS* de três pontos. Com esses pontos e as coordenadas locais seria possível então traduzir qualquer coordenada *GPS* para uma região simbólica e vice-versa.

No caso do mapeamento simbólico, como não existe informação geométrica, seria necessária uma vinculação explícita entre a coordenada *GPS* e uma região simbólica, por exemplo, considerando sempre o ponto central da região.

A partir dessa integração com *GPS* surge uma nova necessidade que é a disponibilização de informações em termos de coordenadas (globais ou locais). Para acomodarmos esse novo requisito seria necessário apenas que o componente **Location Cache Manager** armazenasse os dois tipos de informações para cada dispositivo (coordenadas e regiões simbólicas). Seriam necessárias também mudanças nas *APIs* de consulta apresentadas na Seção 4.4. Por exemplo com a duplicação dos métodos onde temos, como parâmetro ou tipo de retorno, uma região simbólica substituindo-a por uma coordenada.

Desta forma seria possível acomodar qualquer sistema de inferência de localização nesta arquitetura, desde que esse sistema forneça a informação de



localização em termos de coordenadas ou em termos de regiões simbólicas, tais como as do LIS.