

Uma arquitetura de controle de processos ambientais baseada em geoprocessamento

An architecture for geoprocessing-based environmental process control

Luiz Alberto Oliveira Lima Roque*

Atividades humanas têm alterado os recursos naturais de maneira dramática. A fim de lutar contra essa degradação feroz, há uma grande necessidade de se criarem sistemas de informação geográficos orientados ao monitoramento ambiental. A sociedade deve rastrear a situação real dos seus recursos ambientais, e isso pode ser executado por meio do processo de seleção dos objetos relevantes para o monitoramento, medição de suas características ou atributos e análise de todos os dados fornecidos por meio de modelos matemáticos e estatísticos; de simulação e de outras ferramentas. Há um padrão, meta ou situação desejada para o comportamento de um sistema ambiental e dados originados do mundo real devem ser comparados a esse padrão. Os dados a serem coletados são em geral dispersos geograficamente e de diferentes tipos, adquiridos por sensores específicos ou por pesquisadores. Assim, este artigo apresenta um Modelo Orientado ao Objeto, baseado em dados geoprocessados, e pode ser pensado como uma estrutura para todo tipo de sistemas de informação voltados ao controle e monitoramento de processos ambientais. Este modelo é orientado para a Internet, no sentido de que o dado a ser coletado pode estar geograficamente distribuído. Após sua captura, o dado é enviado, pela Internet, a Sistemas Gerenciadores de Base de Dados (SGBD) específicos, para ser armazenado. Depois, esse dado bruto pode ser disseminado também pela Internet, para pesquisadores ou outras pessoas interessadas. Aplicativos baseados neste modelo poderão fazer diversas

Human activity has changed natural resources in dramatic ways. In order to fight this fierce degradation, there is a great need of geographic information systems oriented to environmental monitoring. Society must track the real situation of its natural environmental resources, an action which can be accomplished by selecting important objects to be monitored, by measuring their characteristics or attributes, and by analyzing all the data using mathematical models, statistics, simulation and other tools. There is some standard, goal or desired situation for the behavior of an environmental system and, opposing this standard, the data originated from the real world must be compared. Data to be collected are of different kinds and geographically dispersed, caught by sensors disposed on specific hardware, known as Data Acquisition Systems. Thus, we present an Object-Oriented Model, based on geoprocessed data, which could be considered as a framework for any kind of control information system focused on environmental processes. This model is Internet-oriented, meaning that data to be collected can be geographically distributed. After being captured, data are sent, via Internet, to specific Data Base Management Systems (DBMS) to be stored. This raw data can be disseminated, also via Internet, to researchers or other interested individuals. Some analysis will be done with the spatial data stored in the DBMS before being presented to users. If, for various reasons, these data remain within unsafe concentration limits, some processes will be triggered in order to avoid environmental damages to air, soil or water. With this framework, models could also be developed for different kinds of ecosystems

* Mestre em Engenharia de Computação. Doutorando em Engenharia de Reservatórios e Modelagem Computacional. Instituto Federal Fluminense – *Campus* Macaé. Email: luizlimaroque@gmail.com. Endereço: Rod. Amaral Peixoto, Km 164 Imboassica 27793-030 – Macaé, RJ, Brasil

análises com os dados espaciais armazenados no SGBD. Se os dados monitorados estiverem fora dos limites seguros de concentração, devido a diversos motivos, alguns processos serão disparados, no intuito de evitar danos ambientais ao ar, às águas e aos solos. Com esta estrutura, pode-se desenvolver modelos para diferentes tipos de ecossistemas e arquivá-los numa base de dados específica, visando a seu posterior reuso. Um protótipo baseado em Java foi desenvolvido para mostrar as funcionalidades do modelo.

Palavras-chave: Modelo Orientado ao Objeto. Controle de processos. Monitoramento ambiental. Modelos estatísticos e matemáticos. Geoprocessamento. Java.

and archived in a specific database aiming at being reused later. A Java based prototype was developed to show the model functionalities.

Key words: Object-Oriented Model. Database. Process control. Environmental monitoring; Mathematical and statistical models. Geoprocessing. Java.

Introdução

Desde o advento da revolução industrial, no século dezoito, a intervenção humana sobre a natureza se tornou muito intensa. Junto com a invenção da máquina a vapor, os homens adquiriram a tecnologia para automatizar processos produtivos e criar equipamentos como locomotivas, navios, máquinas de costura e dispositivos para extração de carvão, o que ampliou as relações comerciais. Desmatamentos provocados por incêndios têm crescido em progressão geométrica, para dar lugar a novas cidades, parques industriais, zonas agrícolas, pastos e estradas. Desde então, o lançamento de monóxido de carbono à atmosfera tem crescido enormemente, agravando o efeito estufa, o que provocou um aquecimento global sem precedentes na idade moderna, acompanhado de consequências como derretimento de icebergs e aumento no nível do mar.

Os gases emitidos pelas indústrias e automóveis reduziram a qualidade do ar metropolitano, causando diversas doenças. A expansão demográfica ocorreu sem o devido planejamento sanitário, degradando os recursos aquáticos. Desta forma, um grande porcentual da população mundial sofre de doenças relacionadas à poluição da água. Assim, pode-se concluir que o monitoramento e o controle de processos ambientais são essenciais à sobrevivência humana. Num trabalho, Ongley & Booty (1999), sugerem que a situação real dos recursos ambientais seja determinada pela seleção dos parâmetros adequados ao monitoramento e pela análise das características ou atributos, de todos aqueles dados juntos ou isolados, usando modelos matemáticos, estatísticos, simulação e outras ferramentas.

O objetivo deste artigo é apresentar um modelo para geoprocessamento de dados, orientado ao objeto, à Internet, independente do domínio em que está sendo tratado o problema (se é o caso de poluição aquática ou aérea, prevenção de incêndios, monitoria de tráfego, controle de produtividade, etc.). A mudança do domínio ambiental implicaria, no máximo, a alteração do tratamento estatístico necessário. O modelo é baseado nas especificações da Linguagem de Modelagem Unificada (UML). Sua estrutura foi concebida para operar um sistema ou Arquitetura de Monitoramento Ambiental (AMA) genérico, estabelecendo métodos para medição, armazenamento e visualização de variáveis ou parâmetros ambientais, conforme exposto nos seguintes passos:

i) Entrada, definição e registro das variáveis ambientais a serem monitoradas, através de uma interface padrão, na qual o usuário do sistema de monitoramento definirá o que, quando e onde monitorar. A variável monitorada será qualificada, recebendo nome, unidade de medida, além de forma, locais e periodicidade de coleta.

ii) Armazenamento das variáveis monitoradas em vários servidores remotos de banco de dados (situados em locais distintos).

iii) Análise e apresentação das variáveis monitoradas, por métodos que executarão consultas aos servidores remotos de banco de dados, tratamentos matemático e estatístico, predição de valores futuros, visualização de variação temporal, etc.

iv) Controle do Processo Ambiental Monitorado, para viabilizar ações que evitem a continuidade de eventos catastróficos ao meio ambiente.

Arquitetura

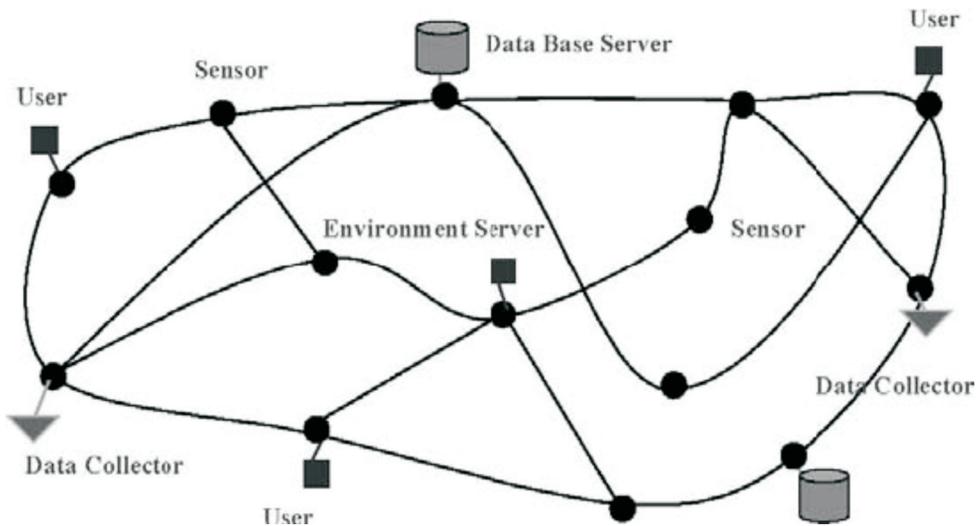


Figura 1 - Arquitetura de monitoramento ambiental

A figura 1 apresenta um esquema da arquitetura de monitoramento ambiental, que abrange sensores, coletores de dados, servidores ambientais, servidores de banco de dados e usuários, todos interligados pela Internet. Todos os sensores mostrados estão geograficamente distribuídos e podem ter um Protocolo de Internet (IP). Os sensores são os pontos de partida do processo de monitoramento; eles podem medir parâmetros ambientais na água, no ar e no solo e enviá-los a servidores remotos, para serem armazenados em bancos de dados distribuídos.

Os servidores de banco de dados armazenam todos os dados obtidos pelos sensores, isto é, as variáveis ambientais medidas e os parâmetros de referência mais comumente usados em monitoramento ambiental. Se o sensor possuir IP, as variáveis por ele medidas são remetidas aos servidores de banco de dados em tempo real, via Internet. Os coletores de dados ou “Data Collectors” constituem uma Interface de Programas Aplicativos (API), elaborada para armazenar no sistema de monitoramento ambiental parâmetros obtidos por sensores que não possuam IP. Os coletores de dados têm a mesma função dos sensores que possuem IP, com a diferença de os parâmetros ambientais medidos serem atualizados apenas quando o usuário acessar o sistema e não em tempo real.

O núcleo da arquitetura é um conjunto de classes e métodos, montado para criar uma estrutura que pode ser usada para todo o tipo de sistema de informação voltado ao monitoramento ambiental. Essa estrutura vem com uma Interface Gráfica do Usuário (GUI), que proporciona a comunicação entre o usuário e o sistema. Desta forma, o usuário pode informar ao sistema quais são as variáveis que lhe interessam medir, o tipo e as coordenadas geográficas dos diferentes sensores usados, e assim por diante. Além disso, o sistema tem um conjunto de métodos estatísticos e matemáticos para analisar os dados já armazenados nos bancos de dados. Há também métodos de visualização e apresentação, a fim de que os usuários possam coletar e ver os dados monitorados.

O modelo foi concebido para ser orientado ao objeto, à Internet e servir de suporte a sistemas de informação voltados ao monitoramento ambiental que rodem em qualquer sistema operacional existente. Para concretizar a independência de plataforma, sistemas baseados neste modelo deverão ser desenvolvidos em Java. Os programas desenvolvidos em Java são compilados em um código de *bytes* independente da arquitetura utilizada, que pode ser executado em qualquer plataforma que possua um interpretador para a linguagem. Assim, desde que a Máquina Virtual Java esteja instalada em seu computador, qualquer usuário poderá acessar o sistema de monitoramento ambiental, descarregando os respectivos *applets* por meio do Localizador Uniforme de Recursos (URL) do servidor ambiental. A linguagem Java também possui outras duas das três principais características desta arquitetura de monitoramento ambiental: é orientada ao objeto e à Internet.

Paralelamente ao desenvolvimento dos modelos mencionados, este artigo também propõe que sejam utilizados sistemas de aquisição de dados ambientais para a obtenção

dos parâmetros georreferidos que validarão os modelos desenvolvidos. As tecnologias de circuitos integrados para processamento e aquisição de dados, que surgiram nos últimos anos, possibilitam a construção de sistemas de captação de dados ambientais muito mais versáteis e econômicos. Os novos microcontroladores são compactos, baratos e demandam pouca energia, facilitando o desenvolvimento de sistemas de aquisição de dados automatizados, capazes de registrar variáveis ambientais de maneira prática e eficiente.

Modelo orientado ao objeto

O modelo orientado ao objeto compreende sete pacotes (cinco dos quais serão posteriormente detalhados), a seguir sintetizados:

i) Interface de Definição do Usuário, por meio do qual o usuário informa ao sistema quais são os parâmetros que ele deseja medir (nome do parâmetro, unidade de medida, intervalos de medida, data inicial, data final e regiões de interesse), os tipos e as coordenadas espaciais de sensores;

ii) Gerenciador de Banco de Dados, que alimenta os servidores de banco de dados com as variáveis ambientais coletadas pelos sensores, além de realizar as consultas conforme os parâmetros estabelecidos na Interface de Definição;

iii) Coletor de Dados, que proporciona uma interface para que o usuário informe ao sistema os dados manualmente coletados ou registrados;

iv) Servidor Ambiental, onde todos os *applets* podem ser baixados pelos usuários geograficamente dispersos na Internet;

v) Classes matemáticas e estatísticas, que contêm um conjunto de ferramentas que permitirão ao usuário analisar e coletar informações extras sobre o ambiente monitorado;

vi) Classes de visualização e apresentação, que fornecerão uma gama de funções elaboradas, a fim de possibilitar ao usuário a visualização de informações processadas e agregadas, relacionadas aos parâmetros ambientais armazenados nos servidores.

vii) Controlador de Processos Ambientais que, a partir dos parâmetros ambientais coletados pelos sensores e armazenados em bancos de dados, poderá desencadear um ou mais eventos concebidos a fim de impedir ou atenuar o andamento de processos nocivos ao meio ambiente.

Interface de definição do usuário

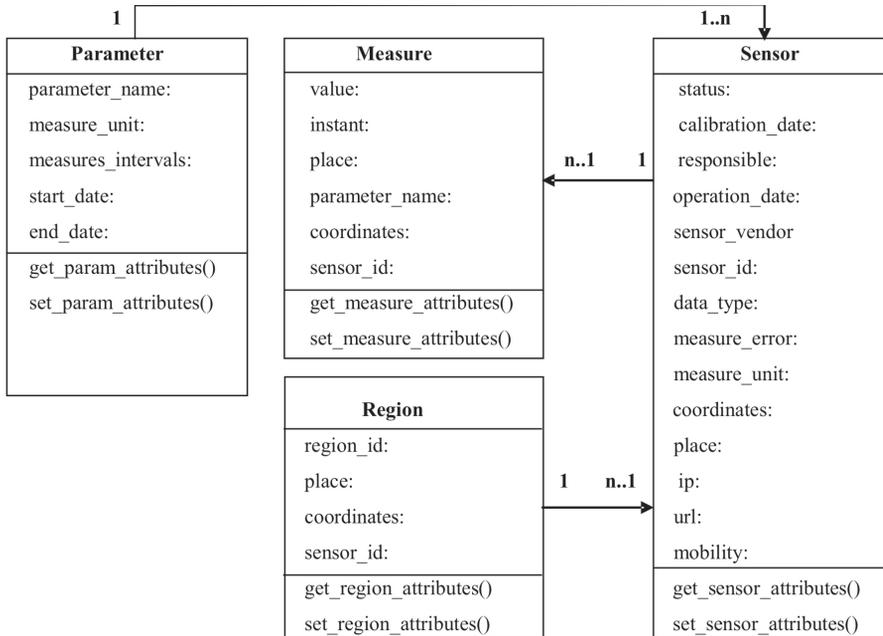


Figura 2 - Classes da interface de definição do usuário

O pacote da Interface de Definição do Usuário compreende quatro classes (figura 2) que devem fornecer, em sua instância, informações variáveis sobre sensores, medidas, regiões e parâmetros ambientais. Logo, essas classes serão nomeadas *Sensor*, *Region*, *Measure* e *Parameter*.

O conjunto de atributos da classe *Sensor* pode ser dividido em três categorias: Medição, Localização e Técnica. O monitoramento ambiental pode ser sintetizado pela ação de medir as grandezas físicas de determinados parâmetros, com certa precisão, num dado intervalo de tempo, durante um período requerido. Assim, a categoria Medição é responsável pela unidade de medição dos parâmetros ambientais e sua respectiva precisão através dos atributos *measure_error* e *measure_unit*. O atributo *data_type* conterá informações sobre a procedência dos parâmetros medidos (água, ar ou solo).

O segundo grupo de atributos da classe *Sensor* pertence à categoria Localização e compreende quatro membros: *mobility*, *coordinates*, *url* and *ip*. Sensores ambientais têm mobilidade variável, podem ser estáticos ou dinâmicos. Essa condição é especificada pelo atributo *mobility*. Todos os sensores são relacionados a coordenadas geográficas (latitude e longitude), ligadas ao atributo *coordinates*. Se o sensor for estático, suas coordenadas geográficas serão constantes. Por outro lado, se um sensor é definido como dinâmico, quando sua localização for modificada, suas coordenadas geográficas também serão trocadas. Os atributos *ip* e *url* representam o Protocolo de Internet (IP) e o Localizador Uniforme de Recursos (URL) do sensor. Mais quatro atributos da classe *Sensor* são

encontrados na categoria Técnica: *status*, *calibration_date*, *sensor_vendor*, *operation_date* e *responsible*.

Um sensor pode estar desativado, sob manutenção ou em operação, condições relacionadas ao atributo nomeado *status*. Cada sensor deve ter um responsável técnico e um fornecedor, sendo essas informações representadas pelos atributos *responsible* e *sensor_vendor*. O atributo que armazena a última data de calibração do sensor é designado *calibration_date*. Outro atributo, nomeado *operation_date*, especifica a data inicial de operação do sensor.

O parâmetro ambiental (grandeza física) a ser avaliado é identificado por um nome e uma unidade de medida. A periodicidade ou intervalo da medida pode ser horária, diária, mensal ou anual. Além disso, a medição do parâmetro tem datas que determinam o início e o fim do monitoramento. Essas características são especificadas na classe *Parameter*, respectivamente pelos atributos *parameter_name*, *measure_unit*, *measure_intervals*, *start_date* e *end_date*.

A classe *Region* foi criada para representar a região geográfica onde se deseja efetuar o monitoramento ambiental. É composta por quatro atributos: *sensor_id*, *place*, *coordinates* e *region_id*. O primeiro atributo identifica os sensores associados ao local de interesse, ao passo que o segundo e o terceiro guardam o nome e as coordenadas geográficas da região, respectivamente. O atributo *region_id* identifica a região escolhida para efetuar o monitoramento ambiental.

Cada medição efetuada tem características peculiares como valor da grandeza física, momento da aferição (agregando data e horário), nome do local onde foi efetuada, nome do parâmetro ambiental avaliado e coordenadas geográficas do local da medida. Os atributos *value*, *instant*, *place*, *parameter_name* e *coordinates*, da classe *Measure*, representam aquelas particularidades do processo de medição.

Um mesmo parâmetro ambiental poderá ser medido por um ou diversos sensores. Assim, o relacionamento entre as classes *Parameter* e *Sensor* é de um para muitos, representado pela notação '1..*'. Cada sensor ambiental executará uma ou várias medições. Logo, o relacionamento entre as classes *Sensor* e *Measure* é de um para muitos, representado pela notação '1..*'.

Classes de visualização e apresentação

As classes de exibição são relacionadas a tópicos como visualização de dados, coleção de dados, definição de dados e interface gráfica do usuário. Um diagrama de classes de visualização e apresentação é mostrado na figura 3. O esquema começa com uma classe nomeada *Screen*, que encerra os atributos *language* e *style_sheet*. O atributo *language* ajusta o idioma exibido na interface gráfica. O atributo *style_sheet* trará a URL da Folha de Estilos em Cascata (CSS), que modificará tópicos como estilo, fonte e cor

do caractere exibido na interface gráfica, além das cores da própria tela. A interface gráfica do usuário foi concebida para dividir a tela em muitas caixas. Cada caixa será responsável por mostrar ao usuário a descrição das funções da estrutura. Como exemplo, uma caixa poderia mostrar um gráfico sobre uma temperatura medida por um sensor, enquanto outra caixa exibiria um gráfico sobre a velocidade do vento medida por outro sensor. Assim, uma classe *Screen* pode instanciar uma ou muitas classes nomeadas *Box*.

A classe *Box* apresenta um relacionamento de associação recursiva, já que uma instância dela poderá ser dividida em uma ou várias classes *Box*, sendo seus atributos: *box_name*, *x_dimension*, *y_dimension*, *style_sheet*, *action*, *status*, *x_coordinate* e *y_coordinate*. Os três primeiros indicam o nome, a largura e a altura da caixa. O quarto atributo traz a URL da Folha de Estilos em Cascata (CSS), para ajustar o estilo, o tipo e a cor da fonte dos caracteres exibidos, além das cores de fundo e de frente da caixa. O atributo *action* pode assumir os valores “insert”, “edit” ou “delete”, informando que um objeto da classe *Box* está no modo de inserção, edição ou exclusão de dados.

Elaborado para informar o resultado das ações escolhidas para uma caixa, o atributo *status* pode assumir os valores “success”, “duplicated name”, “unfounded name”, “type mismatch” ou “unfilled attribute”. Os dois últimos atributos informam a abscissa e a ordenada do ponto da tela em que será desenhada a caixa especificada. Duas classes derivam da classe *Box*: *Query_Definitions* e *Visualization*.

A classe *Query_Definitions* define, após a consulta ao banco de dados, quais dos parâmetros ambientais recuperados, juntamente com os devidos índices ambientais calculados, serão exibidos por cada caixa de apresentação que o sistema criar. Portanto, seus atributos deverão especificar a lista de itens (parâmetros e índices) a serem mostrados, o intervalo temporal da consulta e o nome da agregação espacial dos sensores, respectivamente nomeados *items_list*, *time_interval* e *place*. Todas as variáveis monitoradas consultadas no servidor são armazenadas no atributo *parameters_list*.

Visando fornecer informações a respeito de todos os índices ambientais que o sistema pode calcular, elaborou-se o atributo *indexes_list*. É necessário conferir ao sistema a capacidade de criar índices ambientais a partir de equações fornecidas pelo usuário. Para tal fim, definiu-se o método *create_index()*, que efetuará uma análise léxica e sintática da formulação matemática do índice criado, para calculá-lo, caso seja possível. O propósito do atributo *place* é identificar a qual região espacial pertence a lista de parâmetros ambientais que o usuário escolheu para serem visualizados numa determinada caixa de apresentação. O atributo *box_name* tem a função de identificar a qual caixa serão aplicadas as definições de consulta estabelecidas em um objeto da classe *Query_Definitions*. Para informar os nomes de todas as caixas de visualização criadas, concebeu-se o atributo *box_name_list*. O atributo *action* especifica se o nome de uma caixa será incluído, alterado ou retirado do atributo *box_name_list*, assumindo os valores “insert”, “edit” ou “delete”.

A classe *Visualization* é composta dos atributos *itens_list*, *time_interval*, *place*, *visualization_type* e *box_name*. O atributo *visualization_type* determinará se a visualização requerida dos dados consultados será um gráfico, um mapa ou uma tabela, assumindo os valores “graphic”, “map” ou “table”. O atributo *itens_list* fornece parâmetros e índices ambientais, correspondentes à fonte dos dados para o processo de visualização, enquanto *time_interval* indica a taxa temporal para coletar a amostra de parâmetros ambientais. O atributo *box_name* tem a função de identificar em qual caixa se aplicarão as definições de visualização estabelecidas em um objeto da classe *Visualization*.

Como a visualização dos dados consultados será efetuada na forma de gráfico, mapa ou tabela, três classes podem ser derivadas da classe *Visualization*: *Table*, *Graphic* ou *Map*. A classe *Graphic* contém os atributos *x_coordinate*, *y_coordinate*, *x_interval*, *y_interval*, *label*, *x_label*, *y_label*, *graphic_type* e *visualization_type*. O atributo *label* se refere ao rótulo de todo o conjunto gráfico, enquanto os atributos *x_label* e *y_label* indicam o rótulo das ordenadas e abscissas. Analogamente, os atributos *x_coordinate*, *y_coordinate*, *x_interval* e *y_interval* fazem referência às coordenadas gráficas e seus respectivos intervalos sobre os eixos X e Y. O atributo *graphic_type* identifica o tipo de gráfico que será visualizado, assumindo os valores “linear”, “logarithmic”, “pizza” ou “bars”. O atributo *visualization_type* determinará de que forma os dados consultados serão visualizados (tabela, gráfico ou mapa). Cada gráfico pode mostrar uma ou muitas medidas temporais. Então, o relacionamento entre as classes *Graphic* e *Measure* (pertinente ao pacote da Interface de Definição do Usuário) será indicado pela notação um para muitos ou “1..*”.

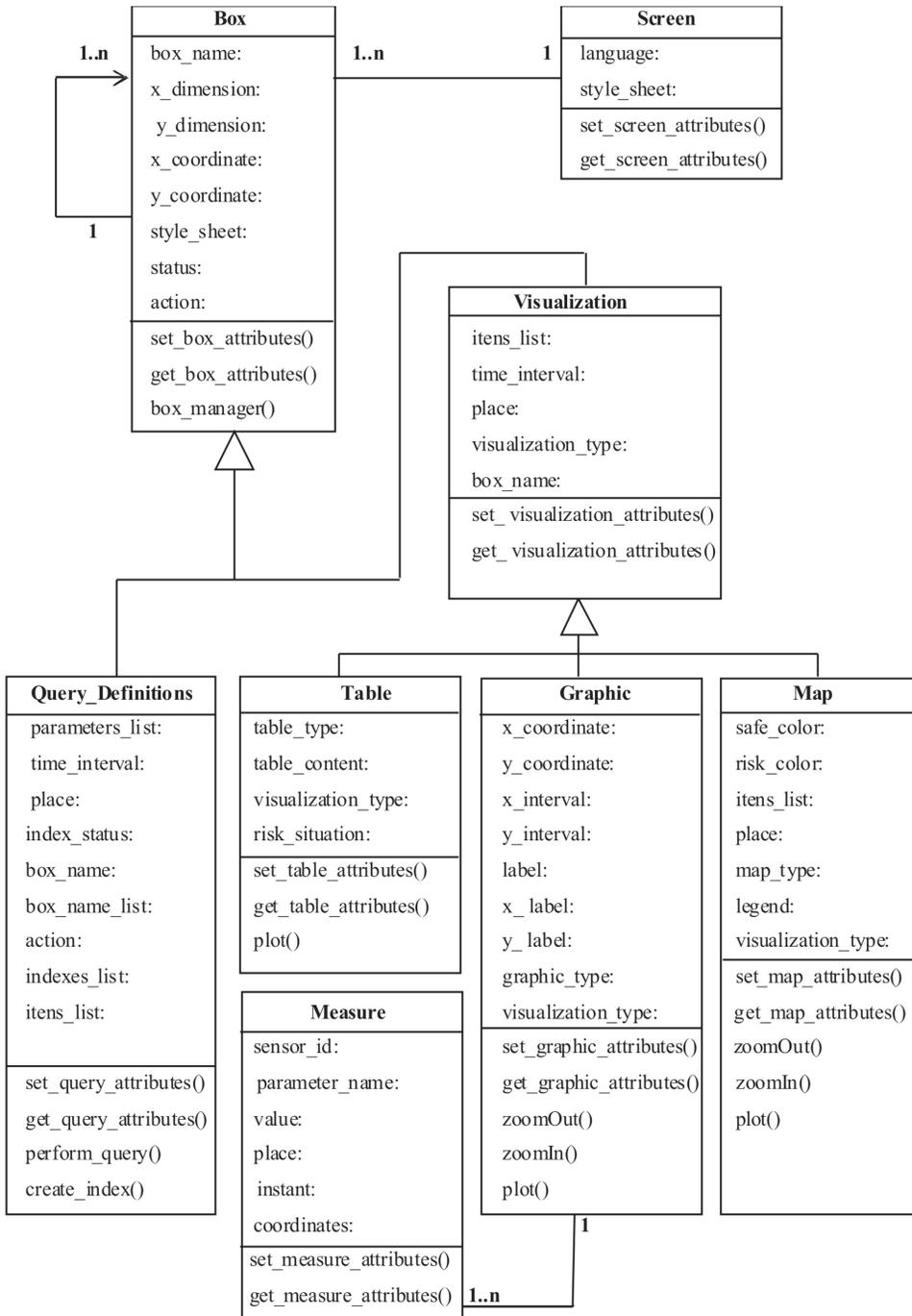


Figura 3 - Diagrama das classes de visualização e apresentação

A classe *Table* é responsável pela exibição numérica das medições coletadas, enquanto a classe *Mapa* traz os dados adquiridos (localização do sensor) dispostos conforme as regiões geográficas de sua ocorrência. Os métodos da classe *Table* serão

chamados apenas se o atributo *visualization_type* armazenar o valor “table”. Os atributos da classe *Table* são *visualization_type*, *table_type*, *table_content* e *risk_situation*. Podendo assumir os valores “data_sheet” ou “conventional”, o atributo *table_type* diz respeito aos tipos de tabela apresentados, que podem ser folha de dados ou convencional. O atributo *table_content* pode assumir um dos seguintes valores: “environmental report”, “measured parameters”, “searched parameters” ou “environmental indexes”. O atributo *risk_situation* pode armazenar os valores “true” ou “false”, para indicar se há risco real de dano ao meio ambiente.

A classe *Map* apresenta os atributos *safe_color*, *risk_color*, *location_color*, *legend*, *items_list*, *place*, *visualization_type* e *map_type*. Foi concebida para gerar um mapa com a localização de todos os sensores monitorados, numa mesma região e a lista de parâmetros escolhida pelo usuário. Os sensores cujos valores medidos se encontrem numa faixa segura serão exibidos numa coloração azul, caso contrário apresentarão cor vermelha.

Classes estatísticas e matemáticas

Para que o sistema forneça a funcionalidade desejada, deverá proporcionar ao usuário os métodos estatísticos e matemáticos adequados ao tratamento dos dados brutos, oriundos dos parâmetros ambientais coletados e ainda não processados.

O diagrama de classes estatísticas e matemáticas, exibido na figura 4, compreende as classes *Operation_Definition*, *Mathematical* e *Statistical*, além de uma classe externa, nomeada *Visualization*, oriunda das Classes de Visualização e Apresentação.

Cada instância da classe *Visualization* pode receber apenas uma sequência de tratamentos matemáticos e estatísticos por vez, especificada por meio do atributo *operations_list*. Assim, o relacionamento entre as classes *Visualization* e *Operation_Definition* terá multiplicidade de um para um, ou seja, “1..1”.

A superclasse *Operation_Definition*, da qual são derivadas as classes *Mathematical* e *Statistical*, é composta dos atributos *parameters_list*, *time_interval*, *start_date*, *end_date* e *operations_list*. Os quatro primeiros atributos fornecerão os subsídios para a consulta aos servidores que armazenam os dados ambientais coletados pelos sensores, indicando a lista de parâmetros (índices ou variáveis ambientais) escolhidos, o intervalo de coleta da amostra medida (pode ser horário, diário, semanal, mensal ou anual), a data inicial e a data final do monitoramento.

O sistema deve conferir ao usuário a capacidade de especificar métodos estatísticos e matemáticos que não estejam disponibilizados na arquitetura de monitoramento ambiental. A fim de alcançar esse objetivo, elaborou-se o método *create_operation()*, que realizará análise léxico-sintática em operações estatísticas e matemáticas fornecidas pelo usuário, para então calculá-las, caso seja possível.

Gerenciador do banco de dados

O pacote que abrange as classes gerenciadoras de banco de dados é composto de duas classes, conforme mostrado na figura 5. A primeira delas, nomeada *Database_Feeder*, promove a atualização dos parâmetros coletados pelos sensores, remetendo-os aos respectivos servidores ambientais. A outra classe, chamada de *Database_Query*, realiza consultas nos servidores ambientais baseadas nos parâmetros fornecidos pela interface de definição do usuário.

O atributo *url* armazena o endereço no servidor ambiental, enquanto *database_name* identifica o nome do banco de dados. O atributo *status* informa se houve sucesso na atualização ou consulta realizadas ao banco de dados.

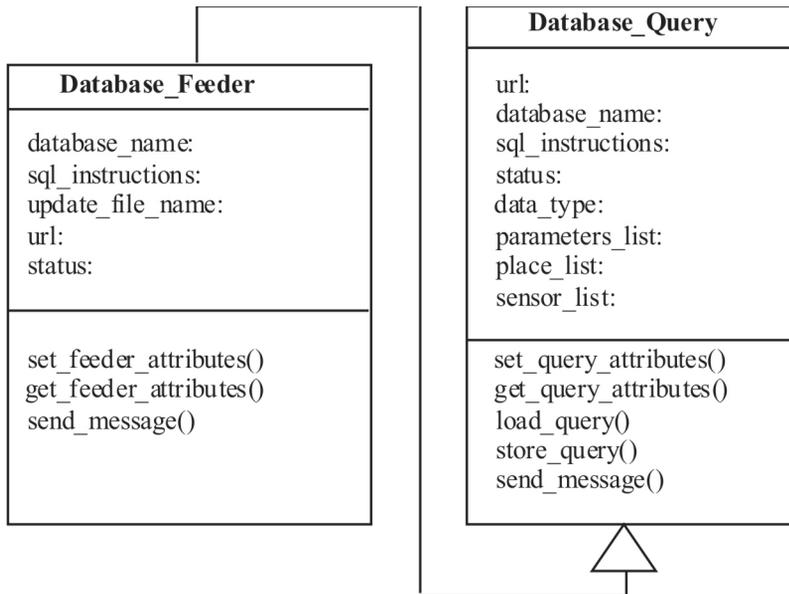


Figura 5 - Classes gerenciadoras do banco de dados

As operações de acessar e manipular o banco de dados são executadas por instruções elaboradas na Linguagem de Consulta Estruturada (SQL) e armazenadas num arquivo juntamente com o registro das informações a serem inseridas no banco de dados, e especificadas por todos os atributos das classes definidas na Interface de Definição do Usuário. Para fornecer os subsídios necessários à localização daquele arquivo com o nome registrado, a classe *Database_Feeder* usa o atributo *update_file_name*.

O conteúdo da informação a ser atualizada ou consultada (instruções em SQL e registro das informações a serem inseridas no banco de dados) está guardado no atributo *sql_instructions*.

Controlador de processos ambientais

Uma série de classes foi concebida com o propósito de controlar processos ambientais potencialmente danosos ao meio ambiente. Essas classes estão descritas no pacote Controlador de Processos Ambientais, uma delas é a classe externa *Table*, pertinente às classes de visualização e apresentação e a outra é a classe *Controller*, cujos atributos são *risk_id*, *risk_name*, *place*, *coordinates*. O atributo *risk_id* identificará numericamente a situação de risco, enquanto *risk_name* armazenará o tipo de risco ambiental que ocorre no local monitorado.

Dois dos primeiros métodos da classe *Controller* têm a função de recuperar e modificar o valor de seus atributos, respectivamente. O último método, identificado por *env_recover()*, desencadeará processos que restaurarão a integridade do ambiente monitorado. Pode-se ilustrar o mecanismo de funcionamento de um objeto da classe *Controller* para o caso de monitoramento de índice de risco de incêndio. O método *get_table_attributes (risk_situation)* informa se há uma situação de risco no local monitorado, de acordo com o valor armazenado no atributo *risk_situation*. Em caso afirmativo, o método *set_controller_attributes (risk_name)* guardará o nome do risco ambiental, enquanto o método *env_recover()* acionará remotamente um alerta para o corpo de bombeiros.

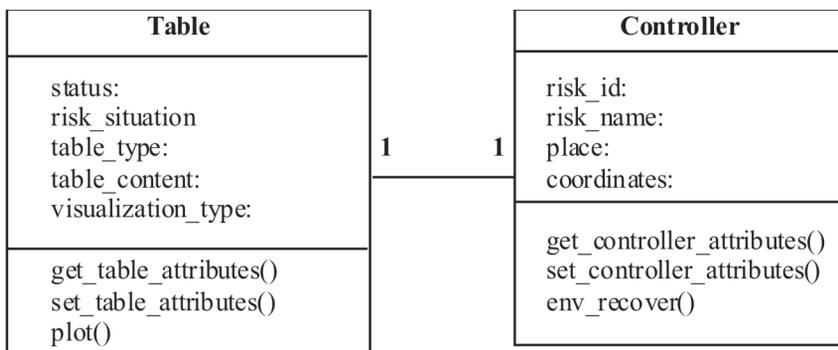


Figura 6 – Classes do pacote controlador processos ambientais

Conclusão

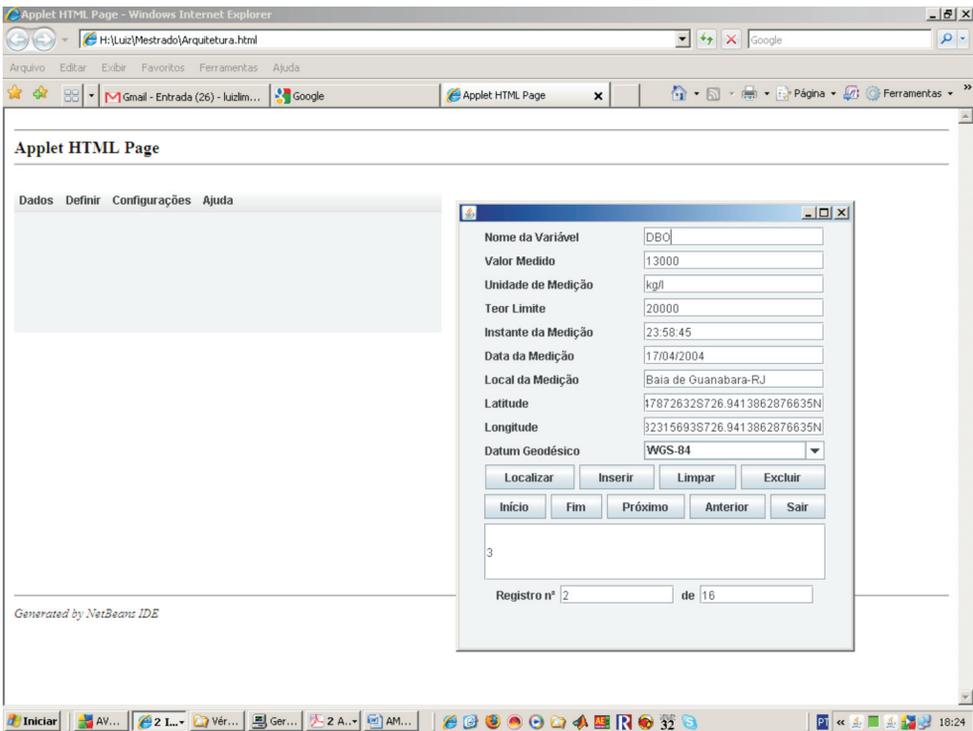
Observando as necessidades comuns dos sistemas de monitoramento ambiental, conclui-se que a utilização do modelo apresentado neste artigo pode beneficiar pesquisas, evitando repetição do desenvolvimento de programas ao utilizar o padrão aqui sugerido, além de dar espaço à criação de um trabalho cooperativo. O Sistema de Monitoramento Ambiental Global, patrocinado pelo Centro para Pesquisas Ambientais Globais (CGER), fornece alguns exemplos de sistemas de controle de processos de monitoramento

ambiental, que podem tirar proveito de uma implementação de software baseada na estrutura aqui apresentada. O trabalho Deitel & Deitel (2001), sugere que sistemas desenvolvidos em Java apresentariam a melhor relação custo-benefício na obtenção da independência de sistemas operacionais. Pensando nisso, criou-se uma aplicação baseada num protótipo em Java (cujas telas de consulta são mostradas na figura 7), que independe de plataforma e opera a base de dados relacionais, com o objetivo de testar a validade da estrutura proposta neste artigo.

O projeto do hardware necessário ao sistema de aquisição de dados ambientais está em andamento, no presente momento, para testar a arquitetura proposta na análise de modelos de qualidade de água na bacia de rios do norte fluminense. Esses modelos especificarão as equações matemáticas que governam o transporte de contaminantes na bacia, bem como a resolução numérica dessas equações, para determinar com precisão a variação temporal da concentração de poluentes em qualquer posição do corpo de água monitorado.

Como referência a trabalhos futuros, destaca-se que o autor deste artigo orienta um aluno do curso de engenharia de automação no desenvolvimento de mais dois protótipos, baseados nas linguagens de programação C++ e MATLAB.

A importância dessas ações reside no fato de que elas incrementarão as habilidades computacionais dos estudantes, além de inseri-los em temas de pesquisas inerentes à preservação ambiental.



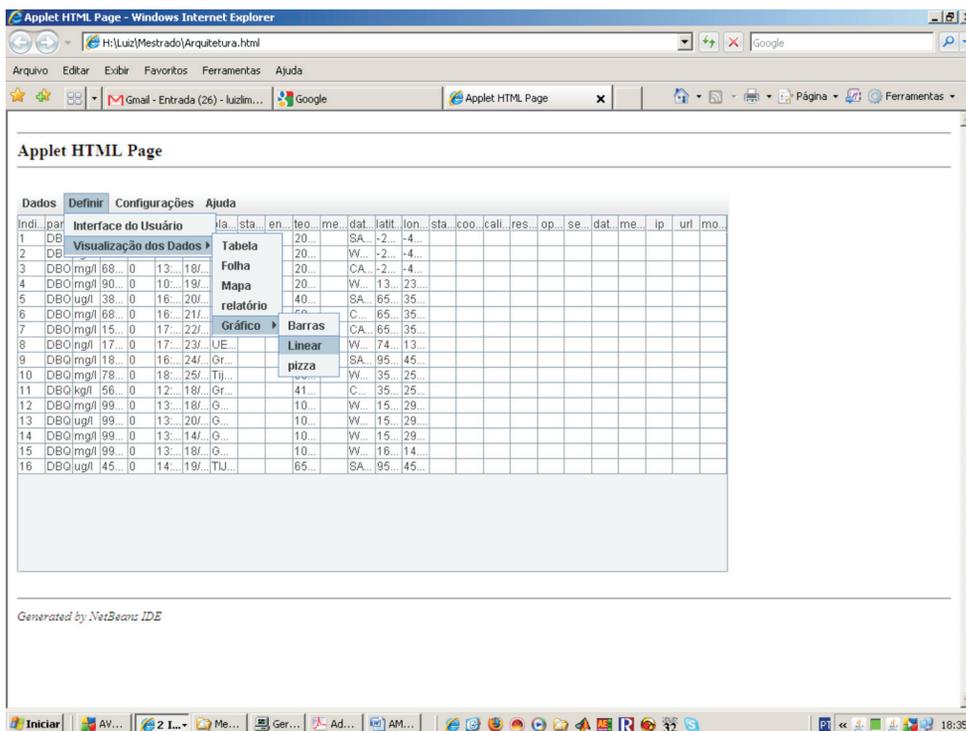


Figura 7 – Telas de consulta do aplicativo em Java baseado na arquitetura de monitoramento proposta

Referências

HIRANO, T.; HIRATA, R.; FUJINUMA, Y.; SAIGUSA, N.; YAMAMOTO, S.; HARAZONO, Y.; TAKADA, M.; INUKAI, K.; INOUE, G. CO₂ and Water Vapor Exchange of a Larch Forest in Northern Japan. *Tellus*, v. 55B, p. 244-257, 2003. Disponível em: <<http://www-cger2.nies.go.jp>>. Acesso em: 1 set. 2010.

PERIÁÑEZ, R. GISPART: a numerical model to simulate the dispersion of contaminants in the Strait of Gibraltar. *Science Direct – Environmental Software*, v. 20, n. 6, p. 797-802, dec. 2004.

FEDRA, K. *Model-based Decision Support for Integrated Urban Air Quality Management*. In: ADRIANO, D.C.; ISKANDAR, A.K.; MURARKA, I.P. (Eds.) *Contamination of Groundwaters. Advances in Environmental Science*. Northwood, UK: Science Reviews, 1994. p. 189-220. Disponível em: <<http://www.ess.co.at/docs/papers/fedra99.html>>. Acesso em: 9 jan. 2010.

UNNINAYAR, S. Climate System Monitoring. *Science Direct: The Science of the Total Environment*, v. 56, n. 2, p. 55-65, jun. 2003.

UNNINAYAR, S.; SCHIFFER, R.A. In-situ observations for the global observing systems: A compendium of requirements and systems. *NASA Office of Mission to Planet Earth*, v. 56, n. 2, p. 55-65, Jan. 1997. Disponível em: < <http://www.oco.noaa.gov>>. Acesso em: 13 jan. 2010.

JAWAD, S.; TOUMAMARK, W.; ELTGROTH, K.; PAIKE, D. C. Expert interface for modeling air quality impacts from superfund sites. *Science Direct: Environmental Software*, v. 10, n. 4, p. 223-239, 24 jul. 2000.

LOPES, A.M.G. WINDSTATION: A software for the simulation of atmospheric flows over complex topography. *Science Direct: Environmental Software*, v. 18, n. 4, p. 81-96, mar. 2003.

ONGLEY, E.D. Matching water quality programs to management needs in developing countries: the challenge of program modernization. *Science Direct: European Water Pollution Control*, v. 7, n. 4, p. 43-48, set. 1996.

OBJECT MANAGEMENT GROUP (OMG). *Unified Modeling Language 2.0 Infrastructure Specification*. Version 2.0 – formal/03-09-15, abr. 2004. Disponível em: < <http://www.omg.org/docs/ptc/03-09-15.pdf>> Acesso em: 11 jan. 2010.

SANTOS, I. dos; BRAGA, Sérgio M.; FERNANDES, C. V. S. Monitoramento Automático de Qualidade da Água: uma visão crítica para a Bacia do Rio Barigüi. In: SIMPÓSIO BRASILEIRO DE RECURSOS HÍDRICOS, 15., 2003, Curitiba. Disponível em: < http://www.lactec.org.br/publicacoes/2003/063_2003.pdf> Acesso em: 15 set. 2009.

Artigo recebido em: 04 abr. 2010

Aceito em: 16 ago. 2010