Universidade do Porto
Faculdade de Engenharia
FEUP

Nuno Miguel Belo Oliveira Ferreira

Desenvolvimento de um Módulo de GIS para Integração na Aplicação SPOTS na Siemens S.A.

3) LEIC
04/FERn

LEIC
Outubro, 2004

Faculdade de Engenharia da Universidade do Porto

Licenciatura em Engenharia Informática e Computação

Desenvolvimento de um Módulo de GIS para Integração na Aplicação SPOTS na SIEMENS S.A

Relatório do Estágio Curricular da LEIC 2003/04

Nuno Miguel Belo de Oliveira Ferreira

Orientador na FEUP: Prof. A. Augusto de Sousa

Orientador na SIEMENS S.A.: Engº Manuel Jorge Rodrigues

Setembro de 2004

Universidade do Porto
Faculdade de Engenharia
Biblioteca
Nº
CDU
Data / /200

004/0473) LEIC/EIC5202 2004/FERn
Universidade do Porto
Faculdade de Engenharia
Biblioteca
Nº 81525
CDU
Data 21/03/2006

Aos meu pais que tudo fazem por mim.

Resumo

Para obter uma melhor compreensão relativamente à motivação e ao enquadramento do projecto de estágio torna-se necessário fazer uma breve descrição de uma das principais aplicações da Siemens – o SPOTS.

O SPOTS é uma aplicação de monitorização e análise de desempenho de redes de telecomunicações usada à escala global. A grande utilidade do SPOTS reside essencialmente em duas das suas funcionalidades:

- ***Reporting:*** as funcionalidades de *reporting* englobam essencialmente serviços que permitem a criação de relatórios especificos de telecomunicações; estes relatórios são actualizados em tempo real, permitindo assim avaliar e analisar constantemente o estado da rede.
- ***Alarming:*** as funcionalidades de *alarming* referem-se essencialmente a um serviço de alarmes em tempo real que alerta o utilizador sempre que se verifica a existência de uma anomalia na rede.

Apesar da grande utilidade da aplicação, esta não permite que o utilizador tenha uma percepção visual localizada de tudo o que está a acontecer. Por este motivo surgiu a ideia de introduzir no SPOTS um módulo de informação geográfica que permitiria ao utilizador, através de mapas, ter uma percepção visual do estado da rede e da localização dos seus componentes. Para além disto, o acesso à informação específica da rede seria facilitada uma vez que os mapas iriam suportar grande parte do processo interactivo.

O trabalho do projecto desenvolveu-se em três fases. Na primeira fase foi feita uma análise da aplicação SPOTS e das tecnologias envolvidas. Na segunda fase foi feita uma análise do estado da arte dos sistemas de informação geográfica e, finalmente, passou-se à fase de projecto de alto nível.

Pretendia-se uma aplicação com um leque reduzido de funcionalidades típicas de GIS, para reduzir o tempo de implementação e, ao mesmo tempo, algo capaz de revolucionar um pouco o próprio conceito do SPOTS, que não tinha ainda forma de fazer uma apresentação visual dos componentes de rede. Para isso foram definidos alguns requisitos que contemplam a criação de mapas, adição de *layers* e dispositivos, algumas operações do foro gráfico e operações específicas do SPOTS. Tendo, novamente, o objectivo de uma implementação rápida em conta, optou-se por usar uma abordagem simplificada da arquitectura integrada dos sistemas GIS.

Uma vez definida a arquitectura e identificados os componentes passou-se então para a fase de implementação e testes.

Neste documento descreve-se o projecto, os seus objectivos e o trabalho realizado. Apresentam-se algumas tecnologias envolvidas e descrevem-se todas as fases de desenvolvimento. No final retiram-se as conclusões relativamente ao projecto na sua globalidade. Não se pretendem detalhar todos os acontecimentos relacionados com o estágio, o objectivo é transmitir essencialmente os aspectos mais importantes do trabalho realizado.

Agradecimentos

Apresento os meus especiais agradecimentos ao Eng.º Manuel Jorge Rodrigues, o meu orientador na empresa, pelos conselhos e apoio dados que foram determinantes para a realização do projecto e para a minha valorização como futuro profissional.

Agradeço a todos os professores que me leccionaram durante a licenciatura, pela transmissão de conhecimentos e hábitos de trabalho que em muito contribuíram durante a realização deste projecto. Um agradecimento especial aos professores Raul Moreira Vidal e A. Augusto de Sousa, o meu orientador da faculdade, pelo acompanhamento prestado e pela disponibilidade demonstrada ao longo da minha vida académica.

Um agradecimento muito especial aos meus pais que sempre estiveram do meu lado para me levantar quando as forças pareciam desaparecer.

Uma palavra de apreço aos meus companheiros de curso e amigos com quem vivi momentos inesquecíveis.

Finalmente, gostaria também de agradecer ao Programa PRODEP que subsidiou o estágio curricular.

Índice de Conteúdos

Lista de Figuras

Lista de Tabelas

1 Introdução

Neste capítulo apresenta-se a empresa onde foi realizado o estágio, os objectivos do estágio, o conteúdo do projecto e é feita uma breve descrição da estrutura do documento.

1.1 Apresentação da Instituição de Estágio SIEMENS S.A.

Nesta secção apresenta-se a empresa onde se realizou o estágio – a Siemens. Inicialmente é feita uma breve apresentação da história da empresa, passando-se depois para uma contextualização da realidade da Siemens em Portugal. Finalmente, é feita uma descrição global da estrutura da empresa e das principais áreas de negócio.

História da Empresa

A empresa que deu origem à Siemens, a Telegraphen-Bau-Anstalt von Siemens and Halske, foi fundada em Berlim no ano de 1847, por Werner von Siemens, o seu primo Johann Georg Siemens e Johann Georg Halske. O objectivo era construir instalações telegráficas e outro equipamento eléctrico.

Em pouco tempo, a empresa começou a espalhar linhas de telégrafo por toda a Alemanha. Em 1855 estabeleceu uma secção em St. Petersburg para as linhas russas e em 1858 uma secção em Londres para as linhas inglesas. Com a empresa a crescer e depois de começar a produção em massa, Halske, que estava menos inclinado para a expansão, retirou-se, em 1867, deixando o controlo da empresa aos irmãos Siemens e seus descendentes.

Em 1890 a Siemens tornou-se uma sociedade de responsabilidade limitada, sendo Carl Siemens (irmão de Werner), Arnold Siemens e Wilhelm Siemens (filhos de Werner) os sócios. Mais tarde, em 1897, tornou-se uma sociedade de responsabilidade ilimitada, a Siemens and Halske AG.

Em 1903 a Siemens and Halske transferiu a sua actividade de engenharia-potência para uma nova empresa, a Siemens-Schuckertwerke (absorvendo a Nürnberg, Schuckert and Co.). A partir de 1919, as duas empresas eram presididas pelo mesmo dirigente, que era sempre um membro da família Siemens. Em 1932, e depois de sete anos de colaboração, a Reiniger Gebbert and Schall fundiu-se com a Siemens originando a Siemens-Reiniger-Werke AG, concentrando-se na produção de equipamento médico e terapêutico, especialmente máquinas de raios X e microscópios eléctricos.

As empresas da Siemens conheceram uma grande expansão durante o terceiro Reich (1933-45). Todas as fábricas trabalhavam na sua capacidade máxima e estavam dispersas pelo país para evitar possíveis danos causados por ataques aéreos. No final da guerra, uma grande percentagem de fábricas e equipamento na zona ocupada pelos soviéticos foi expropriada. Durante o período da guerra-fria houve interesse soviético na reconstrução económica da Alemanha de Leste, e a Siemens expandiu-se gradualmente entrando no mercado eléctrico europeu. Assim, no ano de 1960, a Siemens era novamente uma das maiores empresas do ramo a nível mundial.

Em 1966, todas as companhias relacionadas com a Siemens se fundiram, originando uma nova empresa, a Siemens AG.Desde então até aos dias de hoje, a empresa conheceu uma enorme expansão e, actualmente, com 500 centros de produção em 50 países e presente em

190 países, a Siemens está representada em todo o mundo. Conta com mais de 400 mil colaboradores, 60% dos quais trabalha fora da Alemanha.

A SIEMENS em Portugal

Em Portugal, a Siemens está sediada em Alfragide e conta com uma delegação em Perafita na zona do Porto –Figura 1. Existem ainda as fábricas de Sabugo (transformadores), Corroios (quadros eléctricos), Seixal (a Indelma, que produz cablagens para a indústria automóvel) e Vila do Conde (a Infineon, que produz memórias dinâmicas). Actualmente a empresa conta com mais de 5 mil colaboradores neste país.

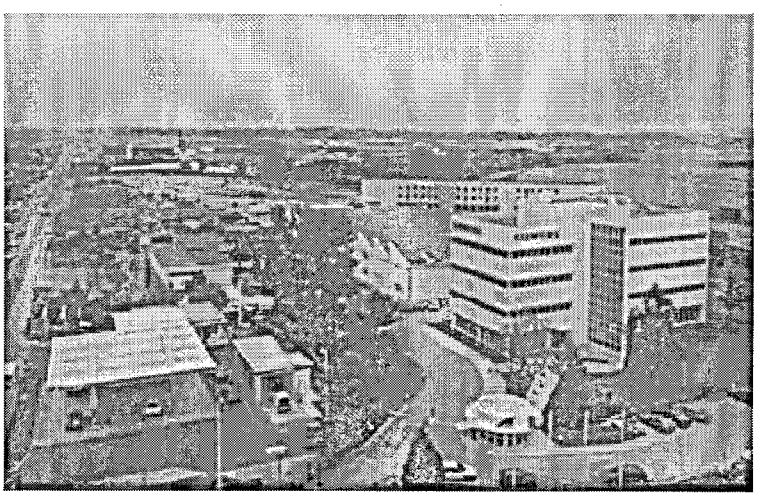

Figura 1: Instalações da delegação norte da Siemens

Estrutura da Empresa

A Siemens está organizada em diversas unidades de negócio, a saber:

- ***Information and Communications:*** aposta numa estratégia que assenta no desenvolvimento de três domínios, nomeadamente protocolo internet (IP), mobilidade e comércio electrónico, nos quais pretende alcançar uma posição de liderança, num mercado que se caracteriza pela globalização, liberalização, convergência, dinamismo e extrema complexidade.
- ***Transportation Systems:*** abrange a construção de veículos, sistemas de sinalização e segurança, a electrificação e alimentação de vias ferroviárias, sistemas de telecomunicações para informações de clientes e também assistência e manutenção.
- ***Medical Solutions:*** reside na compreensão dos processos ao nível dos consultórios médicos e hospitais, na identificação dos problemas e na concretização de soluções globais para lhes fazer face.
- ***Power Generation:*** tem por objectivo oferecer as melhores soluções possíveis aos clientes cada vez mais expostos à pressão da concorrência.
- ***Automation and Control:*** aposta no desenvolvimento de soluções de controlo e recepção dos materiais, passando pelo processo de fabrico, até à expedição dos produtos acabados; oferece aos clientes a possibilidade de automatizar toda a sua cadeia de produção sem rupturas.

Torna-se importante salientar que o projecto de estágio foi desenvolvido na área IC *(Information and Communications)*, mais concretamente no sector WON *(Wired and Optical Networks)*.

1.2 O Projecto SPOTS GIS na Instituição de Estágio SIEMENS

O estágio foi integrado num projecto que vem sendo desenvolvido e mantido pela Siemens há já alguns anos – o SPOTS.

O SPOTS é uma aplicação de monitorização e análise de desempenho de redes usada à escala global. Em linhas muito gerais, a aplicação permite ao utilizador manter, em tempo real, a percepção do estado dos diversos componentes de uma rede, actualizando constantemente relatórios concebidos para esse fim. O sistema disponibiliza ainda, também em tempo real, e para além de muitas outras funcionalidades, um mecanismo de alarmes que visa alertar para a ocorrência de uma eventual anomalia na rede.

Com este estágio na área de Sistemas de Informação Geográfica *(GIS)*, pretende-se analisar o estado da arte da tecnologia e os sistemas deste tipo existentes no mercado e ainda implementar um protótipo de um módulo de GIS passível de ser integrado no SPOTS. O objectivo desta integração é possibilitar a associação entre informação geográfica e informação específica das redes de telecomunicações, tornando assim mais intuitiva a percepção visual do estado da rede e a localização dos seus componentes para o utilizador.

No entanto, é importante salientar que o módulo GIS a implementar não visa contemplar todas as funcionalidades típicas de um *software* deste tipo. Pretende-se um protótipo com funcionalidades de GIS mínimas que permita avaliar a possibilidade e facilidade de integração de uma extensão deste tipo em versões futuras do SPOTS.

Outro objectivo consiste em compreender e aplicar o processo de desenvolvimento de *software* praticado na empresa, sendo o estagiário responsável por todos os passos, a saber: análise de requisitos, especificação, desenho, implementação e testes.

1.3 Organização e Temas Abordados no Presente Relatório

O presente capítulo, o primeiro, serve de introdução ao projecto de estágio realizado, apresentando a empresa onde foi realizado, identificando os objectivos e as motivações que estiveram na origem do mesmo e, finalmente, fazendo uma breve descrição da estrutura do documento.

No segundo capítulo é apresentado um resumo sobre GIS. Este capítulo resulta de um estudo que foi feito no início do estágio, com o propósito de perceber os conceitos e a finalidade de um sistema desta natureza.

No terceiro capítulo são apresentadas as tecnologias que foram usadas durante a fase de implementação do projecto, nomeadamente, XML *(Extensible Markup Language)* e CORBA *(Common Object Request Broker Arvhitecture)*. São feitas apresentações gerais de ambas as tecnologias, uma vez que apresentações detalhadas cairiam fora do âmbito deste documento. Aqui resumem-se os conceitos que as tecnologias envolvem e as respectivas utilidades.

No quarto capítulo é apresentada a fase de especificação do projecto de estágio. Neste capítulo são feitas referências ao processo de desenvolvimento de *software* praticado na

Siemens, ao ambiente de desenvolvimento e ao planeamento do projecto, aos requisitos definidos e à arquitectura adoptada.

No quinto capítulo é apresentada a fase de desenvolvimento do projecto de estágio. Apresentam-se os componentes desenvolvidos e referem-se alguns problemas encontrados. É também feita uma alusão aos testes realizados e aos resultados obtidos.

No sexto e último capítulo, temos as conclusões do projecto de estágio. Inclui algumas considerações gerais relativamente ao projecto de estágio e ainda algumas considerações relativamente a perspectivas de desenvolvimento do projecto

2 Sistemas de Informação Geográfica

Neste capítulo é feita uma descrição dos *Sistemas de Informação Geográfica* (GIS). O capítulo baseia-se num estudo realizado por Gilberto Câmara nesta área [1].

2.1 Introdução

Actualmente existem múltiplas definições para os GIS e cada uma delas tenta, de certa forma, privilegiar aspectos de uma tecnologia que, estando na fronteira de várias áreas do conhecimento, é encarada e absorvida de formas distintas pelos especialistas de cada área. Algumas definições possíveis seriam:

"GIS são sistemas automatizados usados para armazenar, analisar e manipular dados geográficos, ou seja, dados que representam objectos e fenómenos em que a localização geográfica é uma característica inerente à informação e indispensável para analisá-la." [1]

"(...) an information system that is designed to work with data referenced by spatial or geographic coordinates. In other words, a GIS is both a database system with specific capabilities for spatially referenced data, as well as a set of operations for working [analysis] with the data." [2]

Ao analisar as definições dadas, facilmente se pode perceber que estão essencialmente centradas nos dados geográficos, cuja existência e utilidade, mais do que comprovada, constituem a razão de ser de qualquer GIS. Na realidade os sistemas de informação convencionais há muito que procuram formas de representar, alfanumericamente, dados geográficos. No entanto, a riqueza e o real valor da informação geográfica dificilmente é capturada de forma adequada usando unicamente formas de representação alfanuméricas. Por este motivo surgiram novas necessidades de recursos que permitissem extrair mais valor da informação geográfica disponível – surgem então os GIS.

Mas afinal em essência o que é um GIS? Um GIS é um *software* (uma tecnologia) que efectua a associação entre informação geográfica e bases de dados convencionais, mantendo dentro do possível uma interface com o utilizador baseada em gráficos, nomeadamente mapas geográficos. Desta forma, é possível pesquisar e obter informação, não só através das suas características alfanuméricas, mas também através da sua localização espacial. Basicamente, os GIS combinam um conjunto de técnicas cuja utilização oferece ao administrador, arquitecto, engenheiro, etc, uma visão inédita do seu ambiente de trabalho, em que todas as informações disponíveis sobre determinado assunto estão ao seu alcance e relacionadas com base no que lhes é fundamentalmente comum – a geografia.

A recolha de informação sobre a distribuição geográfica de recursos minerais, animais, plantas, etc, sempre desempenhou um papel importante nas actividades das sociedades organizadas. Até há relativamente pouco tempo esta recolha era, no entanto, feita apenas usando documentos e mapas em papel, o que facilitava o acesso e a manipulação de informação, mas dificultava profundamente qualquer análise que combinasse, em simultâneo, vários mapas e dados. Esta análise agregada tornou-se possível com o desenvolvimento simultâneo, na segunda metade do século passado, de tecnologias em diversas áreas: computadores, bases de dados, computação gráfica, cartografia, topografia, investigação

operacional, optimização de redes, etc. Os GIS representam a consolidação de todos estes desenvolvimentos e técnicas sob a forma de ferramentas práticas e de utilidade real.

Sempre que a questão *"onde?"* esteja incluída no leque de problemas a serem solucionados por um sistema informático, existe uma forte possibilidade de adoptar um GIS. As funcionalidades deste tipo de sistemas revolucionaram a forma de trabalhar e de pensar das pessoas que usam a geografia no seu dia a dia. Possibilitam projectar situações potenciais ou ideais, fazer previsões e desenvolver modelos de simulação, isto para além da convencional análise dos dados existentes.

A utilização deste tipo de sistemas permite criar modelos que se aproximam bastante da nossa percepção intuitiva da realidade, o que faz com que o utilizador consiga interagir melhor com a informação que tem à sua disposição, aumentando assim a sua produtividade e a qualidade do seu trabalho.

2.2 Evolução

A ideia de apresentar diferentes camadas de dados em vários mapas e tentar, depois, relacioná-los por sobreposição já é antiga. Existem mapas da batalha de *Yorktown*, desenhados pelo cartógrafo francês Louis-Alexandre Berthier, que mostravam movimentos de tropas usando esse princípio. Em meados do século XIX, o "Atlas to Accompany the Second Report of the Irish Railway Comissioners" apresentava dados relativos à população, fluxo de tráfego, geologia e topografia sobrepostos no mesmo mapa básico – um caso de utilização empresarial e não militar. Em 1854 o Dr. John Snow usou um mapa para representar as localizações dos casos de morte por cólera no centro de Londres, conseguindo localizar um poço contaminado que estaria na origem do surto da doença – um dos primeiros casos de utilização de análise geográfica orientada à saúde pública.

Apesar destes exemplos remotos do uso de GIS, o arranque da concepção dos GIS tal como hoje são conhecidos só se deu no início dos anos 60. Neste período e apesar da precariedade dos recursos disponíveis começaram a ser desenvolvidas algumas aplicações reais, de entre as quais se destacam:

- Planos integrados de transportes, desenvolvidos nas décadas de 50 e 60 em Detroit e Chicago.
- O Sistema de Informações Geográficas do Canadá, projecto que arrancou em 1962 e processou os dados recolhidos pelo *Canada Land Inventory*, cruzando mapas com vários tipos de informação.
- O projecto STORET[1], do serviço de saúde pública dos Estados Unidos, que estando orientado ao suprimento de água potável e controle de poluição, agregou a informação relativa à qualidade da água e processos de tratamento que foi recolhida por diferentes organizações.
- O projecto MIDAS (1964), do serviço florestal americano, que foi considerado o primeiro GIS para administração de recursos naturais.
- O DIME, do *U.S. Bureau of the Census*, também na década de 60, que foi desenvolvido para fazer representações digitais de ruas e zonas censitárias.

[1] Mnemónica de *STOrage and RETrieval* (STORET)

Também do ponto de vista académico se verificou uma importante contribuição para o desenvolvimento da tecnologia, sendo a maior dada por Harvard onde Howard Fisher iniciou um projecto de investigação para desenvolver um *software* de mapeamento de uso geral. Aqui foi criado o "Harvard Laboratory for Computer Graphics and Spacial Analysis" que influenciou fortemente o desenvolvimento dos GIS até ao início da década de 80.

Existem duas grandes famílias de visualização de dados em GIS: o GIS *vector* e o GIS *raster*. Nos GIS orientados ao vector as imagens são descritas usando elementos geométricos (pontos, linhas e polígonos) posicionados num sistema de coordenadas cartesianas. Por exemplo, a localização de um ponto pode ser dado pelas suas coordenadas; objectos com características lineares (para fins de GIS), tais como estradas e rios, podem ser armazenados como uma sequência de coordenadas; os polígonos que podem ser usados para modelar áreas como distritos, concelhos, etc, podem ser armazenados como um ciclo de coordenadas. O armazenamento de dados sob a forma vectorial pode ser muito útil para descrever modelos de características discretas, sendo menos indicado quando se trata de modelos com características relativamente estáveis, como por exemplo os tipos de solo de uma dada área; Nos GIS orientados ao *raster* as imagens são digitalizadas e representadas segundo uma matriz de pixeis. Este tipo de GIS é essencialmente orientado a modelos cujas características mudam continuamente.

Tal como foi dito anteriormente, foi na década de 60 que os GIS se começaram a desenvolver e a assemelhar ao que conhecemos hoje. Os GIS orientados ao *raster* desenvolveram-se mais rapidamente, principalmente por terem as suas estruturas de dados bastante semelhantes às usadas em técnicas de digitalização, que naquela época estavam já razoavelmente desenvolvidas. Por outro, lado os GIS orientados ao vector tiveram um desenvolvimento muito mais lento, isto porque os algoritmos de processamento vectorial estavam ainda a começar a aparecer e eram bastante "pesados" em termos computacionais para as máquinas da altura.

O GIS orientado ao *raster* mais conhecido que apareceu nessa época foi o MAP *(Map Analysis Package)*, desenvolvido por Dana Tomlin. Nessa mesma época, Jack Dangermond, que trabalhou no laboratório de Harvard, deu início ao desenvolvimento de um GIS orientado ao vector que se tornou no actualmente conhecido Arc/Info (um dos GIS mais usados em todo o mundo).

Na década de 70, e graças ao rápido desenvolvimento dos computadores, os GIS orientados ao vector sofreram um avanço acentuado. Esse avanço gerou o declínio do uso e desenvolvimento dos GIS orientados ao *raster*, que passaram a ser encarados como uma solução de segunda classe. Esta visão surge principalmente devido à fraca resolução espacial que estes sistemas ofereciam e por requererem grande capacidade de armazenamento de dados.

Nos anos 90 renasceu o interesse sobre os GIS orientados ao *raster*. Isto aconteceu porque os utilizadores chegaram à conclusão que, para uma dada situação, uma das duas famílias de GIS seria mais adequada do que a outra, havendo até casos em que se complementariam. Surgem assim as primeiras soluções GIS integradas ou mistas.

Assim e em resumo, podemos classificar os GIS, do ponto de vista tecnológico, segundo três grandes grupos, a saber:

- Os cartográficos, que eram sistemas herdeiros da Cartografia e com suporte limitado de bases de dados, constituindo o paradigma típico de trabalho com um mapa.

Começaram a ser desenvolvidos no início da década de 80 para ambientes da família VAX e, a partir de 1985, para sistemas PC/DOS, sendo principalmente usados em projectos isolados que não requeriam a criação de arquivos digitais de dados. Este tipo de GIS pode ser encarado como orientado a projecto *(project-oriented GIS)*.

- Os bancos de dados geográficos, que chegaram ao mercado no início da década de 90 e foram concebidos para serem usados num ambiente cliente-servidor, ligados a um sistema de gestão de bases de dados convencionais e com capacidades de processamento de imagem. Este tipo de GIS pode ser encarado como orientado ao suporte organizacional *(enterprise-oriented GIS)*.
- Bibliotecas geográficas digitais ou centros de dados geográficos. Surgem como evolução do segundo grupo e chegaram ao mercado há relativamente poucos anos. Caracterizam-se por fazerem a gestão de grandes bases de dados geográficas que podem ser acedidas através de redes locais ou remotas, usando a *Web* como *interface*. São sistemas orientados à troca de informação entre organizações e cidadãos que normalmente têm acesso a bases de dados públicas *(society-oriented GIS)*.

2.3 Estrutura de um GIS

Numa visão abrangente, podemos identificar num GIS cinco componentes fundamentais:

- Interface com o utilizador
- Entrada e integração de dados
- Consulta e análise espacial
- Visualização e impressão
- Armazenamento de dados (organizados sob a forma de uma base de dados geográfica)

Tipicamente estes componentes organizam-se de uma forma hierárquica. No nível mais próximo do utilizador está a interface que define como trabalhar e controlar o sistema. No nível intermédio o GIS fornece, normalmente, mecanismos de processamento de dados espaciais (entrada, edição, análise, visualização e saída). No nível mais interno temos o sistema de armazenamento de dados, que é orientado para dados de cariz geográfico e permite fazer pesquisas de dados espaciais e seus atributos.

De uma forma geral, as funções de processamento de um GIS actuam sobre os dados numa "área de trabalho". A ligação entre os dados geográficos e as funções de processamento é, normalmente, feita usando mecanismos de selecção e pesquisa que definem restrições sobre o conjunto de dados. Dois exemplos ilustrativos de selecção de dados são:

- "Mostrar as cidades do distrito do Porto com população entre os 10 000 e os 20 000 habitantes" *(consulta por atributos espaciais e não espaciais)*.
- "Mostrar as antenas rádio num raio de 5 km da Câmara Municipal do Porto" *(consulta com restrições espaciais)*.

Na Figura 2 temos um diagrama que ilustra a arquitectura tipo (de alto nível) e as relações entre os subsistemas (ou componentes) principais de um GIS. Cada GIS, de acordo com os seus objectivos e necessidades, implementa os diferentes componentes de forma distinta.

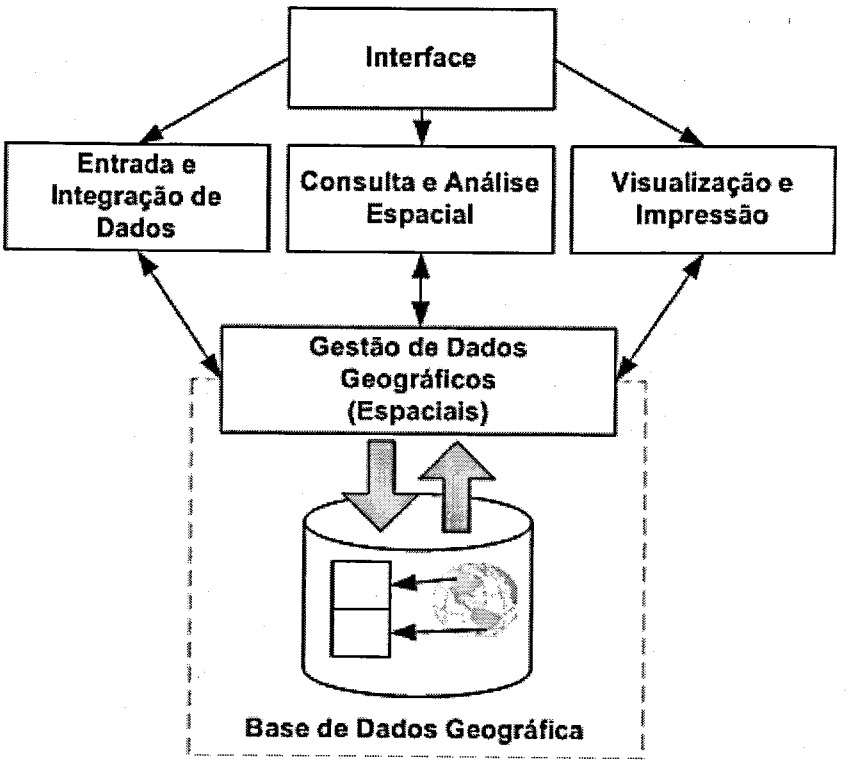

Figura 2: Estrutura base de um GIS

Os GIS actualmente presentes no mercado funcionam segundo uma variedade de arquitecturas internas, tendo cada uma delas pontos fracos e fortes que poderão ter grande influência em aspectos como: desempenho, capacidade de gestão de grandes bases de dados, capacidade de utilização simultânea por vários utilizadores e capacidade de integração com outros sistemas.

Com algumas alterações, ou maior nível de especificação e detalhe, qualquer GIS comercial existente hoje no mercado enquadra-se em uma das seguintes arquitecturas internas que serão apresentadas nas sub secções seguintes:

- Tradicional
- Dual
- Baseada em CAD
- Relacional
- Orientada a objectos
- Baseada em *desktop mapping*
- Baseada em *rasters*
- Integrada (*rasters* e vectores)

As primeiras cinco arquitecturas correspondem especificamente a GIS orientados ao vector. Isto deve-se ao facto de existirem muitas variações para a filosofia de armazenamento e utilização de vectores, o que é reflectido claramente nos GIS. Este número de variações não se verifica nos sistemas orientados ao *raster*, uma vez que o processamento de imagem é mais uniforme entre as diversas aplicações comerciais existentes.

É importante, no entanto, salientar que com o crescimento constante do nível de complexidade dos GIS comerciais, esta separação (divisão) de arquitecturas começa a perder gradualmente nitidez. Um exemplo típico deste fenómeno é a crescente incorporação de recursos e funções a sistemas *desktop mapping*, tornando-os mais poderosos e aproximando-os do que seria denominado *"desktop GIS"* [3].

Arquitectura Tradicional

Na Figura 3 é apresentado um diagrama que ilustra a arquitectura interna dos GIS mais tradicionais. Por "tradicional" entenda-se que se trata da arquitectura dos primeiros GIS, concebidos numa época em que a novidade estava na integração de dados gráficos com dados alfanuméricos num ambiente comum.

Nos sistemas que seguem este tipo de arquitectura o utilizador pode aceder aos dados usando a interface gráfica[2] ou através de uma linguagem de programação, normalmente muito simples, constituída essencialmente por macro comandos que oferecem a possibilidade de encadear comandos que são disponibilizados na interface gráfica. Do ponto de vista do utilizador, a linguagem de programação acaba por servir como segunda *interface* semelhante a uma linha de comandos. Neste tipo de sistemas existe também a possibilidade de personalização da GUI, de modo a incluir comandos ou sequências de comandos (*macros*) desenvolvidos pelo utilizador.

Os comandos formulados usando a GUI ou a linguagem de programação são executados pelo "núcleo" *(core)* do sistema, que é responsável pelo processamento das funções geográficas e pela gestão dos dados. Este "núcleo" pode ser implementado de forma monolítica, contendo todas as funções do GIS. No entanto é mais comum, até por questões comerciais, ser implementado de forma modular, cabendo ao "núcleo" unicamente a implementação de funcionalidades básicas e de gestão de dados. Estes módulos (edição gráfica/topológica, impressão de mapas, processamento de redes, modelagem digital, gestão de imagens, etc) podem ser comercializados separadamente, o que permite ao utilizador configurar o seu ambiente de trabalho com custos mais reduzidos.

O principal aspecto desta arquitectura reside na forma como é feita a gestão dos dados gráficos e alfanuméricos. A maioria dos produtores opta por incluir a codificação dos dados gráficos em estruturas próprias, ou seja, estruturas de dados concebidas e implementadas dentro do ambiente do produtor e tratadas como segredo comercial. Desta forma os dados gráficos são codificados em ficheiros binários, cuja leitura e interpretação (correcta) só pode ser feita por quem conhecer a estrutura de codificação.

A codificação dos dados alfanuméricos segue a mesma lógica, embora neste caso não exista a preocupação de ocultar a forma de armazenamento. Normalmente é usada uma estrutura tabular, semelhante à dos Sistemas de Gestão de Bases de Dados (SGBD) relacionais, com registos de tamanho fixo. Desta forma para se conseguir interpretar dados alfanuméricos é preciso conhecer a sua estrutura (número de campos e respectivo tamanho e tipo de conteúdo). Tipicamente, o tratamento dos dados alfanuméricos é feito num ambiente próprio dedicado à gestão de bases de dados, sendo este ambiente totalmente integrado no produto, não tendo qualquer sentido fora dele.

---

[2] A interface gráfica é representada no diagrama usando a mnemónica GUI que representa *Graphical User Interface*.

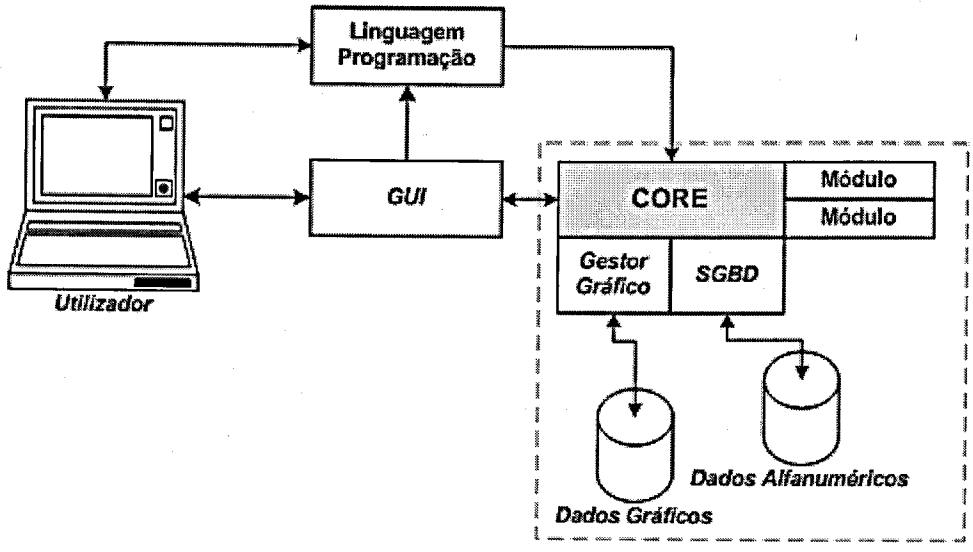

Figura 3: Diagrama da arquitectura tradicional

Resumo das características principais:

- *Gestão em separado de dados gráficos e alfanuméricos.*
- *Armazenamento de dados gráficos em estruturas proprietárias (abrangidas pelo segredo comercial).*
- *Armazenamento de dados alfanuméricos em bases de dados geralmente proprietárias e integradas inteiramente no produto.*

Arquitectura Dual

Na Figura 4 temos o diagrama que ilustra a arquitectura dual, uma evolução da arquitectura tradicional apresentada no ponto anterior do documento. A única diferença da arquitectura anterior para esta reside no facto desta adoptar um SGBD relacional externo para gerir os dados alfanuméricos. Esta opção reflecte essencialmente a intenção de não "reinventar a roda", utilizando produtos disponíveis no mercado para executar parte das tarefas do GIS. Na implementação pouco muda conceptualmente, verificando-se apenas uma maior facilidade no que se refere ao desenvolvimento do "núcleo" do sistema. No entanto, o GIS e o SGBD relacional, sendo produtos diferentes, têm que comunicar de alguma forma para que possam realizar as funções que lhes estão destinadas. Esta comunicação é gerida pelo "núcleo" e é restrita às operações básicas de bases de dados: inserção, remoção e consulta de dados.

Do ponto de vista do utilizador, esta alternativa permite o desenvolvimento de aplicações convencionais dentro do ambiente de um SGBD relacional, que partilhem os atributos alfanuméricos dos dados geográficos. O problema que surge com esta implementação prende-se com o facto do SGBD relacional não conhecer a estrutura gráfica proprietária, existindo assim o sério risco de serem introduzidas inconsistências na base de dados geográfica. Imagine-se, por exemplo, que um utilizador que usa exclusivamente a vertente alfanumérica da aplicação apaga da base de dados um registo alfanumérico, que representa um conjunto de atributos para determinada informação geográfica. Dado que o GIS desconhece que esta informação geográfica deixou de ter atributos, a informação torna-se assim inconsistente. Por este motivo o acesso aos atributos alfanuméricos da informação geográfica só poderá ser feita seguindo determinados critérios que terão que ser mantidos e controlados pela aplicação.

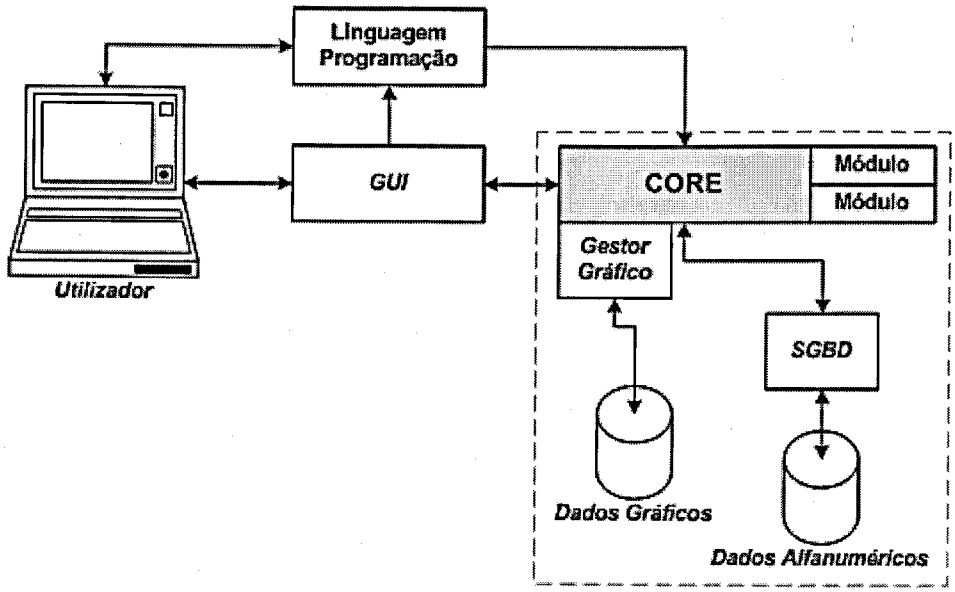

Figura 4: Diagrama da arquitectura dual

Resumo das características principais:

- *Semelhantes à arquitectura anterior, exceptuando a gestão da base de dados alfanuméricos que neste modelo é feita por um sistema externo, tipicamente um SGBD relacional padrão de mercado.*

Arquitectura Baseada em CAD

Uma extensão lógica do raciocínio que levou ao armazenamento de dados alfanuméricos em SGBD relacionais padrão de mercado corresponde à implementação da gestão dos dados gráficos usando também ferramentas padrão de mercado. Neste contexto, as ferramentas que melhor se adequam às necessidades dos GIS são os sistemas CAD *(Computer Aided Systems)*. A Figura 5 apresenta uma arquitectura deste tipo.

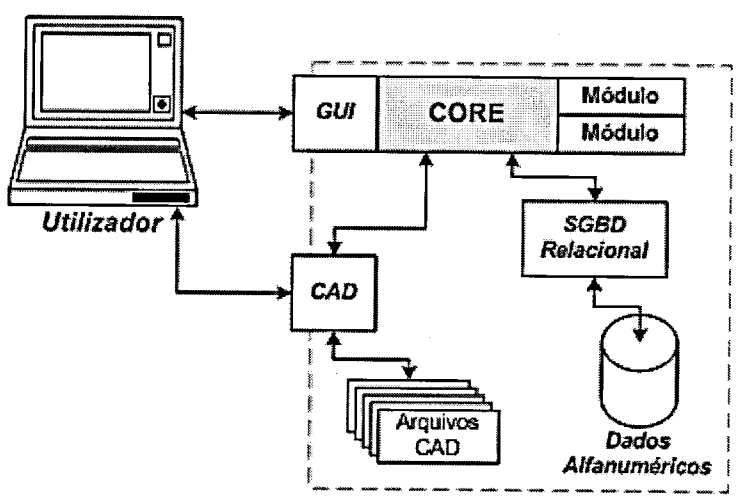

Figura 5: Diagrama da arquitectura baseada em CAD

Nesta arquitectura, o núcleo do GIS, para além de implementar e disponibilizar as funcionalidades geográficas básicas, trata da integração entre o gestor gráfico (CAD) e o

SGBD. Funcionalidades geográficas adicionais ou mais específicas são fornecidas usando o sistema de módulos, tal como nas arquitecturas já apresentadas. A *interface* gráfica do sistema é geralmente baseada na do sistema CAD.

Os GIS baseados nesta arquitectura têm dois (grandes) pontos fracos. O primeiro, mais grave, está relacionado com a facilidade com que se podem introduzir inconsistências na base de dados geográfica, à semelhança da arquitectura anterior. O segundo está relacionado com a utilização de arquivos CAD para armazenar os dados gráficos. Uma vez que os sistemas CAD não possuem recursos de indexação espacial, o acesso aos arquivos é puramente sequencial, degradando assim significativamente o desempenho do sistema [1]. O principal ponto forte reside na facilidade de utilização dos recursos de edição.

Resumo das características principais:

- *Gestão de dados gráficos feita por uma ferramenta CAD, geralmente externa ao GIS.*
- *Gestão de dados alfanuméricos feita por um SGBD relacional padrão de mercado e externo ao GIS.*
- *Possibilidade de manipulação directa dos arquivos gráficos usando sistemas CAD.*

Arquitectura Relacional

A combinação de problemas das arquitecturas anteriores, em especial a possibilidade de introdução de inconsistências na base de dados geográfica, levou à criação de uma nova abordagem. A ideia principal desta nova arquitectura – Figura 6 – consiste em aplicar, na gestão da base de dados geográfica, os excelentes recursos disponibilizados pelos SGBD relacionais (que permitem garantir a integridade, controlar os acessos e recuperar falhas).

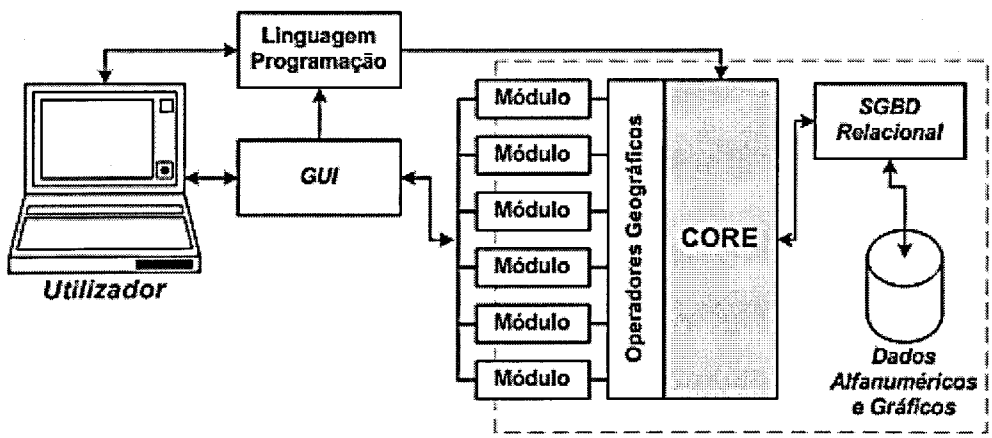

Figura 6: Diagrama da arquitectura relacional

Nos GIS relacionais, os dados gráficos são organizados em tabelas, tal como os dados alfanuméricos. É usado um sistema de chaves para relacionar as tabelas que formam assim um esquema relacional cuja integridade é garantida pelo SGBD.

Desta forma todas as funções de gestão de dados ficam a cargo do SGBD. No entanto, para realizar os tipos de consultas e operações mais frequentes no ambiente GIS, têm que ser feitas algumas extensões como: implementação de mecanismos de indexação espacial, inclusão de operadores geográficos ("contém", "contido", vizinho"...) como extensões da linguagem SQL.

Estas extensões são normalmente implementadas no núcleo do GIS que é depois responsável por traduzi-las em operações previamente existentes no SGBD.

Resumo das características principais:

- *Os dados geográficos e alfanuméricos são armazenados de forma integrada na base de dados relacional externa ao GIS.*
- *Implementação de recursos de geoprocessamento (operadores espaciais, ferramentas de análise, etc) apresentadas como extensões ou complementos ao modelo relacional.*
- *Grande robustez de implementação (devido às garantias de integridade dadas pelo esquema relacional)*
- *Grande estabilidade (devido ao avançado grau de desenvolvimento dos SGBD relacionais).*

Arquitectura Orientada a Objectos

Esta arquitectura é bastante similar à anterior, apresentando a diferença de: o armazenamento de dados geográficos utilizar objectos. Esta função é realizada por um SGBD orientado a objectos, que pode ser um produto genérico de mercado ou um produto proprietário. Todas as operações executadas no âmbito do GIS são baseadas num modelo de dados orientado a objectos, que contém toda a informação sobre cada classe de objectos, incluindo características gráficas e alfanuméricas e aspectos do seu comportamento – Figura 7

Aproveitando estas características, a implementação do GIS com uma arquitectura cliente-servidor passa a ser mais natural, uma vez que o tráfego entre um *"núcleo cliente"* e um *"núcleo servidor"* pode consistir apenas em objectos que circulam numa rede. A comunicação entre o servidor e a base de dados (orientada a objectos) pode, se não for uma ligação proprietária, ser implementada com base em padrões como o ODTP *(Object Data Transfer Protocol)* ou o CORBA *(Common Object Request Broker Architecture)*. Na prática, no entanto, a preferência ainda é pelas implementações proprietárias, com o núcleo servidor fortemente integrado com o SGBD orientado a objectos.

Outro ponto de destaque desta arquitectura é a linguagem de programação. Esta é em geral uma linguagem computacionalmente completa, dotada de recursos que permitem fazer uso da maior riqueza semântica introduzida pelo modelo de dados orientado a objectos. Seria muito complicado utilizar uma linguagem como o SQL para aceder a este tipo de base de dados, o que torna ainda mais relevante a existência de uma linguagem computacionalmente completa (e tipicamente mais poderosa).

Existe a possibilidade de ligar o *"núcleo cliente"* a um SGBD relacional externo, desde que este atenda a alguns padrões, viabilizando assim a integração do GIS com aplicações convencionais externas. É no entanto importante salientar que é da responsabilidade das aplicações garantir a integridade entre os dados mantidos pelo GIS, sob a forma de objectos, e os dados mantidos pelo SGBD relacional.

Resumo das características principais:

- *Presença (marcante) de um módulo responsável pela modelação de dados.*
- *Possibilidade de ligação a um SGBD relacional externo.*
- *Tendência para se basear em padrões, sistemas abertos e filosofia cliente-servidor.*

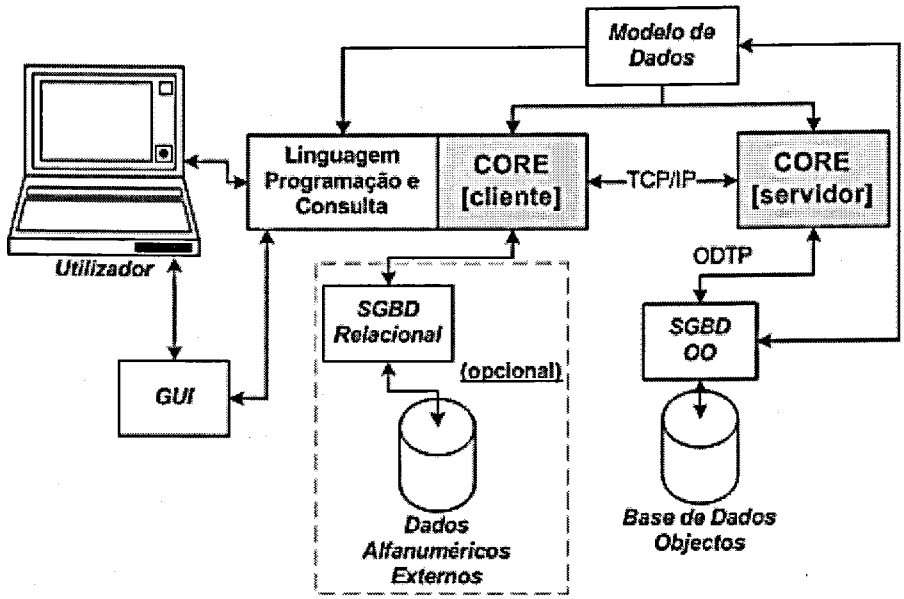

Figura 7: Diagrama da arquitectura orientada a objectos

Arquitectura Baseada em Desktop Mapping

Os GIS baseados nesta arquitectura são normalmente sistemas que se concentram em facilitar as actividades de apresentação de informações sob a forma de mapas. Não são, no entanto, sistemas adequados para actividades de cartografia, uma vez que não possuem recursos de edição e de entrada de dados muito sofisticados. Também não são indicados se o objectivo é gerir um grande volume de informação, uma vez que a sua estrutura de arquivos tende a ser bastante simples, usando até (muito frequentemente), de forma directa, arquivos gráficos e alfanuméricos de outras aplicações (AutoCAD, Excel, Access, etc) – Figura 8.

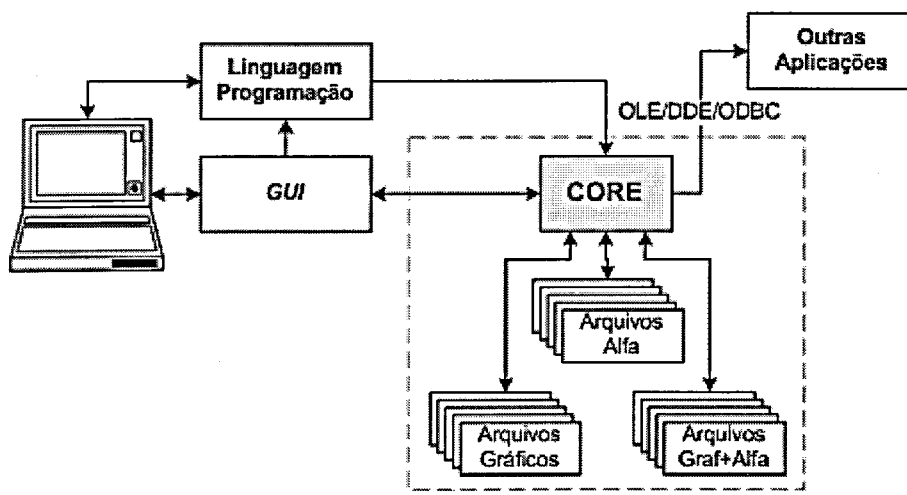

Figura 8: Diagrama da arquitectura baseada em desktop mapping

O ponto forte das aplicações que são projectadas usando os conceitos de *desktop mapping* reside exactamente na facilidade de integração de dados de diversas fontes. Como se tratam de aplicações orientadas especificamente ao utilizador final, isto é, o profissional de uma determinada área de interesse que pretende criar um mapa para incluir num relatório, ou até

facilitar uma análise espacial, são normalmente desenvolvidas em ambiente *Windows*. Têm custos relativamente baixos, apresentando a tendência de incorporar cada vez mais funções.

A interface com o utilizador, como no caso de outras aplicações *Windows*, pode ser customizada com o auxílio de uma linguagem de programação simples, tipicamente no estilo *Visual Basic*. Esta linguagem também permite a criação de alguns tipos de funções e aplicações limitadas.

A comunicação destes sistemas com outras aplicações é normalmente feita usando alguns dos recursos mais comuns do *Windows*, como OLE *(Object Linking and Embedding)*, DDE *(Dynamic Data Exchange)* e ODBC *(Open Database Connectivity)*.

Resumo das características principais:

- *Geralmente baseada em Windows; concentra esforços numa boa interface com o utilizador e em recursos para produção de mapas.*
- *Recurso frequente a arquivos externos independentes.*
- *Ausência de um sistema de gestão de dados gráficos ou alfanuméricos, podendo, no entanto, em geral comunicar com sistemas existentes deste tipo.*

Arquitectura Baseada em Rasters

Nas aplicações baseadas neste modelo, a verdadeira base de dados está na imagem e nos seus atributos. Assim sendo, não existe um SGBD propriamente dito, existe apenas um conjunto de arquivos de imagem codificados da forma mais conveniente, tanto em termos de ocupação de espaço em disco como em termos de facilidade de recuperação – Figura 9.

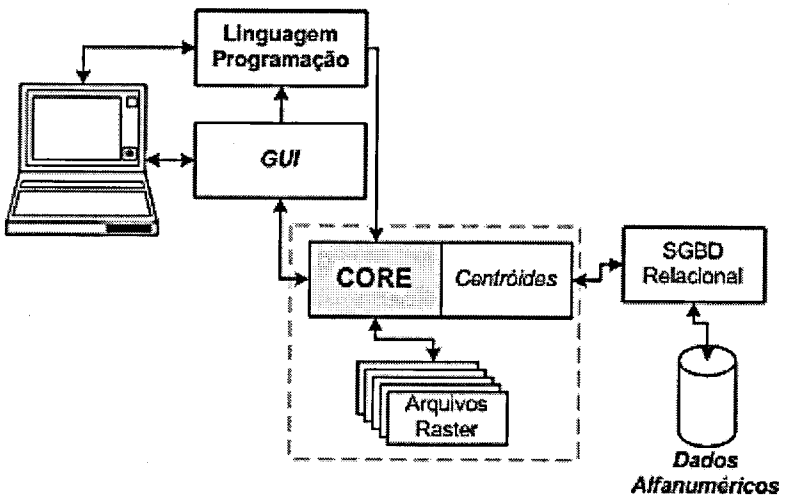

Figura 9: Diagrama da arquitectura baseada em *rasters*

No entanto, é muitas vezes (na maior parte das vezes) necessário associar um conjunto de informações alfanuméricas a uma imagem ou a partes de uma imagem. O recurso que é geralmente usado por este tipo de GIS é a criação de objectos vectoriais, que podem ou não ser apresentados (normalmente sobrepostos à imagem). Estes objectos podem ser pontos, linhas ou áreas, existindo no caso das linhas ou áreas um *"centroíde"* (ou ponto chave), que consiste num único ponto contido na linha ou na área que é escolhido para ser o ponto de referência dos dados alfanuméricos correspondentes. Assim, cada elemento geográfico com

possibilidade de associação com informação alfanumérica é associado ao *"centróide"*, que por sua vez está representado numa base de dados (normalmente relacional).

Resumo das características principais:

- *Informações gráficas são normalmente armazenadas como arquivos independentes (devido aos grandes volumes).*
- *Comunicação com uma base de dados relacional externa é feita através de vectores que são definidos sobre a imagem.*

Arquitectura Integrada

Na área ambiental, onde é grande a necessidade de integração de dados de diferentes formatos, como imagens, mapas, modelos de terreno, etc, uma das tendências vem sendo o desenvolvimento de tecnologias que permitam o tratamento simultâneo de dados matriciais e dados vectoriais. Devido à grande capacidade de armazenamento necessária para imagens, e dado que os SGBD (mais convencionais) de mercado não possuem ainda um suporte eficiente para este formato, a arquitectura de um GIS integrado é normalmente uma extensão da arquitectura dual que inclui gestão de arquivos gráficos no formato matricial *(raster)* – Figura 10.

Resumo das características principais:

- *Gestão independente de dados gráficos e alfanuméricos.*
- *Armazenamento de gráficos em estruturas proprietárias e de dados alfanuméricos em bases de dados relacionais.*
- *Capacidade de processamento de dados vectoriais e de imagens.*

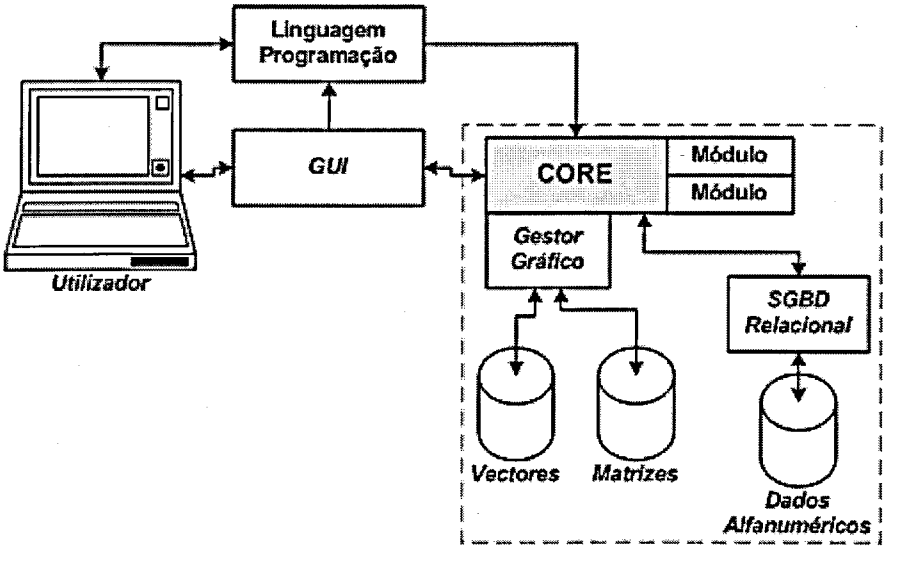

Figura 10: Diagrama da arquitectura integrada

2.4 Funcionalidade

De uma forma geral, cada GIS é inicialmente desenhado para solucionar um conjunto muito específico de problemas, evoluindo posteriormente para se tornar uma ferramenta de uso mais

abrangente. Esta vocação original, típica do *software,* não limita no entanto o alcance (abrangência) deste tipo de sistemas, mas a verdade é que, na realidade, cada um terá os seus pontos fortes e pontos fracos. Não se pode, portanto, esperar encontrar um GIS que corresponda perfeitamente às necessidades de determinado projecto, pois esse sistema provavelmente não existe.

No entanto existem funcionalidades básicas, típicas dos GIS e que podem ser encontradas em qualquer *software* deste tipo, e outras que os distinguem, nomeadamente ao nível da entrada de dados, gestão e recuperação de informação e manipulação e análise.

Funcionalidade Básica

Como foi referido existem funcionalidades básicas que são típicas dos GIS. O que varia de caso para caso é a fiabilidade e a qualidade da implementação. De qualquer forma, a disponibilidade deste tipo de funcionalidades pode ser suficiente para garantir o sucesso de muitas aplicações. De uma forma geral qualquer sistema deste tipo permite:

- Representar graficamente informação de natureza espacial, associando aos gráficos informação alfanumérica tradicional. Estas representações são feitas normalmente sob a forma de vectores (pontos, linhas e polígonos) e/ou imagens digitais (matrizes de *pixeis*).
- Pesquisar informação com base em critérios alfanuméricos, à semelhança de um sistema de gestão de bases de dados tradicional, e com base em relações espaciais topológicas, tais como: continência, adjacência, intercepção, etc[3].
- Limitar o acesso e controlar a entrada de dados através de um modelo de dados previamente definido.
- Visualizar rapidamente a informação geográfica, usando para isso algoritmos de indexação espacial[4].
- A importação e exportação de dados de/para outros sistemas deste tipo, ou para outro tipo de *software*, por exemplo *software* de aplicação gráfica.
- A entrada e manutenção de dados, utilizando equipamentos como o rato ou um *scanner*.
- Gerar resultados para uma variedade de dispositivo de saída como: monitores, impressoras, *plotters*, etc, sob a forma de mapas, gráficos e tabelas.
- Desenvolver funcionalidades específicas e customizar a interface de acordo com as necessidades do utilizador.

---

[3] ***Topologia:*** *Ramo da Matemática que estuda as propriedades das configurações geométricas que não são alteradas por transformações ou deformações elásticas homomórficas.* Isto é, topologia estabelece um conjunto de técnicas que nos permitem perceber as relações espaciais inerentes ao posicionamento relativo dos objectos independentemente das suas dimensões ou coordenadas exactas. Relações de continência (contém ou está contido), adjacência (ao lado de) e conexão (ligado a, relacionado com) são deduzidas com base em técnicas de topologia.

[4] ***Indexação Espacial:*** Procedimento através do qual é possível chegar rapidamente a uma lista de objectos contidos numa dada região (normalmente rectangular) do espaço. É utilizado quando se pretende seleccionar regiões bem delimitadas de uma grande base de dados (por exemplo, numa operação de *zoom*) sem que seja necessário percorrer todos os dados existentes.

Entrada de Dados

Ao contrário de alguns sistemas de bases de dados convencionais, o GIS é capaz de armazenar informações variadas, de natureza gráfica, como vectores e imagens. Para que isto seja possível, é necessário que o GIS possua módulos ou *interfaces* que permitam ao utilizador incorporar e visualizar dados gráficos no GIS. Para além disto, o sistema tem que ser capaz de detectar falhas ou inconsistências nos dados antes de os introduzir na base de dados geográfica. Em suma, no referente à entrada de dados, um sistema GIS deve ser capaz de:

- Permitir a digitalização de dados gráficos (recorrendo, por exemplo a um *scanner*), fornecendo posteriormente meios que permitam definir padrões vectoriais e associar informação alfanumérica (se existente) à imagem digitalizada.
- Permitir a associação de imagens digitais à base de dados geográfica. Isto pode ser feito através de recursos de georeferenciamento ou então introduzindo a própria imagem na base de dados.[5]
- Fazer análises de consistência relativamente aos dados vectoriais, tentando assim detectar erros na topologia ou inconsistências relativamente ao modelo de dados.
- Executar procedimentos de correcção sobre os dados adquiridos, tentando assim melhorar a sua qualidade e prepará-los para a incorporação na base de dados geográfica. Estes procedimentos incluem: *edge matching*, eliminação de vértices desnecessários, suavização de curvas, etc.
- Receber, converter e tratar dados provenientes de arquivos em formato padronizado de outros sistemas de informação, geográficos ou não.

Gestão e Recuperação de Informação

Uma vez formada a base de dados geográfica, o GIS tem que ser capaz de a gerir. Isto significa que o sistema tem que ser capaz de:

- Manter a consistência da base de dados entre operações realizadas pelos utilizadores.
- Garantir a integridade das relações entre dados gráficos e alfanuméricos.
- Controlar os acessos simultâneos aos dados.
- Executar operações de *backup* e recuperação de informação.
- Garantir a recuperação total ou parcial da base de dados em caso de falhas.
- Controlar acessos não autorizados aos dados.

De um modo geral, todas as operações acima referidas são típicas de sistemas de gestão de bases de dados. No caso dos sistemas GIS existe a complexidade adicional da incorporação de dados gráficos e consequente necessidade de garantia da integridade entre este tipo de dados e dados alfanuméricos.

[5] Esta funcionalidade implica que o sistema seja capaz de ler arquivos de imagem codificados em diversos formatos e convertê-los para um formato interno.

Naturalmente o GIS tem que garantir aos utilizadores um acesso rápido e eficiente à informação por ele mantida. Para que isto seja possível é necessário introduzir alguns recursos e técnicas que organizem a informação na base de dados de forma inteligente. Estas técnicas incluem a indexação espacial (que procura organizar a informação por proximidade geográfica) e outros recursos que permitam e facilitem a consulta da informação por parte do utilizador. Para esta finalidade, utiliza-se tipicamente uma linguagem de pesquisa, como o SQL, enriquecida por comandos e operadores de natureza espacial.

Manipulação e Análise

As funções de manipulação e análise de dados geográficos podem ser agrupadas de acordo com o tipo de dados que cada uma delas trata: análise geográfica, processamento de imagens, modelação de terreno, etc.

- ***Análise Geográfica:*** permite a combinação de informações temáticas. Pode ser feita no domínio vectorial ou matricial. Um conjunto importante de procedimentos de análise geográfica, definido por Tomlin (1990), denominado "Algebra de Mapas" representa a base para a implementação de operadores de análise em diferentes sistemas. Estas funções incluem: reclassificação, intersecção, operações boleanas e matemáticas sobre mapas e consultas a bases de dados.
- ***Processamento de Imagens:*** tratamento de imagens de satélite e de *scanners*. Com o advento dos satélites de alta resolução e de técnicas de fotogrametria digital, as imagens de satélite revelam-se cada vez mais úteis para estudos ambientais. Estas funções incluem: realce por modificação de histograma, filtragem espacial, rotação espectral, etc.
- ***Modelação de Terreno:*** permite o cálculo de declives, volumes, cortes transversais, etc. Estas funções incluem: gerações de mapas de contorno, visualização 3D, cálculo de volumes, análise de perfis, etc.

2.5 Dados Georeferenciados na Internet

Uma das linhas de pesquisa e desenvolvimento em geoprocessamento que cada vez mais desperta o interesse da comunidade é o acesso a informação geográfica via *Internet*. Como qualquer linha de desenvolvimento "recente" não existe nenhum fornecedor de especial destaque.

Nas primeiras abordagens ao problema era oferecido ao utilizador, através de um *browser*, um formulário onde eram introduzidas informações relativamente às áreas de interesse. Uma vez preenchido, este era transmitido para um servidor que o interpretava e convertia num mapa final em formato de imagem, como JPEG ou GIF. Esta imagem era então inserida numa página *web* (gerada em *runtime*) que posteriormente era transmitida para o utilizador. Sendo, à primeira vista, o processo mais natural do ponto de vista do *browser*, uma vez que lida com a apresentação de imagens, coisa que qualquer *browser* é capaz de fazer, é uma alternativa problemática por diversos motivos: não permite que o utilizador navegue interactivamente pelo mapa; a transmissão de imagens é geralmente demorada e, quando realizada de forma repetitiva, tende a sobrecarregar os recursos de rede; introduz uma sobrecarga no servidor que normalmente constrói o mapa a partir de uma base de dados vectoriais e depois o transmite para o cliente.

Pelos motivos apresentados, a transmissão de informação geográfica usando o formato de *raster* foi praticamente abandonado, uma vez que se torna muito mais interessante transmitir essa mesma informação em formato vectorial. Desta forma os objectos vectoriais transmitidos são guardados em memória na máquina cliente, o que permite ao utilizador uma maior liberdade e interactividade, permitindo mesmo operações como *zoom* ou o *pan*.

No entanto esta abordagem (transmissão de objectos vectoriais) tem um obstáculo: nenhum dos *browsers* está preparado para receber e apresentar informação neste formato. Para tornar isto possível, os fornecedores de informação geográfica optam por alternativas diferentes. Uns optam por criar um *plug-in*[6] que funciona ligado ao *browser* e desenha a informação vectorial no *ecrã* à medida que esta vai sendo recebida. Outros optaram por desenvolver uma aplicação *Java* que é transmitida no momento de acesso para ser executada na máquina do utilizador, dispensando assim procedimentos complicados de instalação – Figura 11.

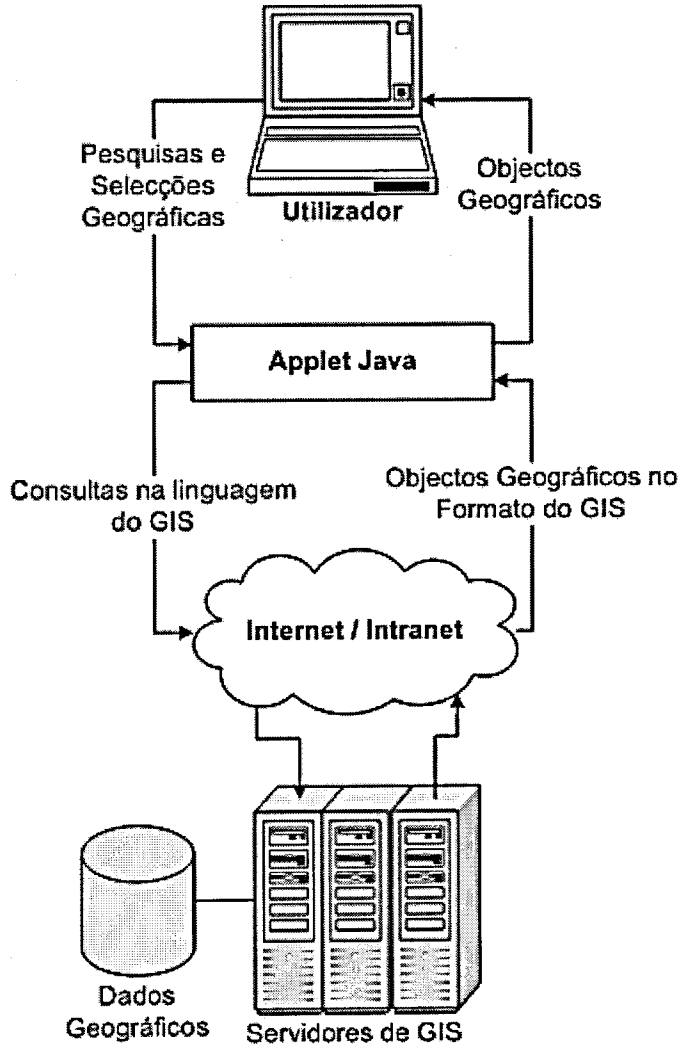

Figura 11: Acesso a dados geográficos via *Internet*

[6] ***Plug-In:*** programa que funciona no computador do utilizador, normalmente ligado ao *browser*.

2.6 Resumo

A recolha de informação de cariz geográfico sempre desempenhou um papel importante na actividade de sociedades organizadas. No entanto, a utilização do papel como suporte físico não representava uma boa solução, uma vez que tornava bastante difícil agregar e relacionar grandes quantidades de dados, o que originou a necessidade de desenvolver ferramentas capazes de desempenhar essas tarefas eficazmente – surgem então os sistemas GIS.

"GIS são sistemas automatizados usados para armazenar, analisar e manipular dados geográficos, ou seja, dados que representam objectos e fenómenos em que a localização geográfica é uma característica inerente à informação e indispensável para analisá-la." [1]

Tendo em conta esta definição, pode facilmente concluir-se que sempre que a questão *"onde?"* esteja incluída na lista de problemas a resolver por um sistema informático, a adopção de um GIS pode representar a solução ideal, uma vez que permite criar modelos que se aproximam bastante da nossa percepção intuitiva da realidade.

Numa visão de alto nível pode encontrar-se num GIS cinco módulos fundamentais: módulo de interface com o utilizador, módulo de entrada e integração de dados, módulo de consulta e análise espacial, módulo de visualização e implementação e módulo de armazenamento de dados. Normalmente estes módulos são organizados de forma hierárquica e funcionam segundo uma variedade de arquitecturas internas, a saber: tradicional, dual, baseada em CAD, relacional, orientada a objectos, baseada em *desktop mapping*, baseada em *rasters* e integrada.

De uma forma geral os sistemas GIS oferecem um pacote de funcionalidades básicas, nas quais se incluem: a representação gráfica de informação de natureza espacial, pesquisa de informação com base em critérios alfanuméricos, importação e exportação de dados, controlo de acesso, etc. Existem no mercado duas grandes famílias de GIS: os *GIS vector*, onde as imagens são descritas usando elementos geométricos posicionados num sistema cartesiano, e os *GIS raster*, onde as imagens são digitalizadas e representadas como uma matriz de pixeis.

Tendo em conta a evidente utilidade de um sistema desta natureza a comunidade *web* vem trabalhando cada vez mais em soluções para o fornecimento de informação geográfica *online*. Actualmente existem algumas soluções bastante interessantes que funcionam com *applets* de *Java* e que são fornecidos por alguns distribuidores de informação geográfica nos seus *sites*.

3 Tecnologias

Neste capítulo são apresentados os estudos feitos às tecnologias envolvidas no projecto de estágio: *Extensible Markup Language (XML)* e *Common Object Request Broker Architecture (CORBA)*. Para além destas duas tecnologias foi ainda utilizada a linguagem de programação *Java* que não será abordada neste documento. A exposição das análises feitas não será muito detalhada nem expositiva uma vez que não se enquadraria no contexto do documento.

3.1 Extensible Markup Language (XML)

Hoje em dia torna-se praticamente impossível evitar que a *World Wide Web* (WWW) e a *Internet* invadam a nossa vida e que tenham um papel de crescente importância no nosso dia a dia.

A *Internet,* que começou por ser uma pequena experiência realizada por um grupo de cientistas do ramo nuclear, é hoje um dos mais fenomenais acontecimentos da história da computação. Há até quem considere que vivemos o equivalente moderno à revolução industrial – o despertar da era da Informação [4].

Na sua proposta inicial ao CERN (*European Organization for Nuclear Research*) [5] em 1989, Tim Berners-Lee (o reconhecido inventor da *Web*) descreveu a sua visão do que é hoje a *Internet* como:

"... a universal linked information system, in which generality and portability are more important than fancy graphics and complex extra facilities."

A utilização da *web*, que está, segundo muitos, ainda na sua "infância", ultrapassou já as fronteiras de meras páginas familiares e o comércio electrónico começa a despertar e a desempenhar um papel determinante. Quando se fala em comércio electrónico não se limita o conceito à possibilidade de encomendar livros, cd's, roupa, ou qualquer outro tipo de material através da *web*, uma vez que isto já é feito há alguns anos e com alguns exemplos de bastante sucesso, como a *Amazon.com*. A definição de comércio electrónico vai um pouco mais longe do que isto e baseia-se nas necessidades emergentes dos últimos anos, e que incluem:

- Utilização da *Internet* para juntar partes de empresas distribuídas globalmente, formando um único bloco.
- Utilização da *Internet* para troca de informação relativa a transacções financeiras (transacções de cartões de crédito, transacções bancárias, etc).
- Utilização da *Internet* para troca de informação médica entre pacientes, médicos, hospitais e companhias de seguros.
- Utilização da *Internet* para fazer a distribuição de *software*: incluindo a possibilidade de modularizar grandes pacotes de *software* presentes em aplicações como por exemplo o *Microsoft Word* para que o utilizador possa usar e pagar apenas as partes que necessita; incluindo a possibilidade de criação de *software* que corre directamente na *Web* sem necessidade de instalação;...

Todas estas possibilidades não passam de meras possibilidades recorrendo apenas à tradicional *Hypertext Markup Language* (HTML). A *Extensible Markup Language* (XML) é uma linguagem usada na estruturação de documentos que foi criada com o intuito de melhorar

a funcionalidade da *web*, fornecendo uma forma mais flexível de identificar a informação. De certa forma, o XML é a tecnologia que possibilita e impulsiona o aparecimento de uma nova forma de ver e trabalhar a *web*, e é provavelmente o acontecimento mais importante (neste contexto) desde o aparecimento do *Java*.

O nome *Extensible* surge porque se trata de uma linguagem que não usa um formato fixo (como, por exemplo, o HTML que é uma linguagem de estruturação predefinida). Na realidade o XML é uma metalinguagem usada para descrever linguagens de estruturação que permitem definir formatos (estruturas) de documentos; isto é, que permite definir um padrão para estruturar a informação contida num documento.

Análise Comparativa entre XML e HTML

Hyper-Text representa informação que respeita um caminho ou uma estrutura que é definida pelo desejo (ou engenho) do utilizador. A informação é ligada e pode ser acedida de forma independente da sua localização e da localização do utilizador.

Partindo desta definição surge então a *Hyper-Text Markup Language* (HTML) que vem dar vida a este conceito, respeitando três requisitos fundamentais:

- ***Associação:*** a informação é completamente interligada, o que permite chegar a um bloco de informação a partir de outro.
- ***Simplicidade:*** é uma linguagem simples.
- ***Portabilidade:*** é uma linguagem multi-plataforma facilmente disseminada por vários tipos de redes.

O HTML veio permitir que quantidades massivas de informação pudessem ser disseminadas rapidamente à escala global, ultrapassando assim a barreira imposta até então pelo local e pela distância.

No entanto, rapidamente se começaram a verificar algumas carências e limitações no referente a:

- ***Inteligência:*** até que ponto a informação se conhece a ela própria?
- ***Adaptabilidade:*** com que facilidade se consegue adaptar a informação a constantes mudanças?
- ***Manutenção:*** com que facilidade se consegue manter e alterar a informação?

O HTML é desprovido de qualquer tipo de inteligência. Por exemplo, a linguagem é capaz de reconhecer um parágrafo ou uma imagem, mas é incapaz de reconhecer qual o parágrafo que está relacionado com determinada informação ou qual a imagem que a representa. A linguagem orienta-se essencialmente ao básico e não ao específico; não permite o acesso a uma pequena porção de uma página, ao invés da página por inteiro.

Em HTML, a semântica das *tags* e o conjunto de *tags* disponíveis são predefinidos, não havendo margem para qualquer extensão por parte do utilizador. Por exemplo, em HTML a *tag* <*h1*> é sempre um cabeçalho de primeiro nível e a *tag* <*Siemens*> não tem qualquer significado, reforçando assim a ideia da fraca adaptabilidade da linguagem.

O *World Wide Web Consortium* (W3C) [6] conjuntamente com os produtores de *browsers* e a comunidade *Web* trabalham constantemente na extensão da definição da linguagem HTML, procurando introduzir novas *tags* que permitam acompanhar a constante evolução tecnológica

e tornem cada vez mais apelativa a apresentação gráfica dos conteúdos. No entanto, estas alterações estão sempre limitadas às capacidades de interpretação dos *browsers*, existindo sempre o problema da compatibilidade das novas facilidades com as versões anteriores. Para qualquer pessoa que queira difundir rapidamente e de uma forma global determinado tipo de informação, novas extensões à linguagem que só são suportadas pelas últimas versões do *Netscape* ou do *Internet Explorer* não são de forma alguma úteis.

Em HTML os identificadores (*markup*) usados para distinguir entre imagens, texto ou *links* encontram-se todos misturados sem nenhuma divisão clara, tornando assim difícil qualquer tipo de alteração ou operação de manutenção.

Numa altura em que as páginas albergam cada vez mais conteúdos, torna-se complicado usar o HTML como linguagem de estruturação. É precisamente neste ponto que entra o XML, uma linguagem de estruturação que possui todas as "virtudes" do HTML e nenhuma das suas limitações.

Em XML tanto a semântica como conjunto de *tags* são conceitos completamente livres, aumentando assim a adaptabilidade da linguagem, uma vez que permite criar identificadores específicos para caracterizar informação específica.

Tal como foi referido na secção anterior, o XML é uma metalinguagem para descrever linguagens de estruturação, isto é, fornece um mecanismo que permite definir *tags* e relações estruturais entre elas. Esta característica traduz-se num aumento significativo do grau de estruturação de qualquer documento, aumentando assim a facilidade de manutenção do mesmo.

Apesar da grande liberdade de especificação que é oferecida pelo XML, qualquer aplicação necessita de um formato para que possa ler e interpretar a informação presente num documento de forma correcta. As *tags* não podem ser colocadas de qualquer forma, sob pena de o documento se tornar inútil.

Para que seja possível validar documentos e garantir que o seu significado não é adulterado, existe *Document Type Definition* (DTD), que permite precisamente definir restrições relativamente ao sequenciamento e encadeamento das *tags*. De uma forma geral, as declarações contidas num DTD representam meta informação que inclui as *tags* autorizadas, a sua sequência e grau de encadeamento, os atributos associados a cada *tag* e os seus tipos, etc.

Em suma, a grande vantagem do XML relativamente ao HTML reside na liberdade de especificação da linguagem e na capacidade de validação oferecida.

Resumo

A *Internet*, que começou por ser uma experiência realizada por um grupo de cientistas do ramo nuclear, representa hoje um papel de grande importância no nosso quotidiano. A sua utilização ultrapassa as fronteiras de meras páginas pessoais arrancando para outras áreas de interesse, como o comércio electrónico que muito veio revolucionar o contexto negocial.

A linguagem predominante no contexto *web* é sem dúvida o HTML *(Hyper Text Markup Language)*. No entanto, trata-se de uma linguagem pouco flexível e bastante fechada, o que dificulta a implementação de novas funcionalidades requeridas pela constante evolução das soluções negociais. O XML *(Extensible Markup Language)* surge precisamente para evitar esses pontos fracos.

Trata-se de uma linguagem de estruturação que permite definir padrões estruturais e formatos para qualquer tipo de documento. Do ponto de vista organizacional, este tipo de tecnologia representa uma mais valia, uma vez que permite definir padrões e normas *standard* para todo o tipo de informação usada.

Trata-se de uma linguagem que, para além da capacidade de associação, portabilidade e simplicidade (típicas do HTML), oferece ainda um certo grau de inteligência, adaptabilidade e facilidade de manutenção.

Em XML, a informação é representada recorrendo a *tags* (elementos) e atributos que permitem especificar a estrutura de um documento. A linguagem oferece ainda a possibilidade de validar formatos e conteúdos existentes nos documentos XML, usando *Document Type Definition* (DTD).

3.2 Common Object Request Broker Architecture (CORBA)

A rápida disseminação dos computadores e estações de trabalho, o aumento das suas capacidades de processamento e o aparecimento de redes de comunicação com grande largura de banda têm levado ao desenvolvimento, cada vez maior, de sistemas distribuídos. Este tipo de sistemas é, hoje em dia, de vital importância para o contexto organizacional, dada a necessidade existente de troca de informação constante, intra e inter organizações.

Os sistemas distribuídos apresentam características, entre as quais se destacam: disponibilidade, desempenho, optimização de custos, autonomia, mobilidade, heterogeneidade, afastamento, concorrência, falta de estado global, ocorrência de falhas parciais, assincronismo, etc. Para que seja possível lidar com todas estas características, diversos modelos e arquitecturas distribuídas têm sido desenvolvidos. No entanto, e tendo em conta a diversidade de características que estes sistemas apresentam, existe uma necessidade de criar especificações abertas, com *interfaces* padronizadas e públicas, levando assim à criação de *middlewares*[7] abertos.

O conceito de sistemas abertos aborda um número significativo de tecnologias e especificações, envolvendo também a questão da interoperabilidade entre vários sistemas baseados em padrões formais (mais fechados). Ao contrário dos sistemas proprietários, os sistemas abertos permitem ao utilizador escolher, de entre um vasto leque de tecnologias e plataformas, a que melhor se adequa às suas necessidades. Oferecem uma elevada interoperabilidade, escalabilidade e portabilidade, o que permite que diferentes sistemas coexistam juntos e que *software* desenvolvido numa plataforma possa ser executado em qualquer outra com adaptações mínimas.

No entanto, apesar das inúmeras vantagens, o conceito de sistemas abertos torna cada vez mais complexo o processo de desenvolvimento de novas tecnologias. A superação dessas dificuldades e a preservação das vantagens da computação heterogénea levou ao aparecimento de diversas organizações e consórcios, como a OMG *(Object Managment Group)* [7], que definiu a arquitectura CORBA.

---

[7] Um ***middleware*** é uma camada de *software* mediador existente entre a aplicação e o sistema de comunicações. Fornece abstracções de alto nível sobre a utilização dos recursos disponíveis, o que facilita bastante a programação distribuída, e gere a interacção entre várias aplicações, distribuídas por plataformas heterogéneas.

CORBA é uma arquitectura padrão que permite especificar um *middleware* (aberto) para sistemas compostos por objectos distribuídos. Possibilita a interacção entre um conjunto heterogéneo de objectos distribuídos.

Nas secções seguintes apresentam-se alguns aspectos mais relevantes da arquitectura CORBA.

Objectos Distribuídos

Um objecto distribuído é essencialmente um componente, uma peça de *software*, que pode interagir com outros objectos distribuídos através de sistemas operacionais, redes, linguagens, aplicações e equipamentos diversos.

Cada vez mais se verifica a tendência de construir sistemas computacionais abertos utilizando objectos distribuídos. Esta tendência verifica-se principalmente porque este tipo de sistemas oferece diversas vantagens sobre os sistemas tradicionais, entre as quais se destacam: a possibilidade de fornecer ao utilizador aplicações personalizadas, incluindo somente os recursos necessários e evitando assim o conhecido *fatware*, e a rapidez com que as arquitecturas das aplicações podem ser desenhadas usando pacotes de *software* existentes (e mais genéricos). Para além disso, os objectos distribuídos possuem exactamente as mesmas características principais dos objectos das linguagens de programação: encapsulamento, polimorfismo e herança, oferecendo dessa forma as mesmas vantagens desse tipo de objectos, como sejam a fácil reutilização, a manutenção e a depuração.

Em [Herbert 94] são realçados vários benefícios da utilização de objectos em sistemas distribuídos. Por exemplo, a utilização de objectos ajuda a lidar com a heterogeneidade de alguns sistemas, pois os serviços fornecidos são separados das suas implementações. Verifica-se também que o encapsulamento se torna vantajoso para implementar unidades de distribuição (que migram de forma independente), unidades de falha e de segurança. Ainda no mesmo texto são discutidas algumas dificuldades que se verificam ao nível da implementação da herança em sistemas distribuídos. A herança permite que diferentes objectos compartilhem código (implementação de outro objecto), o que oferece benefícios óbvios em termos de redução de duplicação de código. No entanto, em sistemas distribuídos, não é possível partilhar código de objectos que se encontrem em nós separados.

Um outro benefício que surge com a utilização de objectos distribuídos e que se aplica frequentemente (e especificamente) em sistemas de tempo real, é o polimorfismo de desempenho (ou polimorfismo temporal). Este mecanismo permite que uma interface possua várias implementações diferentes para o mesmo método, tendo cada uma o seu tempo de execução.

Cada objecto distribuído não trabalha sozinho. Normalmente os objectos são construídos para trabalhar com outros objectos e, para isso, precisam de uma espécie de barramento. Este barramento fornece uma infra-estrutura aos objectos que contém um variado leque de serviços que podem ser herdados em *runtime* para atingir altos níveis de colaboração entre eles. O CORBA representa precisamente esse barramento que é responsável por promover e facilitar a comunicação entre objectos independentes.

Arquitectura CORBA

A arquitectura CORBA começou a ser definida pelo OMG, como já foi referido, em 1989. A OMG é uma organização internacional, suportada por centenas de membros, que abrange um vasto leque de interesses relacionados com a indústria dos sistemas de informação. A carta de princípios desta organização inclui directrizes e especificações relacionadas com a gestão de objectos, que visam fornecer uma estrutura comum que pode ser usada no desenvolvimento de aplicações.

Em 1990, a OMG criou a OMA *(Object Management Architecture)* com o objectivo de fomentar o crescimento de tecnologias baseadas em objectos e fornecer uma estrutura conceptual de suporte para todas as especificações OMG.

O OMA constitui uma visão de alto nível de todo o ambiente distribuído e é composto por quatro componentes principais:

- ***ORB*** *(Object Request Broker)*: constitui a base de toda a OMA e gere toda a comunicação existente entre os seus componentes. O ORB possibilita a interacção entre os objectos num ambiente heterogéneo distribuído, independentemente do local em que os objectos se encontram ou das técnicas usadas para os implementar.
- ***Object Services***: uma colecção de serviços *(interfaces)* que suportam a execução de funções básicas relacionadas com a gestão de objectos como: criação de objectos, controlo de acesso, etc.
- ***Common Facilities***: conjunto de serviços partilhados entre aplicações.
- ***Application Objects***: conjunto de aplicações que seguem a arquitectura tradicional e, que por esse motivo, não são padronizadas pelo OMG.

Estes quatro componentes podem ser divididos em dois grupos: os *system oriented components* (que inclui o ORB e os *Object Services*) e os *application oriented components* (que inclui *Application Objects* e *Common Facilities*).

Na Figura 12 temos uma visão de alto nível dos elementos que compõe a OMA.

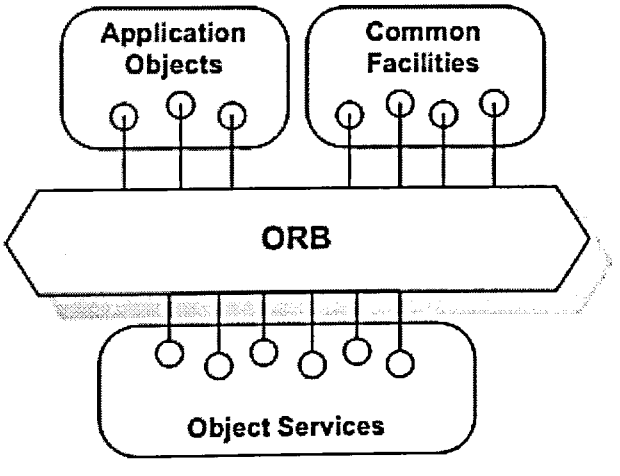

Figura 12: ***Object Management Architecture (OMA)***

OMG Object Model

Para que se possa compreender melhor a arquitectura CORBA, é importante e necessário conhecer o OMG *Object Model*, que define conceitos e terminologias relacionadas com os objectos usados pela arquitectura.

O conceito mais básico que se pode encontrar no *object model* é o conceito de *objecto*. Basicamente, um objecto é uma entidade que fornece serviços a clientes. Um *cliente* é representado por qualquer entidade capaz de requisitar serviços através de eventos denominados requisições *(requests)*. Toda a requisição possui informação associada que consiste basicamente na operação, o objecto destino, os parâmetros a usar e o contexto da requisição.

Os parâmetros são de entrada, de saída ou de entrada e saída e são identificados pela sua posição na requisição. O contexto da requisição destina-se a fornecer informação adicional sobre a requisição. Para além dos parâmetros de saída, as requisições podem também retornar os denominados valores de resultado ao cliente. No entanto, se ocorrer algum erro ou anomalia é gerada uma excepção.

Interface é uma descrição de um conjunto de operações que podem ser requisitadas por um cliente a um objecto. Diz-se que um objecto satisfaz uma interface quando pode ser apontado como objecto destino de qualquer operação descrita na interface.

Uma operação é uma entidade que representa um serviço que pode ser requisitado. Cada operação possui uma assinatura que, de um modo muito geral, descreve os valores válidos dos parâmetros e dos resultados retornados, a excepção definida pelo utilizador que pode ser usada para terminar uma requisição de operação e, finalmente, a informação de contexto que será fornecida à implementação do objecto. A assinatura descreve ainda a semântica de execução da operação em caso de ocorrerem erros ou falhas.

Na implementação de um objecto, um método é o código que é executado para fornecer o serviço e a execução de um método é denominada activação do método.

Object Request Broker (ORB)

Os clientes (objectos cliente) requisitam serviços às implementações de objectos através de um ORB – Figura 13. O ORB é responsável por suportar todos os mecanismos necessários para encontrar determinado objecto, preparar a sua implementação para receber uma requisição e executar a requisição. A requisição é vista pelo cliente de uma forma completamente independente da localização do objecto, da linguagem de programação usada para o implementar, ou de qualquer outro aspecto que não esteja incluído na sua interface.

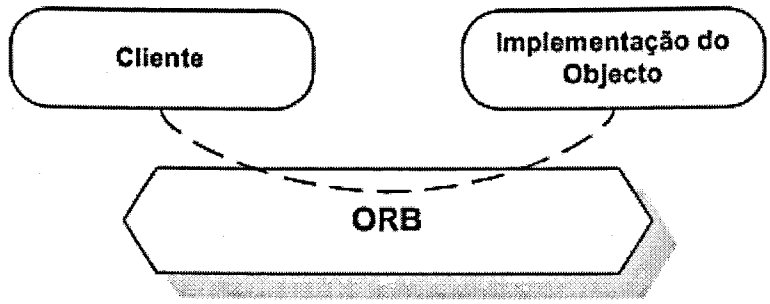

Figura 13: Cliente a enviar uma requisição através do ORB

O CORBA utiliza a OMG IDL *(Interface Definition Language)* para descrever interfaces. A OMG IDL é uma linguagem puramente declarativa (baseada em C++) o que garante que os componentes CORBA sejam auto documentáveis, permitindo assim que diferentes objectos, escritos em linguagens diferentes, possam comunicar e interagir entre si usando redes e sistemas operativos – Figura 14.

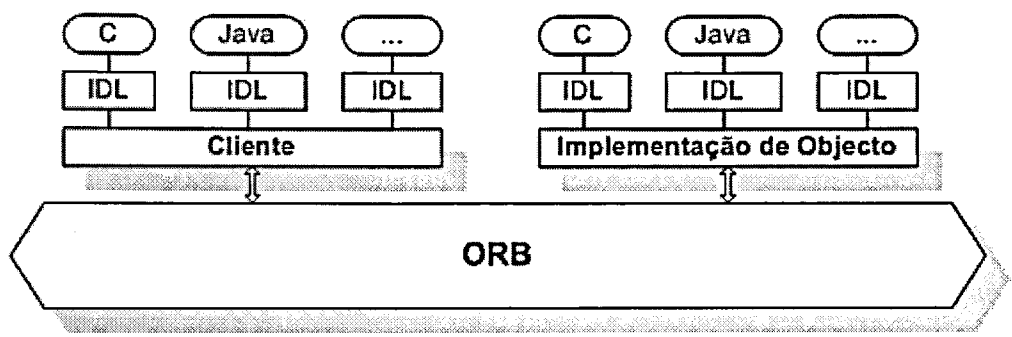

Figura 14: IDL e independência relativa à linguagem de programação

É importante salientar que os objectos não são escritos em OMG IDL, que é uma linguagem puramente descritiva como foi referido. Os objectos são escritos em linguagens de programação que possuem mapeamentos definidos para os conceitos existentes na OMG IDL, como C, C++, Smalltalk, Java, etc.

Estrutura da ORB

Na Figura 15 temos a estrutura básica de um ORB, onde as setas indicam o fluxo de chamadas que podem ocorrer dos clientes para o ORB e do ORB para as implementações dos objectos.

Para fazer uma requisição o cliente pode usar a *interface de invocação dinâmica (DII – Dynamic Invocation Interface)* ou então um *stub*[8] *de IDL* (invocação estática). Para determinadas funções (poucas) o cliente pode interagir directamente com a *interface* do ORB que disponibiliza funções que são independentes do adaptador de objectos utilizado, como, por exemplo, funções que executam operações sobre referências de objectos.

[8] Um ***stub*** não é mais do que um método (rotina) que na verdade não faz nada. O seu propósito é declarar-se a si próprio assim como os parâmetros que aceita. Normalmente usam-se stubs para definir métodos (rotinas) que precisam de ser implementados. Os *stubs* contêm apenas o código necessário para permitir a sua compilação e ligação com o resto do programa.

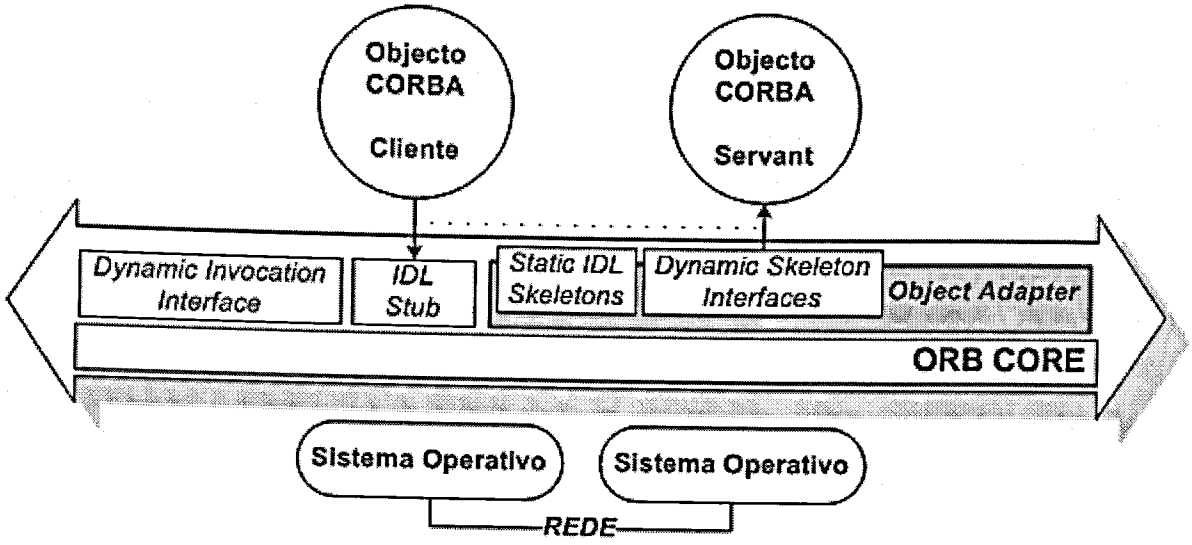

Figura 15: Estrutura básica de um ORB

O facto de permitir invocações estáticas e dinâmicas torna o CORBA muito flexível. No entanto, a invocação estática, possui uma série de vantagens sobre a invocação dinâmica, a saber: é mais fácil de programar, possui uma verificação de tipos muito mais robusta, é mais rápida em termos de execução e é auto documentável. Já a invocação dinâmica permite a adição de novos serviços à implementação do objecto sem alterações nos clientes, permitindo desta forma que os serviços disponíveis sejam descobertos em *runtime*.

A invocação dinâmica é feita usando os serviços do repositório de interfaces, que consiste, basicamente, numa base de dados que armazena as interfaces OMG IDL e disponibiliza serviços que permitem o acesso, armazenamento e actualização dessas interfaces.

Do ponto de vista da implementação do objecto *(servant)* a forma como a requisição é feita, estática ou dinâmica, é absolutamente transparente, uma vez que ambas respeitam a mesma semântica (uma referência para um objecto, uma operação e uma lista de parâmetros).

Para executar a requisição, a implementação do objecto pode requisitar alguns serviços do ORB usando o *object adapter*, que se situa no topo dos serviços de comunicação presentes no núcleo do ORB e fornece um ambiente que permite instanciar objectos, atribuir-lhes referências e enviar-lhes requisições. Com o *object adapter* é possível a uma implementação de um objecto ter acesso a um serviço independentemente de ele estar, ou não implementado no núcleo do ORB. Se o núcleo do ORB oferecer o serviço o *object adapter* fornece simplesmente uma *interface* para o mesmo, caso contrário, o *object adapter* implementa o serviço no núcleo.

Interoperabilidade entre ORBs

A obrigatoriedade de especificação das interfaces dos objectos usando OMG IDL garante a portabilidade dos objectos entre linguagens, aplicações, sistemas e redes diferentes. No entanto, a característica de interoperabilidade entre objectos só foi introduzida em 1994 na versão 2.0 do CORBA. Para que isto fosse possível foi especificada uma arquitectura de interoperabilidade, um suporte para pontes *(bridges)* entre ORBs, um protocolo genérico para comunicação entre ORBs e um protocolo para comunicação entre ORBs orientadas para a *Internet*.

A arquitectura de interoperabilidade do CORBA identifica claramente a existência de vários tipos de domínios de informação específica de ORBs, como: domínios de referências de objectos, domínios de tipos, domínios de segurança, etc. Quando dois ORBs estão no mesmo domínio então o tipo de comunicação é directo. No entanto quando, numa invocação, a informação tem que deixar o seu domínio a comunicação é mediada, uma vez que tem que atravessar uma ponte. Estas pontes asseguram o mapeamento correcto do conteúdo e da semântica da informação quando esta passa de um ORB para outro. Este suporte para pontes entre ORBs pode também ser usado para suportar interoperabilidade com outros sistemas que não sejam CORBA.

Serviços CORBA

O ORB fornece apenas os mecanismos básicos para suportar a interacção entre objectos, o que por si só não é suficiente. São necessários outros serviços, que a OMG vem padronizando para poderem ser utilizados pelas aplicações. Alguns exemplos são:

- ***Serviço de nomes:*** um serviço de localização de objectos que permite a um objecto descobrir outro através de identificadores ou nomes.
- ***Serviço de controlo de concorrência:*** um serviço de gestão de *locks* que coordena acessos concorrentes a recursos partilhados e que permite resolver conflitos de acesso.
- ***Serviço de eventos:*** serviço que permite aos objectos registar (manifestarem) dinamicamente o seu interesse em determinado tipo de eventos. Este serviço viabiliza uma comunicação menos centralizada e assíncrona entre objectos. Os eventos são enviados e recebidos através de um *canal de eventos*.
- ***Serviço de tempo:*** serviço que especifica uma *interface* que, basicamente, permite aos objectos das aplicações obterem o tempo actual e efectuarem comparações com o mesmo.

Resumo

Os sistemas distribuídos assumem cada vez mais um grau de elevada importância, principalmente devido ao crescente desenvolvimento da tecnologia (redes de comunicação de alta velocidade, capacidade de processamento, etc) que tem servido de base para o desenvolvimento de aplicações deste tipo.

A área de utilização destes sistemas facilmente ultrapassou o ambiente local, cruzando limites administrativos e tecnológicos e levando a uma grande necessidade de interoperabilidade e portabilidade. Estas necessidades aliadas à heterogeneidade dos equipamentos e sistemas operativos levaram ao aparecimento de especificações abertas, que permitem a padronização deste tipo de sistemas, assim como ao aparecimento de mecanismos que tornam completamente transparente o factor distribuição.

Os sistemas distribuídos são normalmente construídos com base no modelo cliente/servidor e usando mecanismos remotos de chamada de procedimentos, ao que a extensão ao paradigma de orientação a objectos trouxe inúmeras vantagens, como: reutilização, manutenção, etc.

O OMG, um consórcio formado por empresas e instituições de pesquisa, foi criado com o intuito de criar uma arquitectura padrão orientada para o mundo de objectos distribuídos. Esse padrão, denominado CORBA, é constituído por especificações abertas que facilitam a

portabilidade, e engloba considerações relativamente à interoperabilidade através da padronização de protocolos de comunicação entre os seus objectos.

A especificação da arquitectura CORBA está em constante evolução e novos serviços que vêm satisfazer requisitos específicos de determinados domínios aplicacionais, como aspectos de tempo real, têm sido bastante trabalhados nos últimos tempos.

4 Especificação do Módulo de GIS

Neste capítulo é apresentada a especificação do projecto de estágio. Inicialmente é feita uma alusão ao processo de desenvolvimento de *software* praticado na Siemens, seguindo-se depois a especificação do ambiente de desenvolvimento e do planeamento do projecto. Posteriormente é apresentada uma parte do projecto de alto nível – a especificação – onde se faz uma alusão aos requisitos definidos para o projecto e à arquitectura do protótipo. Para terminar, é feito um resumo do capítulo.

É importante salientar que, dada a complexidade do projecto e a quantidade de informação técnica existente, o presente documento faz uma descrição em termos gerais dos requisitos e da arquitectura da aplicação.

4.1 O Processo de Desenvolvimento de Software na SIEMENS

O projecto de estágio foi desenvolvido de acordo com o processo de desenvolvimento de software praticado na Siemens. Assim, torna-se relevante fazer uma breve descrição das etapas mais importantes do processo e das normas de qualidade usadas pela empresa.

Fase	*Objectivo*
Análise	Especificar as funcionalidades do sistema e subsistemas inerentes.
Desenho	Definir a arquitectura do sistema e subsistemas inerentes.
Implementação	Implementar o sistema e subsistemas inerentes.
Teste	Testar o sistema na sua globalidade, avaliando se está pronto para ser considerado um produto final disponível para entrega.

Tabela 1: Fases do processo de desenvolvimento de software SIEMENS

A fase de **análise** inicia-se com a elaboração de um Planeamento Preliminar do projecto. Este Planeamento Preliminar consiste na avaliação dos riscos associados ao projecto, na elaboração de uma estimativa de custos materiais e humanos, na definição de prazos a cumprir e finalmente na definição da equipa de trabalho. Mais tarde, quando se tem uma ideia mais concreta das reais necessidades é então elaborado o *Plano do Projecto*.

Nesta fase é elaborado um documento designado de *Requirements Specification (R-SPEC)* onde são descritas (sumariamente) as funcionalidades que o sistema deve oferecer e hierarquizados os requisitos, atribuindo um nível de prioridade a cada um e estabelecendo etapas (passos para a realização das diferentes tarefas). Tipicamente, estas etapas englobam requisitos de igual prioridade e com funcionalidades dependentes ou relacionadas.

Ainda nesta fase, é produzido um outro documento que descreve detalhadamente cada funcionalidade a implementar. O documento é designado de *Functional Specification (F-SPEC)* e serve de referência para as fases seguintes do processo. Este documento pode ter vários níveis, correspondendo cada nível ao grau de especificidade do mesmo. Isto é, se estão a ser especificadas as funcionalidades de um sistema de grande dimensão, fazem-se por exemplo dois níveis para a F-SPEC. No primeiro é especificado o sistema na sua globalidade e, no segundo, são especificados os diferentes subsistemas detalhadamente.

Na fase de **desenho** registam-se todos os aspectos relevantes associados à implementação, produzindo um documento designado de *Design Specification (D-SPEC)*, baseado na F-SPEC. Produzem-se por exemplo: diagramas de objectos, modelos dinâmicos, etc. O objectivo do documento é definir a arquitectura do sistema antes da implementação das funcionalidades.

Na fase de **implementação** e tendo como referência a D-SPEC, nomeadamente os diagramas construídos, produz-se o código necessário para implementar as funcionalidades especificadas na F-SPEC. Uma vez escrito o código necessário entra-se numa fase de depuração até ser alcançada uma versão estável da aplicação. Nesta fase é ainda redigido um documento designado de *Test Specification (T-SPEC)*, onde se descrevem os testes a efectuar ao sistema e as condições (de hardware, dados, etc) em que estes testes devem ser realizados.

Na última fase do processo, a fase de **teste**, executam-se os testes de acordo com o estabelecido na T-SPEC, corrigem-se os erros que possam, eventualmente, surgir e produzem-se novas versões da F-SPEC e D-SPEC de acordo com as alterações introduzidas no processo de correcção. Daqui resulta a *beta release* do produto, que corresponde a uma versão pronta para testes de integração (testes realizados no ambiente real, onde o produto vai ser utilizado). Ainda nesta fase é produzida a documentação destinada ao cliente: o *User Manual* e o *Instalation Guide.*

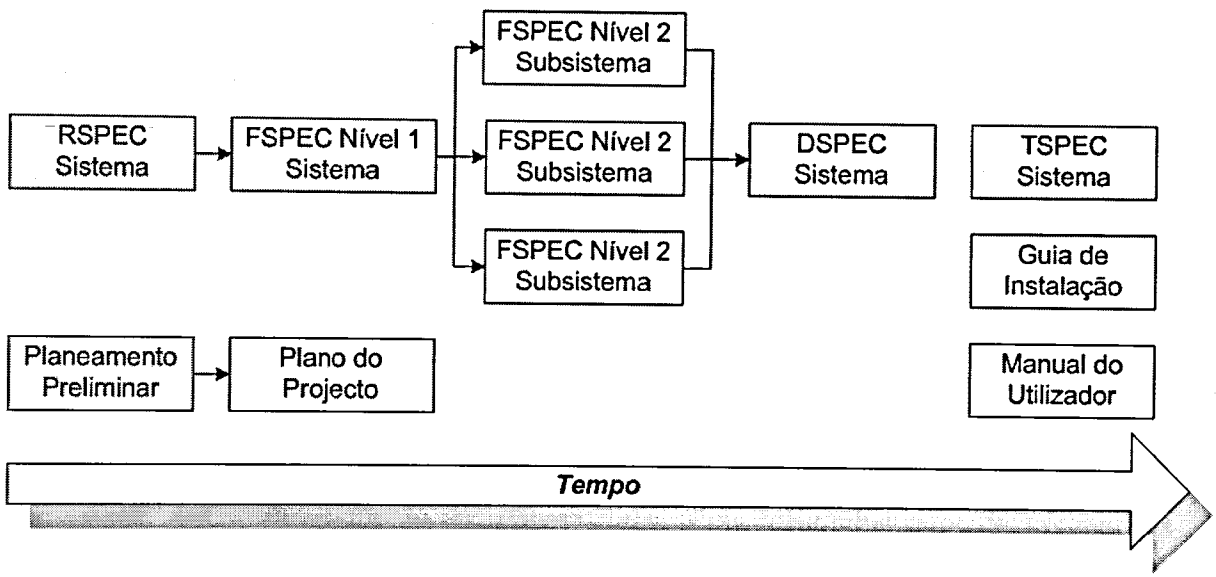

Figura 16: Processo de produção documental

Durante a execução das fases mencionadas é fundamental aplicar a política de qualidade definida na empresa.

4.2 Ambiente de Desenvolvimento e Planeamento

O plano de trabalhos a realizar durante o estágio foi elaborado pelo estagiário conjuntamente com o Eng.º Manuel Jorge Rodrigues, sendo analisado pelo Prof. Augusto de Sousa numa fase posterior. Na Figura 17 temos uma visão geral do plano de trabalhos, onde se especificam as *milestones* principais do projecto.

Foram atribuídas tarefas para um horizonte temporal equivalente a um semestre completo e decidiu-se que o estagiário seria responsável por todas as fases do processo de desenvolvimento e pela elaboração da R-SPEC e F-SPEC, excluindo a D-SPEC e a T-SPEC. Definiram-se ainda os requisitos e os prazos, intermédios e finais, a cumprir.

Os recursos materiais resumem-se a:

- Um PC para implementação do sistema.
- Um servidor de suporte disponível durante a fase de desenvolvimento e testes.
- Software para edição do código fonte.
- Software para compilação do código fonte.
- Software para produção da documentação.

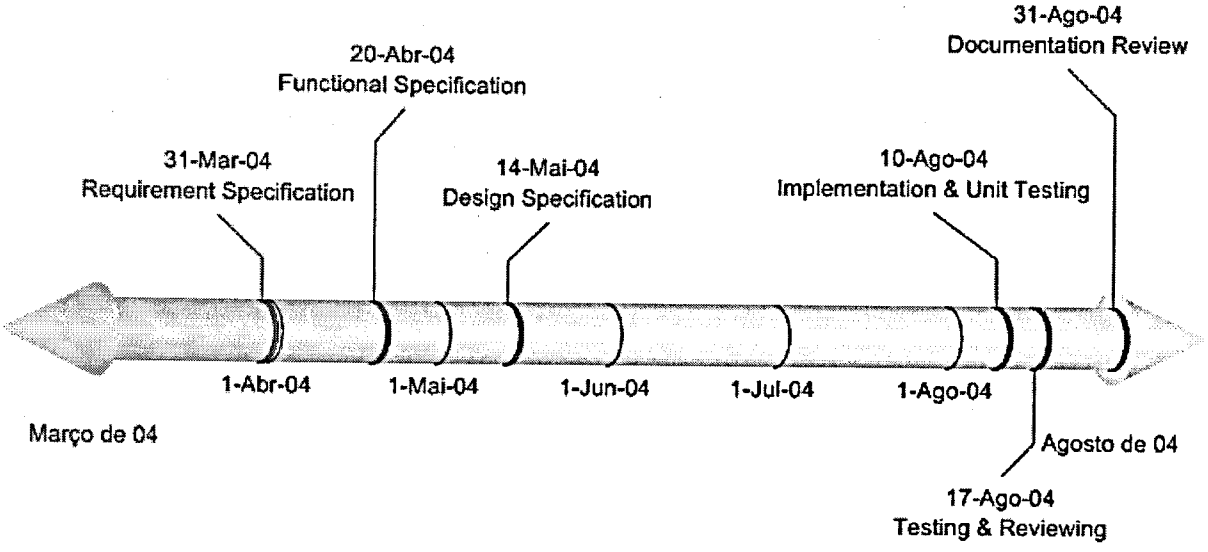

Figura 17: Plano de trabalhos agregado (*milestones* principais)

Recorrendo à Figura 17 é feita agora uma breve descrição do trabalho desenvolvido em cada uma das fases. Na fase de análise, foi feito um estudo dos sistemas de informação geográfica, onde se efectuou um levantamento das funcionalidades base. Simultaneamente, efectuou-se um estudo da aplicação SPOTS, definindo-se os pontos de integração com o novo módulo. Desta fase resultou a R-SPEC, que detalha os requisitos do sistema.

Posteriormente partiu-se para a fase de especificação, onde se definiram todas as funcionalidades e formas de operação da aplicação, tendo daqui resultado a F-SPEC. Terminada a especificação, partiu-se para a fase de desenho, onde se definiram os módulos que iriam constituir a aplicação, assim como a arquitectura de cada um deles. Na fase seguinte, de implementação e testes, desenvolveu-se o sistema e se testaram-se minimamente as suas funcionalidades. Na fase final foi dedicado algum tempo para a revisão de toda a documentação produzida no período de estágio.

O ambiente de desenvolvimento do projecto foi o seguinte:

- Implementação em linguagem Java (*Standard Edition* versão 1.4) e XML recorrendo ao *Eclipse* para edição do código fonte.
- Edição de documentos recorrendo ao *Microsoft Word 2002*.

- Utilização de um Pentium III a 450Mhz com 256Mb memória RAM, como máquina de desenvolvimento.

4.3 Conceitos

Antes de proceder à apresentação do projecto nas suas fases de análise de requisitos e definição de arquitectura, é importante introduzir alguma terminologia usada no contexto SPOTS que ajudará e facilitará a compreensão do resto do documento.

Contador: indicador (normalmente numérico) referente a determinado parâmetro que está associado a um dispositivo de rede.

Medida: conjunto de contadores que fornecem uma informação mais agregada relativamente a um aspecto específico associado a um determinado dispositivo de rede.

Relatório: no contexto SPOTS, um relatório agrupa um conjunto de medidas e contadores relativos a um determinado intervalo de tempo, apresentando-os em formato gráfico e/ou tabular. Os relatórios podem ser estáticos ou dinâmicos. Os relatórios estáticos apresentam resultados para um intervalo de tempo fechado, enquanto que os relatórios dinâmicos possuem apenas uma data inicial e vão sendo actualizados em intervalos de cinco, dez ou quinze minutos.

Alarme: no contexto SPOTS um alarme está sempre associado a um dispositivo de rede e é disparado quando um contador ultrapassa o seu valor limite. Os alarmes podem ser agrupados por severidade, a saber: *critical*, *major*, *minor*, *warning* e *undetermined.*

4.4 Análise de Requisitos

A análise de requisitos para o protótipo do GIS que irá integrar o SPOTS pode ser organizada em dois grupos:

- **Requisitos funcionais** *(requisitos ao nível de funcionalidades específicas do sistema a implementar)*
- **Requisitos não funcionais** *(requisitos ao nível da documentação, desempenho, etc)*

Requisitos Funcionais

Os requisitos funcionais do projecto podem ser divididos em duas categorias distintas: requisitos específicos de GIS e requisitos específicos do SPOTS e de sistemas adaptados às telecomunicações.

No âmbito de funcionalidades específicas de GIS, temos requisitos como: criar e eliminar mapas, guardar e abrir mapas, adicionar e remover *layers* dos mapas existentes, definir propriedades das *layers* e dos objectos nelas contidos, suportar informação geográfica em formato vectorial e em formato rasterizado, permitir operações gráficas básicas, como *zoom*, selecção e *pan*, suportar o cálculo de distâncias entre pontos e, finalmente, definir um indicador de escala sobre o mapa.

Relativamente aos requisitos específicos do SPOTS e de sistemas de telecomunicações, temos: adicionar e remover dispositivos de rede, suportar a geração de relatórios específicos de telecomunicações, suportar o processo de navegação entre vários relatórios existentes, mostrar informação relativa a alarmes no mapa de forma dinâmica, permitir seleccionar e

filtrar alarmes, permitir aceder a informação mais detalhada relativamente aos alarmes existentes e, finalmente, suportar informação de configuração diversa.

Requisitos Não Funcionais

Os requisitos não funcionais do projecto englobam unicamente as necessidades de documentação e optimização do desempenho do protótipo.

A documentação torna-se de vital importância devido ao facto de se tratar de um protótipo que poderá vir a ser usado como base para um módulo que será integrado numa aplicação comercial. Tendo em conta este factor, torna-se importante documentar de forma detalhada e fiel as características e as funcionalidades do protótipo, assim como a sua implementação, para facilitar trabalho futuro que eventualmente possa vir a ser realizado.

Os requisitos de optimização dizem respeito a cuidados relativamente à alocação de memória e outros recursos do sistema, procurando assim maximizar o desempenho do protótipo. Para este fim torna-se necessário verificar todo o código fonte da aplicação e corrigir eventuais falhas de alocação de memória e criação de *threads*.

4.5 Desenho

O modelo de arquitectura adoptado é uma simplificação do modelo da arquitectura integrada apresentado na secção 2.3. A opção de simplificação do modelo deriva do facto de se pretender, com o estágio, a construção rápida de um protótipo que permitisse avaliar o potencial de um módulo GIS no contexto SPOTS e a sua possibilidade de integração. Tendo em conta que a rapidez de desenvolvimento era um requisito optou-se por substituir o SGBDR, presente no modelo original, por uma solução de armazenamento baseada em ficheiros sem separação de conteúdos vectoriais, matriciais e alfanuméricos. Esta é, à partida, uma solução fraca, mas tendo em conta o contexto e o objectivo do estágio, torna-se a ideal, uma vez que reduz em muito o tempo de implementação. A possibilidade de integração de um SGBDR e a separação de conteúdos é real neste contexto e ficou registada em termos de perspectivas de desenvolvimento. Para além da remoção do SGBDR, optou-se também por eliminar a inclusão da linguagem de programação como interface alternativa de forma definitiva por não se enquadrar na filosofia da aplicação SPOTS.

Na Figura 18 temos o diagrama da arquitectura do protótipo. Podemos facilmente identificar grandes semelhanças com o diagrama da arquitectura integrada, como: a separação da interface com o utilizador do núcleo do sistema e a constituição modular da aplicação, que permite o desenvolvimento de componentes usando o conceito de *plug-in*. A grande diferença reside, como já foi referido, na inexistência de um SGBDR para gerir a informação alfanumérica. Neste modelo a informação alfanumérica, assim como a informação vectorial e rasterizada, é acedida directamente pelo gestor gráfico (identificado pelo nome ***Gi2*** no diagrama).

O Gi2, usado no contexto do projecto para fazer a apresentação da informação geográfica, é um componente proprietário da Siemens que foi utilizado e melhorado pelo estagiário para que pudesse ser integrado na aplicação.

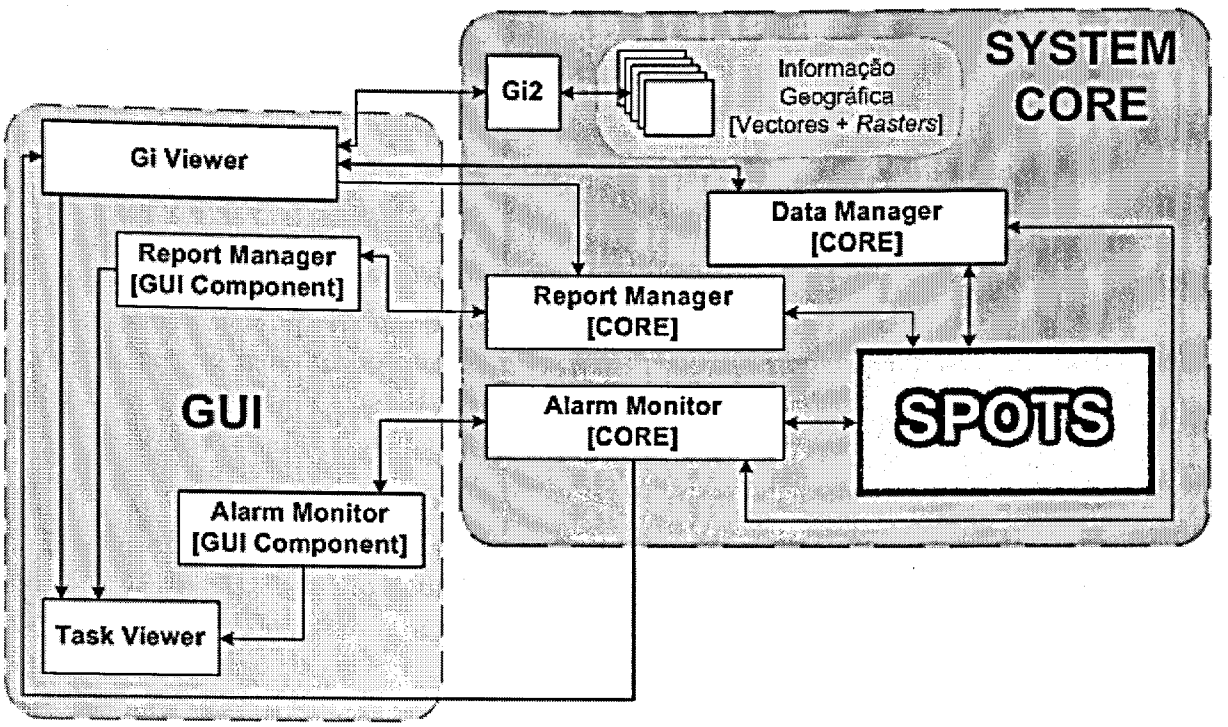

Figura 18: Arquitectura do protótipo

No âmbito do estágio, foram projectados de raiz cinco componentes essenciais para o funcionamento da aplicação, a saber: o *Gi Viewer*, o *Report Manager*, o *Alarm Monitor*, o *Task Viewer* e o *Data Manager*. Cada um deles foi pensado e desenhado para satisfazer e resolver alguns problemas que foram encontrados na fase de análise de requisitos. De seguida será feita uma breve descrição de cada um destes componentes, mencionando o objectivo para o qual foram pensados.

O ***Gi Viewer*** é o componente central da aplicação, uma vez que suporta a maior parte da interacção com o utilizador e é através dele que este tem uma percepção visual da informação geográfica e da informação relacionada com os dispositivos de rede (posição e estado).

O ***Data Manager*** representa um componente essencialmente orientado para a gestão e é responsável por gerir toda a informação manipulada pela aplicação (dados vectoriais, matriciais e alfanuméricos). A necessidade de integrar um componente deste tipo deriva da remoção do SGBDR para gestão de informação alfanumérica, tornando assim necessário arranjar uma forma de manter a coerência dos dados e evitar inconsistências que pudessem resultar em erros grosseiros.

O ***Report Manager*** surge para satisfazer alguns requisitos específicos da área de telecomunicações. Trata-se de um componente que se destina a gerir os relatórios predefinidos na aplicação SPOTS. Estes relatórios permitem obter informações relativas aos dispositivos de rede, são normalmente hierarquizados e organizados por grupos e fornecem informação que é frequentemente requisitada por engenheiros de telecomunicações. A ideia associada à introdução deste componente é fornecer ao utilizador uma forma de organizar os relatórios executados e em execução, permitindo assim um acesso mais rápido e mais fácil à informação desejada e sua visualização.

O ***Alarm Monitor***, tal como o *Report Manager*, é um componente que resulta da integração com o SPOTS e surge para satisfazer alguns requisitos específicos de telecomunicações.

Trata-se de um componente que foi desenhado com o intuito de fornecer ao utilizador uma forma de organizar toda a informação relativa a alarmes que vão surgindo e permitir também o acesso a informação mais detalhada relativamente a esses mesmos alarmes.

Finalmente, o ***Task Viewer*** surge com objectivo de fornecer ao utilizador informações (gráficas) relativamente ao estado das tarefas que vão sendo executadas. Esta necessidade surge porque grande parte das operações relacionadas com relatórios e alarmes são dependentes de um servidor e são executadas em *threads* independentes, o que pode gerar atrasos nos tempos de resposta da aplicação. Para evitar no utilizador, a sensação de que nada está a acontecer, foi introduzido este componente que o vai informando acerca do estado das várias tarefas.

4.6 Resumo

Esta fase do estágio decorreu de acordo com o processo de desenvolvimento de *software* praticado na Siemens. Primeiro definiram-se os requisitos a satisfazer, depois especificaram-se todas as funcionalidades a implementar, projectou-se a arquitectura das partes do sistema em que seria necessário trabalhar e finalmente passou-se à fase de implementação.

Durante a fase de análise foram identificados os requisitos funcionais e não funcionais do projecto. Rapidamente se conseguiu fazer um levantamento geral dos requisitos funcionais, tendo surgido duas categorias fundamentais, a saber: requisitos específicos de GIS e requisitos específicos do SPOTS e dos sistemas de telecomunicações.

A arquitectura adoptada segue uma versão simplificada do modelo integrado, descrito na secção 2.3. A simplificação foi introduzida essencialmente para permitir um processo de implementação mais rápido, uma vez que o objectivo do estágio era obter rapidamente um protótipo que permitisse analisar a viabilidade de integração de um módulo de GIS no SPOTS.

5 Implementação do Módulo de GIS

Neste capítulo apresenta-se o desenvolvimento do protótipo, fazendo uma descrição dos componentes e uma alusão aos problemas encontrados e correspondentes soluções implementadas. É introduzido o papel das tecnologias no contexto do protótipo e, uma vez apresentada, sucintamente, a fase de desenvolvimento, é apresentada, também de forma breve, a fase de testes do protótipo, terminando o capítulo com um pequeno resumo.

5.1 Componentes

Nesta secção é apresentada a fase de implementação dos componentes, fazendo alusão aos principais problemas encontrados.

Gi Viewer

O *Gi Viewer* – Figura 19 – como já foi referido, é o componente central da aplicação, uma vez que suporta maior parte da interacção com o utilizador e é através dele que este tem uma percepção visual da informação geográfica e da informação relacionada com os dispositivos de rede (posição e estado).

Este componente surge para solucionar a questão, colocada inicialmente, de como deveria ser apresentada ao utilizador a informação geográfica e de rede. Para este fim implementou-se então o *Gi Viewer* que é em essência um *bean* que integra vários componentes (entre os quais o *Gi2*) que permitem ao utilizador manipular e aceder à informação como entender.

O *Gi2* é, essencialmente, um painel de visualização ao qual foi associado uma zona de controlo. Trata-se de um painel de desenho *multi-threaded* que permite excelentes desempenhos relativamente às operações gráficas oferecidas (selecção, *pan e zoom*).

Torna-se importante salientar que, no âmbito do estágio, foram implementadas algumas extensões ao componente, nomeadamente ao nível da área de controlo, onde se fizeram melhoramentos ao nível da apresentação e funcionalidade e ao nível do painel de visualização que não estava preparado para suportar *pop-ups*, tendo sido implementado um pacote com essa funcionalidade.

No contexto aplicacional podem existir, simultaneamente, várias instâncias deste componente, correspondendo cada uma delas a um mapa diferente.

A área de controlo comporta uma árvore onde estão representadas as *layers* de informação do mapa e permite ao utilizador movimentá-las para cima e para baixo tornando-as mais ou menos visíveis mediante a opacidade das *layers* sobrepostas. Para além da árvore, existe também, nesta área de controlo, um painel de detalhes que exibe a informação disponível associada aos objectos seleccionados e que permite alterar algumas das suas características (por exemplo: visibilidade, cores de preenchimento, etc).

A implementação desta área de controlo surgiu da necessidade de oferecer ao utilizador uma forma de organizar os mapas e a informação contida em cada um. Para além disto, era ainda preciso fornecer informação relativamente às propriedades dos objectos.

Para além da área de controlo, o componente fornece ao utilizador um conjunto de ferramentas que este pode usar para executar algumas operações sobre o mapa. Estas

ferramentas englobam: selecção, *pan*, *zoom*, *undo*, execução de relatórios, etc. As ferramentas estão acessíveis ao utilizador através da *toolbar* do componente ou através de *pop-up* menus – Figura 19.

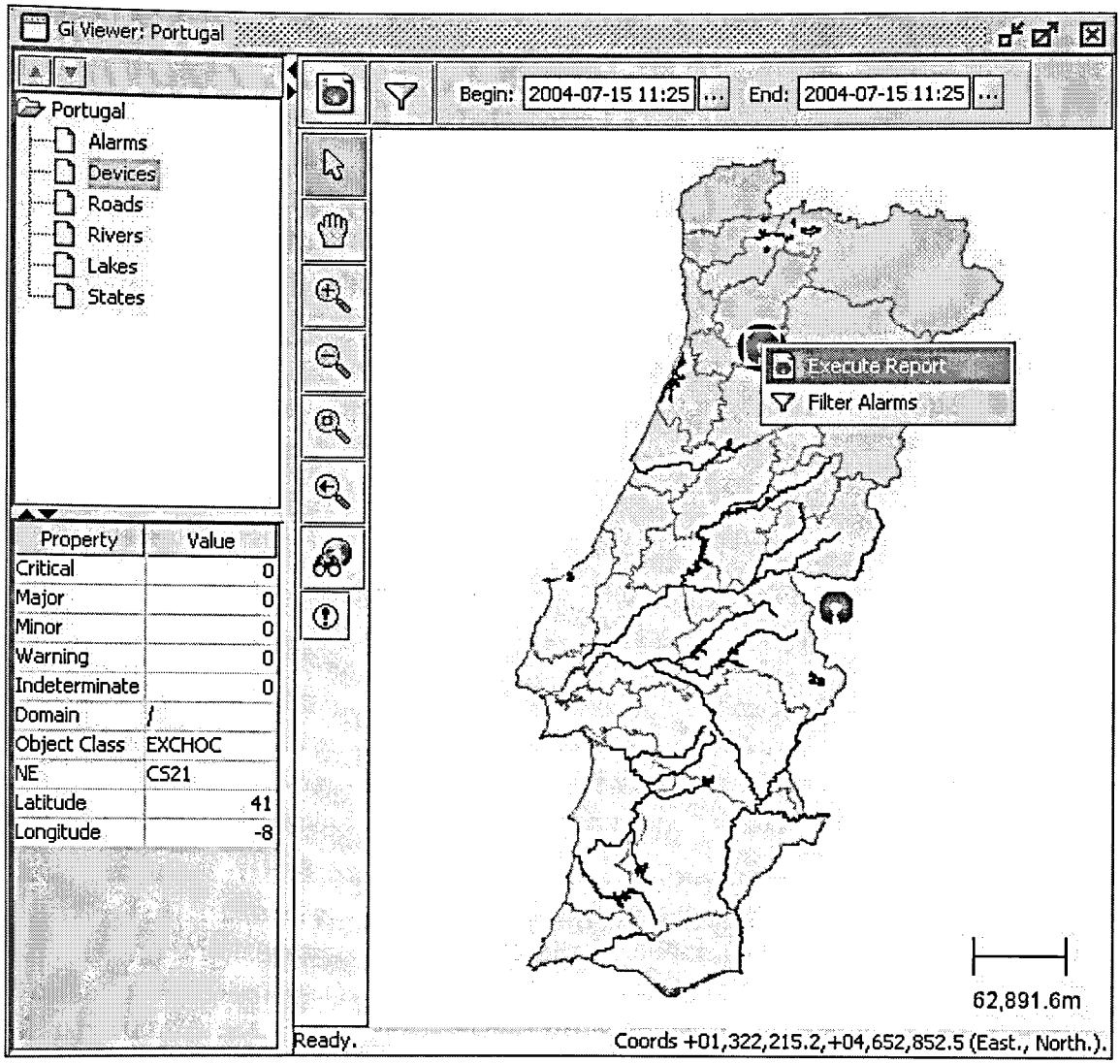

Figura 19: Gi Viewer

Um problema surgido aquando da criação deste componente foi o da localização geográfica dos dispositivos de rede. Nesta fase, a Siemens não possuía ainda acesso a ficheiros XML onde se pudesse encontrar a informação desejada – posições geográficas dos dispositivos. Para que, de alguma forma, se pudesse avançar foi especificado pelo estagiário um formato XML para o documento pelo estagiário e gerados dados fictícios. Para além do formato, foi implementado um pacote para leitura e escrita de documentos XML, tendo em conta que o formato definido é apenas um formato temporário, o pacote é genérico e facilmente adaptável a qualquer tipo de documento.

Data Manager

Com a remoção do SGBDR da arquitectura do protótipo tornou-se necessário arranjar uma forma de garantir a coerência de toda a informação aplicacional, isto para evitar

inconsistências nos dados que possam resultar em erros grosseiros – surge assim o *Data Manager*.

O *Data Manager* é reponsável pela gestão optimizada de toda a informação aplicacional, que contempla: dados vectoriais, matriciais e alfanuméricos, dados de dispositivos de rede, dados de alarmes e dados de relatórios. As suas funções incluem ainda a manipulação de toda a informação interna da aplicação e optimização da utilização dos recursos, garantindo um armazenamento eficaz e desprovido de redundâncias dos objectos em memória.

Trata-se de um componente vital para o funcionamento da aplicação, uma vez que todo o fluxo de informação passa por ele. À primeira vista pode não parecer uma solução muito boa, uma vez que estamos a introduzir uma centralização muito grande, mas, tendo em conta que o SGBDR foi removido do modelo, o desenvolvimento de um componente central tornou-se na alternativa mais viável. Os mecanismos de acesso e pesquisa fornecidos pelo componente foram altamente optimizados, recorrendo a um controlo rigoroso da utilização dos recursos de memória e usando *threads*, em pontos de processamento mais críticos, que permitiram obter níveis de desempenho mais elevados.

Este componente é completamente transparente para o utilizador, uma vez que não possui um módulo gráfico. Trata-se de um componente implementado unicamente ao nível do núcleo do sistema.

Report Manager

Este componente resulta de uma integração feita com a aplicação SPOTS, que fornece uma série de relatórios predefinidos que permitem obter informações relativas aos dispositivos de rede. Estes relatórios são normalmente hierarquizados e organizados por grupos e fornecem informação que é frequentemente requisitada por engenheiros de telecomunicações.

A ideia associada à introdução deste componente é fornecer ao utilizador uma forma de organizar os relatórios executados e em execução, permitindo assim um acesso mais rápido e mais fácil à informação desejada e à sua visualização. Para facilitar a utilização da aplicação toda a informação relativa a relatórios está centralizada neste componente, que se encontra dividido em duas partes fundamentais: *Report Browser* e *Report Display*.

O *Report Browser* é um navegador de relatórios que contém uma listagem de todos os relatórios executados ou em execução que ainda se encontrem presentes no contexto da aplicação e que podem ser acedidos em qualquer instante. Basicamente, este navegador consiste numa tabela onde cada linha representa um relatório particular e fornece informação relativamente: ao tipo de relatório, ao intervalo de tempo a ele associado, aos objectos que são por ele analisados, etc. A selecção de um relatório no navegador implica o seu aparecimento imediato na área de visualização *(Report Display)*, que será apresentada de seguida.

O *Report Display*, tal como o nome indica, é a zona de visualização de relatórios. Esta zona encontra-se dividida em duas partes. Uma apresenta os resultados do relatório em formato gráfico e a outra em formato tabular. Trata-se de uma área altamente parametrizável, uma vez que permite ao utilizador configurar quase todos os aspectos de visualização, como: esconder a tabela ou gráfico, definir escalas do gráfico apresentado, esconder medidas existentes no gráfico ou na tabela, ordenar a tabela por variados critérios, etc.

Na Figura 20 temos um *screenshot* do *Report Manager* com um relatório a ser exibido.

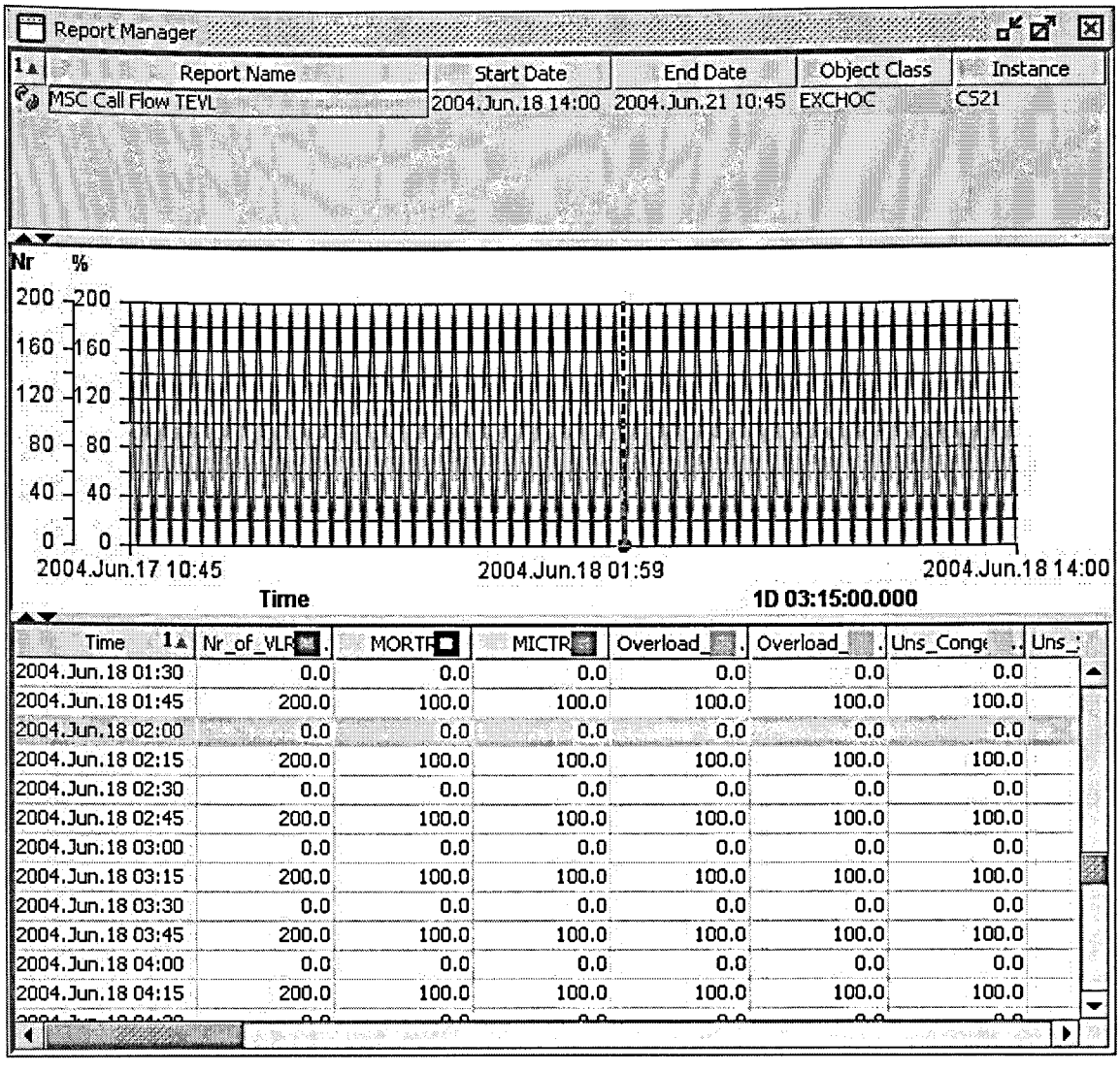

Figura 20: Report Manager

As funções de *reporting* podem ser acedidas pelo utilizador através do *Gi Viewer*, usando a barra de ferramentas ou os *pop-up* menus existentes neste componente e desenhados para esse fim. A informação dos relatórios é transmitida à aplicação via CORBA. Para que fosse possível receber os objectos CORBA desejados, foi necessário integrar um módulo específico da aplicação SPOTS no *Report Manager* e fazer algumas alterações para que este pudesse funcionar.

Alarm Monitor

Este componente, tal como o *Report Manager*, resulta de uma integração com o SPOTS, que tal como foi referido oferece um serviço de alarmes que disparam à medida que vão surgindo anomalias nos dispositivos de rede.

A ideia deste componente é fornecer ao utilizador uma forma de obter informação mais detalhada e mais específica relativamente aos alarmes que vão surgindo na rede. Para que se pudesse fornecer ao utilizador toda a informação necessária de uma forma agradável e, acima de tudo, organizada, optou-se por dividir este componente em duas partes fundamentais: o

Network Browser e o *Alarm Browser*. Na Figura 21 apresenta-se um *screenshot* do *Alarm Monitor* em funcionamento.

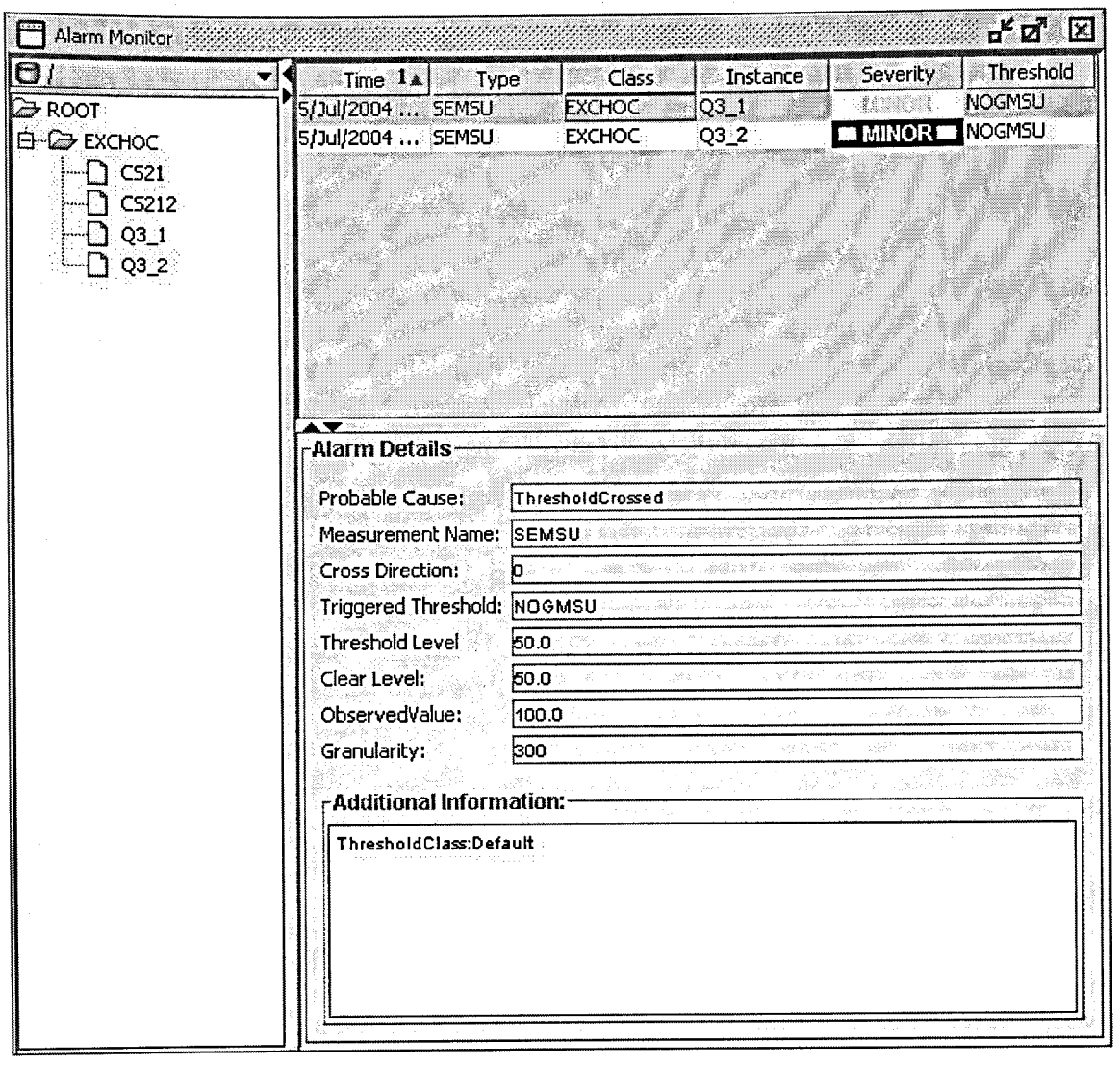

Figura 21: Alarm Monitor

O *Network Browser* é um navegador de dispositivos de rede. Neste navegador existe uma árvore de dispositivos organizada por tipos que permite ao utilizador filtrar rapidamente os alarmes (presentes na tabela), mostrando só os alarmes referentes ao dispositivo seleccionado. O navegador inclui ainda um selector de domínio que permite ao utilizador actualizar a árvore de dispositivos mediante o domínio de rede seleccionado.

O *Alarm Browser* é um navegador de alarmes que se encontra dividido em duas secções. Na secção superior encontra-se uma tabela que permite seleccionar e visualizar os alarmes existentes e que fornece informação geral relacionada com esses mesmos alarmes. Esta informação engloba: instante temporal em que o alarme disparou, o tipo de alarme e respectiva severidade, o dispositivo de rede a que está associado e o seu tipo e a medida ou contador a que o alarme se refere. Na secção inferior existe informação detalhada do alarme seleccionado na tabela. Esta informação detalhada é útil, na medida em que representa uma ajuda preciosa para a compreensão de alguns problemas menos óbvios.

Sobre a tabela de alarmes é ainda possível aplicar os mais variados tipos de filtros, o que permite ao utilizador procurar alarmes seguindo vários tipos de critérios.

O *Alarm Monitor*, tal como o *Report Manager*, recebe a informação desejada via CORBA. Também no *Alarm Monitor* componente foi integrado o módulo CORBA específico do SPOTS.

Task Viewer

Este componente tem como objectivo fornecer ao utilizador informações (gráficas) relativamente ao estado das operações que vão sendo executadas. Esta informação torna-se importante no contexto da aplicação uma vez que grande parte das operações executadas implica a ligação a um servidor remoto, estando assim sujeitas a atrasos e erros que podem despertar no utilizador sensações de inactividade por parte da aplicação. Com a introdução do *Task Viewer*, o utilizador tem a percepção de que a aplicação está a executar o seu pedido através da informação visual que lhe é dada pelo estado da tarefa.

O componente é essencialmente uma tabela que se encontra dividida em três colunas onde são introduzidas as diferentes tarefas (uma linha por tarefa). A primeira coluna contém informação relativa ao estado da tarefa, que é dada através de um pequeno icon. Em cada instante o estado da tarefa pode assumir um de cinco valores possíveis, a saber: executada, cancelada, falhada, a executar ou executada com erros. A segunda coluna contém informação relativa ao instante de tempo em que a tarefa foi requisitada pelo utilizador e, finalmente, a terceira coluna, contém uma descrição breve e sucinta da tarefa – Figura 22.

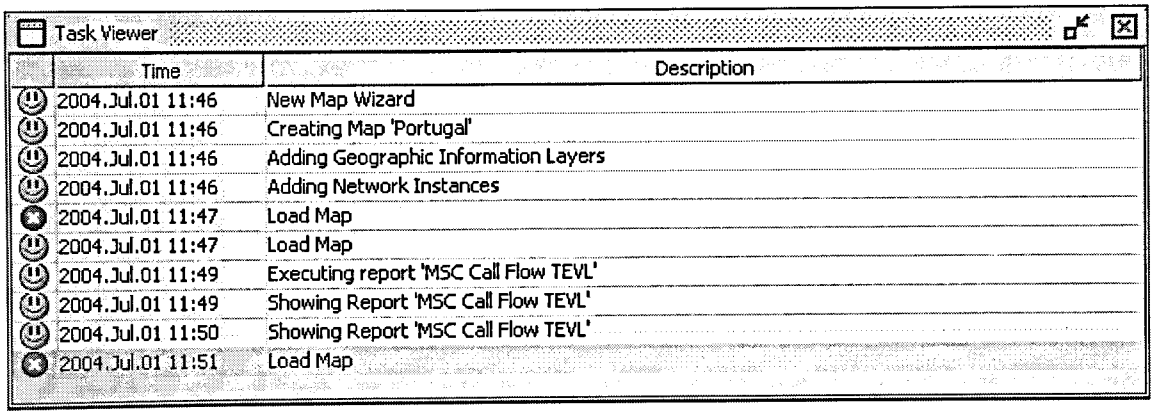

Figura 22: Task Viewer

5.2 Testes

Os testes efectuados ao protótipo foram divididos em três tipos: *unit, module* e *integration*. Na primeira fase (*unit testing)* testaram-se as funcionalidades de cada módulo individualmente. Na segunda fase agruparam-se as funcionalidades de cada módulo e fizeram-se os *module tests*. Finalmente, na terceira fase integraram-se os diferentes módulos e realizaram-se os *integration tests*.

Tendo em conta que, com este projecto, se pretendia apenas avaliar a possibilidade e a facilidade de integração de um módulo de GIS na aplicação SPOTS, concluiu-se que não seria necessário fazer testes muito exaustivos ao protótipo, uma vez que a sua arquitectura teria que

sofrer algumas alterações antes de ser usado numa versão comercial. Nesta secção do documento serão apenas mencionados de uma forma geral os resultados dos testes obtidos para cada um dos componentes.

Relativamente à componente geográfica *(Gi Viewer)* verificou-se um excelente desempenho do gestor gráfico que responde de uma forma muito rápida e eficaz aos pedidos do utilizador. No entanto, foram detectados pequenos erros ao nível do cálculo das transformações geométricas que convertem as coordenadas dos dispositivos de rede, provocando ligeiros desvios relativamente às coordenadas reais[9].

Relativamente ao *Data Manager* verificou-se a razoabilidade da solução para o contexto do estágio, uma vez que permitiu acelerar o processo, garantindo a coerência da informação. No entanto, quando a quantidade de informação a gerir aumenta muito, o desempenho sofre substancialmente, afectando essencialmente o componente gráfico (o que faz mais pedidos ao *Data Manager*). Para contornar esta situação teria que ser implementada uma solução com um SGBDR externo.

Relativamente ao *Alarm Monitor* e ao *Task Viewer* não se detectaram nenhumas anomalias. Os componentes corresponderam da forma esperada e com o desempenho desejado.

Quanto ao *Report Manager* verificaram-se alguns problemas relacionados com falhas do servidor que ainda não eram muito bem tratadas pelo protótipo. O componente demonstrou um elevado desempenho e tornou patente a utilidade de mostrar os resultados do relatório sincronizados, em formato gráfico e tabular.

No respeitante aos testes de integração, pode referir-se que, de uma forma geral, a aplicação correspondeu bem. No entanto, existem ainda alguns erros de código que terão que ser tratados num futuro próximo.

5.3 Resumo

A fase de desenvolvimento do protótipo resumiu-se à implementação e teste dos componentes especificados.

Começou-se por desenvolver o *Gi Viewer*, um componente de visualização de informação geográfica, que permite ao utilizador ter uma percepção visual dos dispositivos de rede, englobando posição e estado.

Posteriormente passou-se para a implementação do *Data Manager*, que é o gestor de informação e optimizador da aplicação. Este componente surge porque na fase especificação se optou por remover o SGBRD do modelo da arquitectura. Com esta remoção tornou-se vital implementar um componente que garantisse a coerência da informação aplicacional.

Prosseguiu-se com a implementação do *Report Manager* e do *Alarm Monitor*. O componente *Report Manager* representa um navegador de relatórios que permite ao utilizador organizar todos os relatórios lançados e visualizá-los, quando desejar, de uma forma rápida e eficaz. Por sua vez, o *Alarm Monitor*, corresponde a um componente de monitorização de alarmes que permite ao utilizador obter informação mais detalhada relativamente aos alarmes que vão surgindo na rede.

---

[9] Tratam-se de desvios quase insignificantes mas que poderão não ser admissíveis numa aplicação comercial.

Finalmente, passou-se para a implementação do *Task Viewer*, que é um componente que fornece ao utilizador informações gráficas relativamente ao estado das tarefas que vai realizando no contexto da aplicação, por exemplo: abrir um mapa, executar um relatório, adicionar um dispositivo de rede, etc.

Uma vez terminada a fase de implementação passou-se para a fase de testes, de onde se obtiveram, em termos gerais, resultados positivos. Desta fase concluiu-se que apesar do protótipo não estar ainda devidamente preparado para integrar uma versão comercial do SPOTS, constitui uma boa base para a criação de um módulo de GIS inteiramente compatível.

6 Conclusões

Neste capítulo do documento são apresentadas as conclusões finais do projecto de estágio. Inicialmente são feitas algumas considerações relativamente ao projecto de estágio, passando-se posteriormente para algumas considerações de carácter mais geral. Finalmente são apresentadas algumas considerações referentes a perspectivas de desenvolvimento da aplicação.

6.1 Considerações Relativas ao Projecto

Os objectivos do estágio, apresentados neste documento, podem ser divididos em três partes. Primeiro era necessário implementar rapidamente um protótipo que permitisse analisar a possibilidade de integração de um módulo de GIS na aplicação SPOTS, assim como analisar a sua viabilidade. Em segundo lugar, era importante introduzir na empresa um maior *know-how* relativamente aos sistemas de informação geográfica. Finalmente, tendo em vista a integração do estagiário no ambiente de trabalho Siemens, era preciso que o estagiário compreendesse e aplicasse o processo de desenvolvimento de *software* utilizado internamente.

Depois de um período de familiarização com o estado da arte relativo a sistemas de informação geográfica e com a aplicação SPOTS, deu-se início ao projecto propriamente dito. Foram definidos os requisitos, especificaram-se as funcionalidades, projectou-se e implementou-se o sistema e, finalmente, efectuaram-se os testes.

Os resultados obtidos permitiram concluir que a integração de um módulo de GIS na aplicação SPOTS é viável e representa uma mais valia face à concorrência. No entanto, o protótipo, que actualmente funciona como uma aplicação *standalone*, teria que ser remodelado em alguns aspectos para que a sua integração fosse um sucesso completo.

Para a realização do projecto foi utilizado o processo de desenvolvimento de *software* praticado na Siemens. Este processo incutiu no estagiário um grande sentido de responsabilidade e melhorou as suas capacidades metodológicas, uma vez que foi exigido um grande nível de rigor nas tarefas a executar.

Quando o estagiário chegou à empresa com a intenção de começar o projecto, beneficiou de "total" liberdade na definição do protótipo, o que aliás representou um factor determinante na escolha da Siemens como empresa de estágio em detrimento de outras propostas. Outro dos aspectos que muito agradou ao estagiário durante os últimos seis meses, está relacionado com o facto das tarefas a realizar não terem sido impostas mas sim sugeridas. A opinião do estagiário foi ouvida e debatida em todas as situações em que isso se impunha.

Por todos estes motivos e, tendo em conta as impressões recolhidas junto das pessoas que acompanharam o projecto, pode-se concluir que este foi um sucesso.

6.2 Considerações Relativas a Perspectivas de Desenvolvimento

Tal como foi referido anteriormente, o objectivo do estágio era desenvolver rapidamente um protótipo que permitisse avaliar o potencial de um módulo GIS integrado no SPOTS e a sua viabilidade. Tendo em conta estes factores e a necessidade de uma implementação rápida, foi necessário simplificar a arquitectura do protótipo, reduzindo assim um pouco o leque de funcionalidades possíveis.

Um dos melhoramentos que poderá ser implementado está relacionado com a questão da base de dados geográfica. A inclusão de um sistema de gestão de bases de dados externo possibilitará um maior desempenho e maiores garantias de segurança e integridade dos dados, assim como permitirá introduzir funcionalidades de pesquisa muito mais avançadas.

Tendo em conta que se trata de uma aplicação orientada às telecomunicações e permite um grande nível de detalhe (podemos chegar ao pormenor de uma rua com este protótipo), teria uma grande utilidade introduzir um mecanismo de pesquisa baseado em conceitos geográficos. Por exemplo, imaginando um cenário de reclamação relativa à fraca qualidade da rede em determinada localidade, bastaria, com um mecanismo de pesquisa baseado em conceitos geográficos, introduzir no sistema a localidade em causa e este imediatamente apresentaria um mapa da localidade e respectivos dispositivos associados. Isto permitiria ao operador verificar rapidamente as causas do problema e movimentar esforços para a sua resolução.

Outra extensão que poderá ser introduzida na aplicação é a possibilidade de o utilizador executar relatórios parametrizáveis, isto é, ter a possibilidade de definir as medidas e os contadores que pretende incluir no relatório. Para implementar esta funcionalidade, uma das possibilidades seria construir um pequeno editor que, baseado na *meta data* do SPOTS, permitiria ao utilizador editar medidas e contadores na especificação do relatório.

Outra extensão de grande utilidade seria dar a possibilidade de o utilizador de definir acções que seriam executadas sempre que chegasse um alarme de determinado tipo. Por exemplo, sempre que surgisse um alarme crítico relacionado com uma antena em determinada zona, enviar um *e-mail* urgente para determinada pessoa.

Referências e Bibliografia

[1] Câmara, Gilberto; *et al;* 1996 *"Anatomia de Sistemas de Informação Geográfica"*, Instituto de Computação da UNICAMP

[2] *Natural Resources Management Gateway (Section – Glossary)*
http://corpslakes.usace.army.mil/employees/glossary.html

[3] *Getting to Know Desktop GIS*
http://www.gis.com/resources/library/dtgis/front.html

[4] *O'Reilly XML.com – XML from the inside out*
http://www.xml.com

[5] *CERN – Europe Organization for Nuclear Research*
http://public.web.cern.ch/public/

[6] *World Wide Web Consortium*
http://www.w3.org

[7] *Object Management Group*
http://www.omg.org

Outra Bibliografia

Capital Region USA Section – History
http://www.capitalregionusa.org/portugal/history/history.html

Extensible Markup Language (XML) Specification
Section 4.7 – Notation Declarations
http://www.w3.org/TR/REC-xml#Notations

ESRI – GIS and Mapping Software
http://www.esri.com/index.html

USGS – Geographic Information Systems
http://erg.usgs.gov/isb/pubs/gis_poster/

[Herbert 94] A. Herbert, *Distributed Objects*, In Distributed Open Systems Edited by F.M.T. Brazier and D. Johansen, IEEE Computer Society Press, 1994

Extensible Style Language (XSL)
http://www.w3.org/TR/WD-xsl

Extensible Markup Language (XML) 1.0
http://www.w3.org/TR/WD-xml

BIBLIOTECA F.E. PORTO

FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
BIBLIOTECA
0000081525

004(
EIC52