

2 Interoperabilidade e GIS

2.1. Motivação

O uso em larga escala da World Wide Web (WWW ou Web, simplesmente) já é uma realidade em grande parte do mundo e é crescente a quantidade de serviços disponibilizados nela. Alguns destes serviços estão relacionados a informações geográficas, deste simples endereços até sistemas de traçado de rotas e visualização de mapas [1].

Um grande problema destes serviços é que cada um possui sua própria forma de lidar com as informações e disponibilizá-las, o que dificulta a interoperabilidade entre serviços, incluindo-se aqui as fontes de informação.

Sistemas de informação geográfica abertos (*Open geographic information systems*) representam a evolução dos SIGs tradicionais, nos quais funções monolíticas de software e modelos de dados tradicionais são implementados de forma a facilitar interoperabilidade e estendibilidade [2].

A arquitetura usada para proporcionar a interoperabilidade entre SIGs é basicamente na forma de cliente/servidor, onde o sistema principal, o que se quer disponibilizar, é o servidor e os clientes são os outros sistemas que irão interagir com este servidor [16].

Assim, para se atingir interoperabilidade, é necessário que se implemente este cliente, de forma autônoma ou estendendo outros softwares, na forma de *plugins*, por exemplo.

Há dois tipos básicos de clientes, denominados *clientes finos* e *clientes gordos* [16].

Por fim, as arquiteturas tecnológicas utilizadas para interconectar estes sistemas são basicamente a ponto-a-ponto (*peer-to-peer*), Web Services e CORBA.

2.2. Exemplos de Projetos

2.2.1. OpenGIS

Conforme adiantado no Capítulo 1, o Open GIS Consortium (OGC ou OpenGIS) é um consórcio que visa proporcionar formas de se construir SIGs abertos e interoperáveis, abrangendo alguns conceitos de informações geo-espaciais, como elementos (*feature*) simples ou complexos, mapas e superfícies (*coverages*) e dados relacionais [2].

O OpenGIS possui três preocupações básicas, que visam auxiliar a interoperabilidade e extensão das aplicações de SIG:

- Ambiente de aplicação interoperável: uma interface configurável que utiliza ferramentas específicas e dados necessários para resolver um problema.
- Espaço de dados compartilhados: um modelo de dados genérico que suporta uma grande variedade de aplicações analíticas e cartográficas.
- Navegação heterogênea às informações: métodos para acessar e explorar as informações e aplicações analíticas disponíveis na rede.

Estes três itens estão interligados em uma arquitetura global de sistema. Embora cada um possua um conjunto distinto de características, é possível compor um *framework* comum definindo como cada componente atua. O OpenGIS possui um projeto, o *Open GeoData Interoperability Specification* (OGIS), que disponibiliza um *framework* arquitetural, orientado a objetos, para acesso e geo-informações, independentes da estrutura de dados e dos formatos de dados usado no modelo de dados.

Do ponto de vista do usuário, o OGIS proporciona acesso às informações de locais remotos sem se importar com o formato destes dados. Do ponto de vista do desenvolvedor, o OGIS provê um conjunto de serviços na rede, para identificar, interpretar e representar um conjunto de dados de um servidor de informações geo-espaciais para um cliente. A Figura 1 mostra a arquitetura onde as interfaces OGIS atuam.

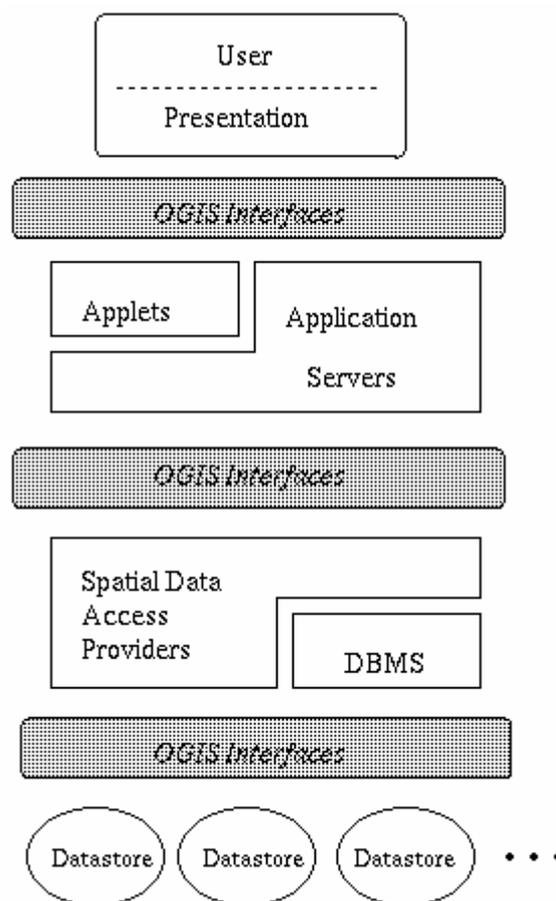


Figura 1 - Interfaces OpenGIS [2]

2.2.2. Projeto Inspire

INSPIRE (*IN*frastructure for *S*patial *I*nfoRmation in *E*urope) é uma iniciativa da Comissão Europeia, desenvolvida em colaboração com estados

membros e países em ascensão. Ela atua disponibilizando informações geográficas relevantes, de forma padronizada e de qualidade que suportam formulação, implementação, monitoramento e avaliação por comunidades regulamentadoras [9].

Os princípios comuns que orientam o projeto INSPIRE são [8]:

- Dados devem ser inicialmente coletados e depois mantidos em um nível de detalhe que garanta acesso eficiente.
- Deve ser possível combinar, sem restrições, informações espaciais a partir de diferentes fontes, localizadas na Europa, e compartilhá-las entre diferentes usuários e aplicações.
- Deve ser possível, para informação coletada em um nível, compartilhá-la em todos os diferentes níveis.
- Informações geográficas necessárias para um bom governo, em todos os níveis, devem ser abundantes.
- Deve ser fácil descobrir quais informações geográficas estão disponíveis, incluindo as exigências específicas de uso, e sobre quais condições elas podem ser adquiridas e usadas.

Informações geográficas devem ser fáceis de entender e interpretar.

A visão do INSPIRE é representada na Figura 2

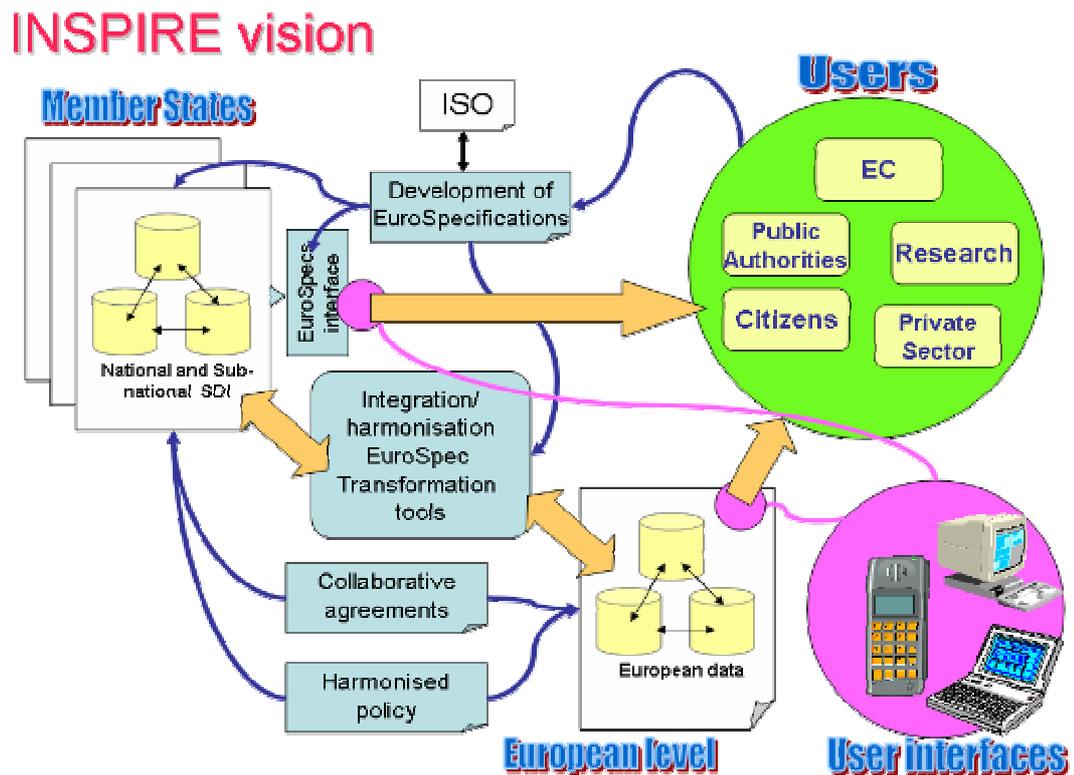


Figura 2 - Visão do INSPIRE [8]

2.3. Exemplos de Tecnologias

2.3.1. Implementações baseadas na Web

Muitas das fontes de informações dispostas na internet são relacionadas a comunidades de regiões específicas, por exemplo, Web Sites contendo informações sobre restaurantes, teatros, filmes e lojas focadas em clientes que usam a Internet para consumir produtos em sua própria região. Conseqüentemente, estes web sites possuem informações que podem ser indiretamente georreferenciadas como endereço, cep, nome de lugar e outras descrições textuais. Estas informações não possuem coordenadas (x, y) associadas diretamente, mas pode ser convertida em dados georreferenciados através de funções de casamentos de endereço, assim a Web pode ser vista como uma grande fonte de dados geoespaciais[1].

Como dito anteriormente, existem basicamente dois tipos de clientes: gordos e finos.

Mais precisamente, *clientes finos* são aqueles mais simples pois concentram todo o processamento dos dados no lado servidor. São rápidos de implementar e fáceis de se disponibilizar. Os clientes finos podem ser implementados como Web Services. Por exemplo, o Web Map Service [10] usa o protocolo HTTP e permite implementar clientes de forma simples, inclusive usando linguagens de scripts, como JavaScript.

Clientes gordos são aqueles com maior capacidade de processamento, dividindo com o servidor o processamento das operações. Naturalmente, sua implementação exige maior esforço. Por exemplo, no Web Feature Service [21], o servidor disponibiliza dados sem um pré-processamento para o cliente. O cliente oferece serviços de visualização dos dados e de consulta espacial e convencional.

O volume de dados trafegado entre o cliente e o servidor no cliente gordo é muito maior que no cliente fino. Porém, clientes gordos fazem muito menos acessos ao servidor dos clientes finos.

2.3.2. Implementações baseadas em CORBA

SIGs são compostos por diferentes componentes especializados que raramente são designados com objetivos de cooperação e integração. Estes componentes são especializadas bibliotecas ou sub-sistemas que foram previamente desenvolvidos e utilizados em diferentes contextos. CORBA provê suporte a flexibilidade e estendibilidade para definir interfaces para estes componentes [14].

O projeto TerraFly [15] fornece um exemplo interessante de sistema para tratamento de dados geográficos baseado em CORBA.

A linguagem de definição de interfaces de CORBA, a IDL, está ligada ao conceito de linguagem orientada a objetos, mas não exclui a integração com linguagens de outros paradigmas, como COBOL, LISP e TCL. Para se integrar a um *framework* definido em CORBA, a aplicação precisa definir a interface em CORBA com a qual ela deseja interagir. Esta interface é devidamente compilada para que interaja com o *framework* definido em CORBA, independente da linguagem em que ele está implementado.

A habilidade de um sistema de descobrir e usar novos componentes em tempo de execução, à medida que eles ficam disponíveis, é chamada de estendibilidade dinâmica do sistema. Esta é uma característica importante para sistemas distribuídos, uma vez que eles introduzem um alto grau de flexibilidade, permitindo que cada componente use outro componente sem a necessidade de uma compilação prévia, ou seja, permite uma amarração tardia.

Este tipo de flexibilidade em sistemas distribuídos demanda uma sofisticada infra-estrutura que suporta a descoberta automática de componentes, sua amarração em tempo de execução e a chamada dinâmica às operações definidas em suas interfaces. CORBA suporta todas estas características.

Serviços de CORBA, como *Naming*, *Event* e *Trader*, oferecem funcionalidades para que o componente descubra outros componentes pelo nome ou pela propriedade dos atributos. A *Interface Repository* (IR) de CORBA armazena todas as interfaces dos objetos que fazem parte do sistema distribuído. A *Dynamic Invocation Interface* (DII), no lado cliente, e a *Dynamic Skeleton Interface* (DSI), no lado servidor, fornecem as funcionalidades necessárias para gerar requisições dinâmicas, ou seja, em tempo de execução [14].

Para viabilizar interoperabilidade através de CORBA em SIGs, o OpenGIS propôs um modelo de dados completo, orientado a objetos, onde é possível acessar informações geográficas e realizar consultas espaciais, temporais e convencionais [12].

Em [13], foi proposta uma forma de utilizar o modelo em CORBA do OpenGIS, mas acrescentando algumas extensões. A solução proposta é implementada em Java, usando Enterprise Java Beans e sistemas de gerenciamento de banco de dados.

Pode-se destacar alguns benefícios importantes proporcionados por SIGs baseados em CORBA [14]:

- Formato de representação de dados mensuráveis
- Visualização de geo-informações e suporte a interação com o usuário
- Representação de mapas e suporte a consultas
- Funções de modelagem de dados complexas

Toda aplicação de SIG processa os dados geográficos em algum formato. A representação padronizada de um formato proporciona flexibilidade de interoperabilidade da aplicação. Para “dados mensuráveis” ,como temperatura, densidade ou altitude, estes formatos devem tender para uma representação canônica de valores mensuráveis, conversão entre diferentes unidades e estendibilidade. Para mapas, outro tipo de dados geográficos muito comum, devem ser especificados formatos de representação padronizados, operações de consultas espaciais e visualização. Sobreposição de mapas e transformações também devem ser suportadas. Funções de modelagem complexas também podem ser definidas em sistemas de SIG baseados em CORBA.

2.4. SIGs de Código Aberto

O desenvolvimento de softwares de código aberto tem recebido uma grande atenção recentemente. De fato, acompanhando projetos de sucesso como o Linux, Apache e Perl, pesquisadores e profissionais de Informática vem se interessando pela produção de softwares de código aberto [3]. Um tópico de interesse particular é a adesão a este tipo de software por órgãos governamentais, com a intenção de reduzir custos com licenças e promovendo desenvolvimento tecnológico local

através do acesso ao código-fonte destes sistemas. Uma pesquisa recente de direito de propriedade intelectual e desenvolvimento internacional do Reino Unido dá a seguinte recomendação: “Países em desenvolvimento e os seus parceiros deveriam revisar a opção de políticas para obtenção de softwares, com uma visão de usar softwares de código-fonte aberto, visando reduzir custos [4]”.

O mundo de software aberto visa proporcionar à comunidade uma série de modelo de padrões para criar e modificar seus produtos. Este modelo de desenvolvimento cresce e se torna popular quando o estabelecimento destes padrões consolida uma tecnologia e permite compatibilidade de soluções de diferentes empresas que competem no mercado, abrindo assim a oportunidade para produtos de código aberto. Muitos usuários podem se beneficiar do esforço intelectual de se estabelecer um padrão. Um exemplo é a linguagem padrão de consulta a banco de dados SQL.

Câmara e Onsrud [5] apresentam uma análise dos desenvolvedores de software abertos de SIG, baseada em uma pesquisa sobre 70 projetos de SIG de código aberto, resumida em parte a seguir.

De acordo com o tamanho, distribuição geográfica e filiação, as equipes de desenvolvimento de SIGs de código aberto podem ser divididas em 3 categorias:

- Projetos individuais: a equipe consiste de 1 a 3 membros geralmente do mesmo local e trabalhando durante seu tempo livre. Os softwares produzidos são aplicações especializadas de tamanho pequeno que possuem restrições específicas. Em geral, o desenvolvedor do software é também o seu primeiro usuário. Como exemplos podemos citar a ferramenta de visualização Vis5D [6].
- Redes colaborativas: o núcleo desta equipe de projeto consiste de 15 a 30 membros geograficamente distribuídos. Os desenvolvedores trabalham separadamente, durante o seu tempo livre, ou alocados ao projeto em meio-período, a partir de um acordo com o seu empregador. Como exemplo, existe a ferramenta de análise espacial GRASS.

- Baseadas em corporações: o núcleo da equipe é parte de uma instituição e consiste de 3 a 8 programadores. Podem existir colaboradores externos, mas a arquitetura principal é feita na instituição e, em alguns casos, pode ter fins comerciais de acordo com os objetivos da corporação. Como exemplo, temos o PostGIS, uma extensão do PostgreSQL,

	Total	Baseado em tecnologia madura	Baseado em Padrões	Baseado em Inovações
Projetos individuais	37 (53%)	12	19	6
Redes colaborativas	4(6%)	1	1	2
Baseadas em corporações	29 (41%)	6	18	5
	70	19 (27%)	38 (54%)	13 (19%)

Tabela 1 – Caracterização e propriedade intelectual de SIGs de código aberto [5].

A proporção relativamente pequena de projetos de inovação (19%) indica que o esforço de desenvolvimento da maioria dos softwares de SIG de código aberto baseia-se nos modelos de produção calcados em padrões (ou em tecnologias maduras). Não são motivados diretamente pela inovação que representam, mas na diminuição do gasto com licenças e da diminuição dos monopólios.

A grande presença de produtos baseados em padrões é reflexo da influência do consórcio OpenGIS. Este resultado deixa a noção de que a parte mais difícil do desenvolvimento do software é a arquitetura conceitual do produto [7]. Os dois projetos baseados em inovação são o GRASS e o R. Ambos têm uma arquitetura simples e bem definida. A sua contribuição, quanto à inovação, consiste das funções de análise que os cientistas desenvolvem usando estes ambientes [5].