

AMBIENTE PARA APRENDIZADO A DISTÂNCIA: ESTUDO DE CASO EM PLANEJAMENTO DE SISTEMAS MÓVEIS CELULARES

André M. L. Miranda – andremoacir@unama.br

Universidade da Amazônia, PROEAD – Programa de Educação a Distância.

Avenida Senador Lemos nº 2809, Sacramento.

66.120-901 – Belém – Pa.

Carlos R. L. Francês – rfrances@ufpa.br

João C. W. A. Costa – jweyl@ufpa.br

Universidade Federal do Pará, Laboratório de Eletromagnetismo Aplicado.

Rua Augusto Corrêa nº 01, Guamá.

66.075-900 – Belém - Pará.

Resumo: *Um desafio para o oferecimento de educação continuada é a disponibilização mais universal de diversos aplicativos de suporte à aprendizagem, para serem utilizados dentro das estruturas curriculares dos cursos de engenharia. Uma alternativa é disponibilizar as ferramentas em portais disponíveis na Web, de forma que estas estejam disponíveis em quaisquer hora e lugar. Entretanto, há a dificuldade de congregar em um portal diversos aplicativos que foram desenvolvidos em diferentes linguagens, sistemas operacionais e arquiteturas de computadores, gerando via de regra incompatibilidade binária entre os aplicativos. Este trabalho apresenta uma proposta de abordagem para gerar interoperabilidade entre softwares, de modo a permitir a integração entre essas diversas soluções já disponibilizadas, intra ou inter instituições. A proposta apresenta como se pode realizar essa integração utilizando-se tecnologias como UML, MVC, CORBA e SOAP para aplicativos desenvolvidos para planejamento de sistemas móveis celulares. Essa solução de integração pode ser estendida a qualquer área de conhecimento, e não somente à engenharia, por se tratar de uma estratégia genérica para integração de sistemas/aplicativos legados e universalização de acesso.*

Palavras-Chave: *Educação a distância, Integração de sistemas, Objetos distribuídos, CORBA, Sistemas móveis celulares.*

1. INTRODUÇÃO

O crescimento do volume de informação, a geração de novos produtos e o advento de novas teorias têm acarretado reavaliações do processo de trabalho e exigido soluções com diferencial de qualidade e prazos curtos para que as empresas possam sobreviver à competitividade. Essa nova realidade empresarial define o bom trabalhador como sendo aquele que aprenda de forma não convencional, e que saiba trabalhar cooperativamente para gerar soluções inovadoras e competitivas, MAIA (2000). Analisando este cenário, percebe-se que o ensino convencional não tem dado condições aos indivíduos de se prepararem no tempo adequado a essas transformações, pois o mercado de trabalho atual demanda profissionais com constante atualização dos conhecimentos e com habilidade para resolver problemas complexos através da cooperação com outros profissionais.

Portanto, faz-se necessário uma nova abordagem na formação e na atualização profissional, fazendo com que as pessoas retomem o trabalho, mas continuem sempre se capacitando. A solução é um novo modelo de formação: aprendizado cooperativo em ambientes ligados em rede e tempos virtuais. Aprender, deixa de ser um processo que ocorre num período definido, para ser uma atividade com a duração direcionada pelo aluno. A tecnologia dos ambientes em rede é um meio para facilitar a interação social, viabilizar a aprendizagem individual, através das interações com um grupo e para possibilitar a criação coletiva de um conhecimento compartilhado, MAIA (2000).

Outra motivação desse trabalho é a constante evolução tecnológica, pois é um fator que contribui para a fragmentação de sistemas em qualquer corporação, fazendo com que as mesmas se deparem com falta de integração entre seus sistemas legados, CUMMINS (2002). Em especial, na área educacional é comum encontrar aplicações legadas (como por exemplo, simuladores) em diversas instituições de ensino que poderiam ser utilizadas de forma integrada a outras aplicações com o propósito de formar uma grande solução computacional para um determinado problema.

Esse aspecto é marcante em engenharia, área na qual inúmeros softwares já estão desenvolvidos para importantes aplicações, nas mais diversas plataformas de programação, por diversos grupos de pesquisadores do Brasil.

Uma forma de tratar esse problema seria reescrever todo o código das aplicações legadas, elegendo linguagens e sistemas operacionais que possam perdurar; porém isso demandaria muito tempo e dinheiro, além de desprezar o *know-how* já estabelecido para as aplicações em uso. Uma possível alternativa, que é a apresentada nesse artigo, seria adicionar entre as aplicações uma camada intermediária chamada *middleware*, a qual seria responsável por tratar toda a heterogeneidade do ambiente, fazendo então com que os mesmos conseguissem interagir.

Em função desse contexto, este trabalho propõe uma abordagem para tratar o problema de heterogeneidade de aplicativos, linguagens de programação e sistemas operacionais, elaborando-se uma estratégia para integrar sistemas legados com o enfoque de educação a distância no domínio da área de engenharia elétrica. Desta forma, são apresentados os aspectos fundamentais referentes à integração de três sistemas legados e à execução dos mesmos via *Web*. Esses sistemas têm a função de realizar algumas tarefas referentes planejamento de sistemas móveis celulares, e são módulos de um sistema maior chamado CELLP, CAVALCANTE (2002), que foi desenvolvido na UFPA.

Originalmente, os módulos que compõem o CELLP não interoperavam e eram executados em um ambiente totalmente heterogêneo. A fim de possibilitar a integração entre esses módulos, utilizou-se a arquitetura CORBA (*Common Object Request Broker Architecture*) para “mascarar” toda a heterogeneidade e complexidade do ambiente, pois CORBA é uma arquitetura que possui vários recursos para integrar sistemas legados, juntamente com ampla variedade de hardwares e softwares heterogêneos, COULOURIS e DOLLIMORE *et al* (2001).

Para que isso fosse possível, houve a necessidade de utilizar a UML (*Unified Modeling Language*) para fazer a engenharia reversa do CELLP, baseando-se no padrão de projeto MVC (*Model View Controller*), a fim de facilitar o processo de integração dos três sistemas e facilitar também a faturação dos mesmos em alguns objetos distribuídos. Esses processos, posteriormente, agilizaram as modificações necessárias no código. Como esse sistema será utilizado no âmbito da educação a distância, é essencial que ele possa ser executado via *web*, e para que isso fosse possível utilizou-se o protocolo SOAP (*Simple Object Access Protocol*). Ao adicionar ao CELLP uma nova filosofia multicamada, implementou-se a versão WebCELLP, MIRANDA (2003). Desta forma, o ponto-chave da proposta está na concepção e implementação do *middleware* (discutido na próxima seção).

No estudo de caso foi criado um portal para *Web*, o qual disponibiliza, de forma prática e didática, um referencial teórico sobre planejamento de sistemas móveis celulares e a possibilidade de execução do sistema WebCELLP. Esse portal servirá de apoio para a disciplina de Sistemas Móveis Celulares do curso de Engenharia Elétrica da UFPA. O portal tem como seus principais objetivos:

Em relação ao ensino-aprendizagem:

- Facilitar o aprendizado, através de uma interface interativa, reforçando o conhecimento adquirido na sala de aula tradicional;
- Disponibilizar o conhecimento para aprendizado e revisões complementares das disciplinas envolvidas;
- Disponibilizar um protótipo para o uso da comunidade acadêmica e empresarial, deixando-o preparado para o acoplamento de novos módulos;
- Possibilitar, por um período de tempo integral, a utilização do portal, levando em conta a disponibilidade e o ritmo do aluno;

Em relação aos sistemas utilizados:

- Possibilitar a integração de sistemas que foram desenvolvidos em linguagens de programação diferentes;
- Possibilitar a interoperabilidade de sistemas operacionais e arquiteturas de hardware diferentes;
- Disponibilizar a utilização do CELLP em formato “universal” de navegação, via *Web*;
- Ampliar a tolerância a falhas do sistema, pela adoção da arquitetura de três camadas.

2. *MIDDLEWARE*

Middleware é a camada de software situada entre o sistema operacional e os componentes de aplicação que tem como finalidade facilitar a comunicação e a coordenação desses componentes que estão distribuídos nos diversos computadores de uma rede, EMMERICH (2000). Portanto, caso seja desejado que um determinado aplicativo possa ser utilizado em uma quantidade razoável de máquinas e sistemas operacionais, uma possível alternativa é o uso de *middleware*.

Como o propósito do WebCELLP é executar em um ambiente distribuído, portanto heterogêneo, foram criados cinco objetos que são gerenciados por dois servidores que executam em hardwares e sistemas operacionais diferentes. Para que fossem geradas as implementações dos cinco objetos especificados no *middleware* CORBA, juntamente com seus *stubs* e *skeletons* (elementos que encapsulam a comunicação) foram definidas quatro IDLs (*Interface Definition Language*), as quais foram submetidas a dois compiladores diferentes. Assim, quando um cliente emitir uma requisição será utilizada uma chamada remota (forma de comunicação entre objetos) para invocar os métodos dos objetos.

Essa requisição é enviada através de uma mensagem para o servidor que gerencia o objeto. O servidor ao receber a requisição, ativa o objeto para que o mesmo possa executar o método solicitado. Quando o objeto tiver posse do resultado, ele informa o servidor, que por sua vez retorna esse resultado ao cliente em uma outra mensagem.

3. MODELAGEM DO SISTEMA

Dos três módulos que compõe o WebCELLP, dois (o *software* Predict e o Traffic Design) foram desenvolvidos no Delphi 7® e executam sobre o sistema operacional Windows®, e o terceiro (o *software* Radio Link) foi desenvolvido no Kylix 3® e executa sobre o sistema operacional Linux®.

O módulo Predict foi desenvolvido principalmente para realizar estudos estatísticos comparativos entre os modelos clássicos de rádio-propagação, CAVALCANTE (2002). A “Figura 1”, mostra uma janela desse *software*, a qual é dividida nas seguintes abas:

- Parâmetros: nela o usuário deverá fornecer os dados referentes aos parâmetros do sistema (potência transmitida, frequência, etc) que foram utilizados na campanha de medições de referência. O usuário também deve selecionar nesta aba, os modelos que serão colocados sob avaliação assim como os parâmetros urbanos necessários para a aplicação dos mesmos;
- Base de Dados: nela o usuário deverá fornecer a base de dados das campanhas de medições realizadas na região de interesse, ele terá também a opção de editar os dados e depois salvá-los em um arquivo;
- Análise Gráfica: nela são gerados gráficos da potência recebida (dBm) versus distância (m), baseando-se nos modelos colocados sob avaliação e nos dados da campanha de medições;
- Análise Estatística: nela são apresentados os desempenhos dos modelos testados frente às medidas. O desempenho é especificado sobre três aspectos: Desvio Absoluto (dB), Erro Médio Absoluto (dB) e Erro RMS (dB). O usuário ainda tem a opção de gerar um relatório apresentando os resultados obtidos e de salvar esses resultados em um determinado projeto.

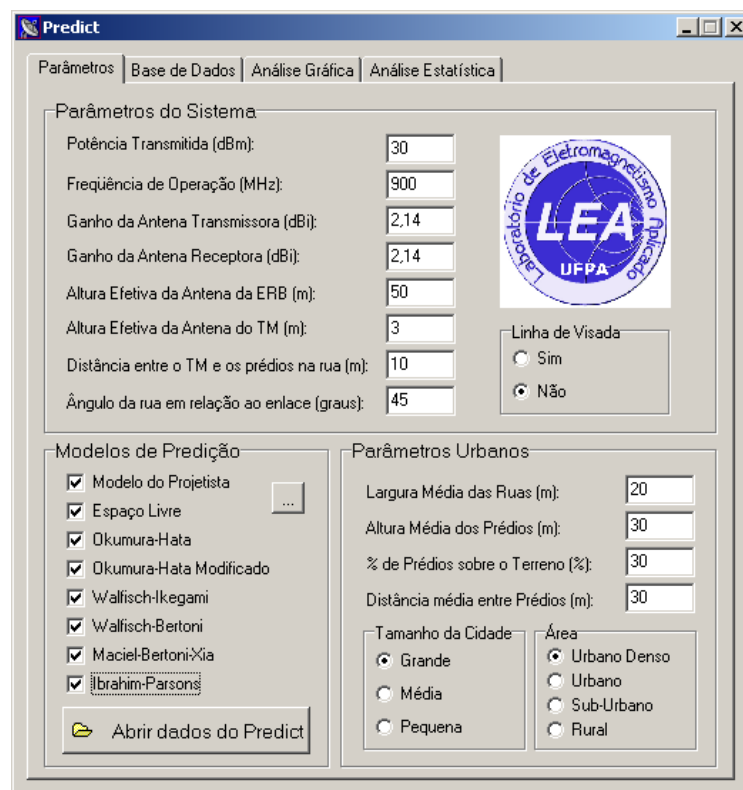


Figura 1 - Janela inicial do Predict.

Já o *software* Radio Link foi desenvolvido para realizar o balanço de potência do sistema, determinando o raio máximo de cobertura de um determinado modelo para os dois enlaces básicos (descida e subida). A janela desse módulo é dividida em duas abas, uma denominada “Dados de Entrada”, onde o usuário fornecerá os parâmetros do sistema e escolherá o modelo que deseja realizar o estudo, e outra denominada “Resultados”, onde o usuário poderá

verificar os resultados dos cálculos realizados, gerar relatório contendo os resultados obtidos e salvar esses resultados em um determinado projeto.

O *software* Traffic Design realiza o projeto de tráfego baseado em um mapa que demonstra o tráfego previsto para uma determinada região. O mapa é dividido em quadriculas que representam o tráfego estimado na área delimitada pela mesma. Nesse módulo o usuário define alguns parâmetros do mapa e o raio das células, para que seja determinado o número de canais necessários para atender tal tráfego. O usuário ainda tem a possibilidade de gerar um relatório do resultado dessa contagem e de salvar esse resultado em um determinado projeto.

Na “Figura 2” é apresentado um caso de uso que mostra a interação entres os módulos do WebCELLP. Inicialmente, o usuário deverá visitar um portal que contém um material teórico sobre planejamento de sistemas móveis celulares e um *link* para a execução do sistema. Ao clicar nesse *link*, o usuário deverá informar seu *login* de acesso, para que seja feita a sua validação, e então o usuário poderá executar qualquer módulo do sistema.

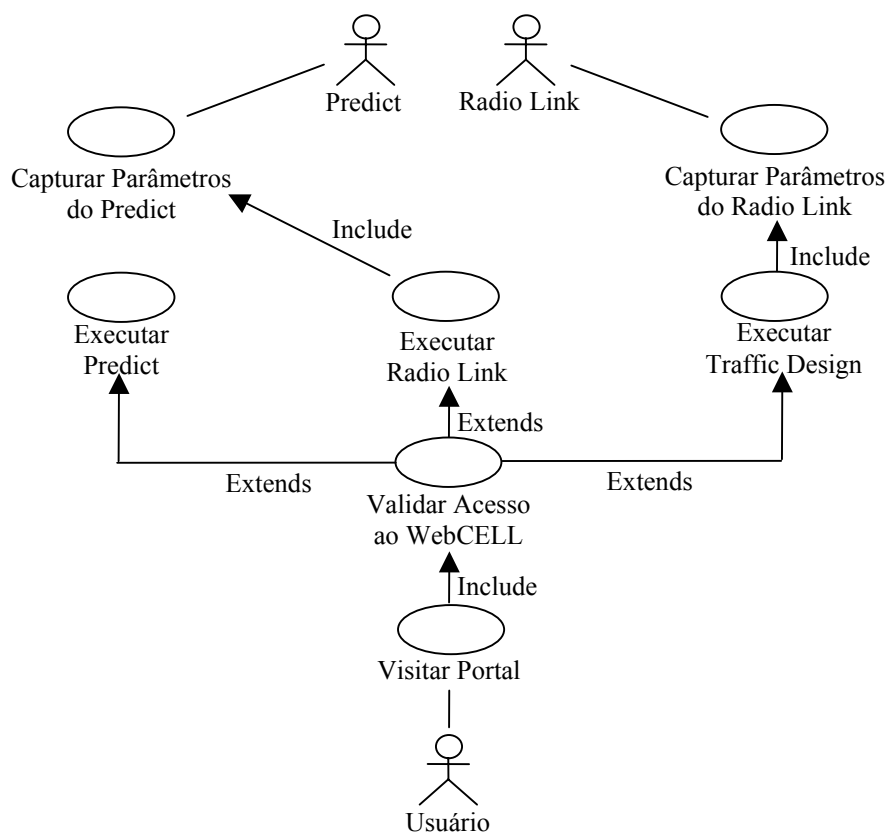


Figura 2 - Caso de uso geral do WebCELLP.

Após a validação, caso o usuário escolha executar o módulo Predict, ele terá como resultado a definição do modelo de predição ideal para a região em estudo. Caso o usuário decida executar o módulo Radio Link, será necessária uma interação com o módulo Predict, para que seja feita a captura de alguns parâmetros e somente após isso é que será feito o cálculo do raio máximo de cobertura da célula.

Na execução do Traffic Design também será necessária uma interação para capturar parâmetros, porém com o módulo Radio Link, para então ser feito o cálculo da contagem de tráfego na célula.

4. MVC

A escolha do padrão de projeto MVC deu-se pelo fato dele ser ideal para projetar aplicações *Web*, BROWN *et al* (2001). Durante a fase de implementação, é importante deixar registrado os seguintes aspectos referentes ao MVC:

- Na Vista (interface gráfica do sistema) utilizou-se a interface gráfica já existente no *software* legado, fazendo-se algumas adaptações, como a retirada das regras de negócios da própria interface.
- No Modelo, onde estão armazenadas as regras de negócios, construíram-se dois servidores de aplicação, um implementado em Delphi 7® e outro em Kylix 3®, sendo que em ambos foram criados objetos utilizando os recursos de programação distribuída do padrão CORBA. Além dos servidores de aplicação, criou-se também um banco de dados no InterBase 6.5®, o qual armazena além das tabelas algumas regras de negócios.
- O elemento que faz o papel de Controlador é o próprio CORBA, que serve de intermediador entre o cliente e os objetos do Modelo.

Na “Figura 3”, pode-se visualizar o modelo MVC do WebCELLP.

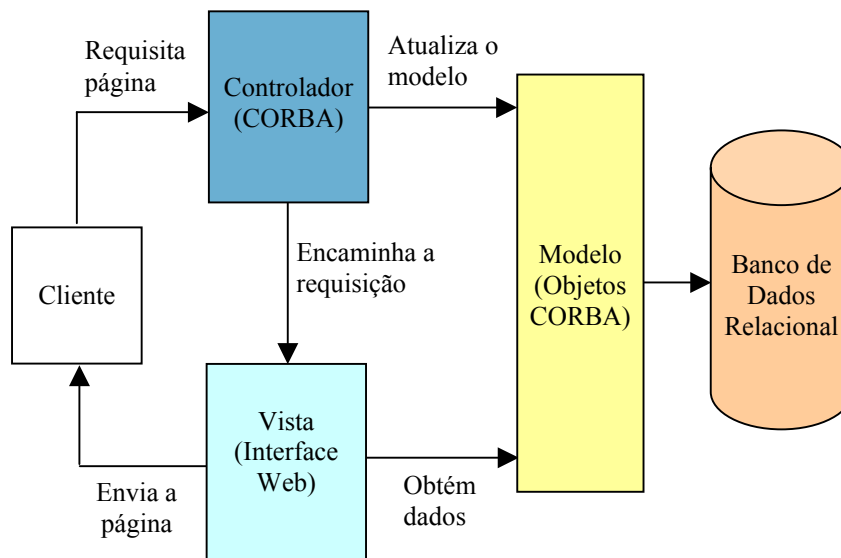


Figura 3 - Modelo MVC do WebCELLP.

5. PLATAFORMA HETEROGÊNEA

No desenvolvimento do WebCELLP foram utilizados dois ambientes de programação visual que usam a linguagem de programação *Object Pascal*: o Delphi® e o Kylix®, que foram construído para serem executados no sistema operacional Windows® e Linux®, respectivamente. Através do Delphi/Kylix pode-se desenvolver desde aplicações genéricas, que não acessam banco de dados, até aplicações Cliente/Servidor com n-camadas que integram várias linguagens, como é o caso desse sistema.

A “Figura 4” apresenta a arquitetura desenvolvida para que o WebCELLP seja acessado via *Web* e opere sobre diversas plataformas e sistemas operacionais. Para isso necessitou-se utilizar os recursos de computação distribuída do Delphi/Kylix, os quais permitem a criação de aplicações com n-camadas, que são compostas por um conjunto de componentes e outras tecnologias como CORBA, DCOM, SOAP e TCP/IP, permitindo assim a construção de uma variedade de aplicações com acesso a banco de dados.

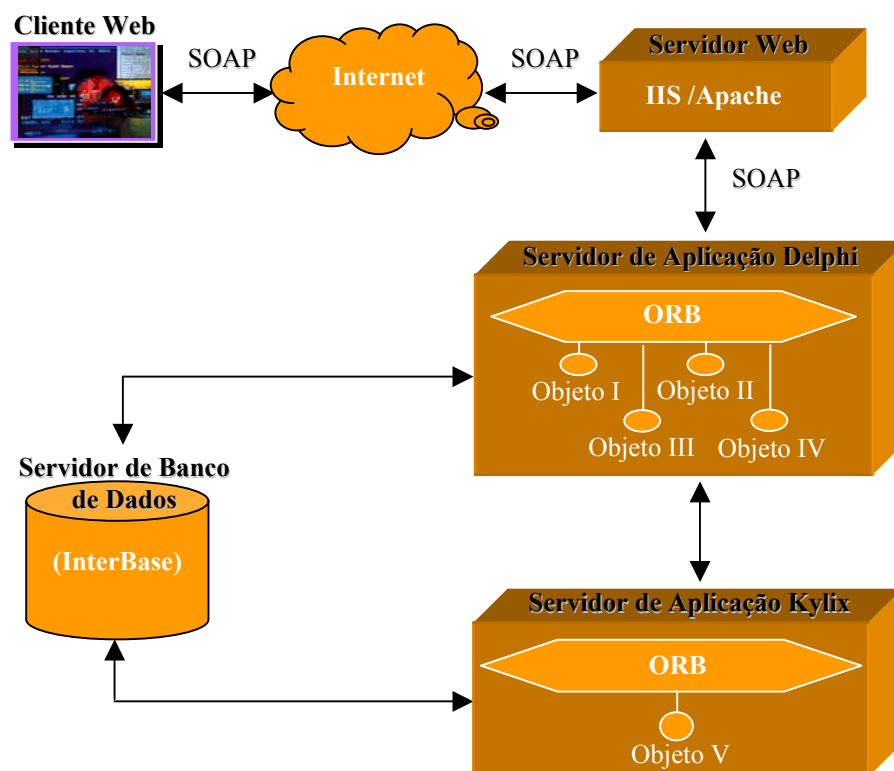


Figura 4 - Arquitetura do WebCELLP.

Especificamente no sistema em questão, utilizou-se o CORBA para prover a comunicação entre a aplicação cliente e a aplicação servidora. Porém, para ser flexível o bastante a ponto de operar sobre o protocolo HTTP, houve a necessidade de utilizar a tecnologia *WebServices*, baseado no protocolo SOAP.

Sendo assim, o cliente *Web* faz a requisição através do SOAP, que encaminha a requisição ao respectivo objeto CORBA no servidor, para que sejam obtidos os dados do banco de dados. De posse desses dados, o servidor os envia para o cliente no formato XML, conforme é mostrado na “Figura 4”. Essa divisão de responsabilidade permite que o cliente visualize, pague, busque e ordene os dados, localmente, atualizando a tela conforme necessário sem fazer novas conexões ao servidor.

Analisando a “Figura 4”, percebe-se que no Servidor de Aplicação Delphi estão quatro objetos CORBA pertencentes ao Predict e ao Traffic Design, e que no Servidor de Aplicação Kylix está um objeto pertencente ao Radio Link. Percebe-se também nos dois servidores de aplicação a presença do ORB (*Object Request Broker*), que é o núcleo da arquitetura CORBA, e tem com principais funções: localização do objeto, ativação do objeto se necessário, envio da requisição do cliente para o objeto e envio da resposta do servidor para o cliente, OMG (2002).

Em relação à base de dados remota, pode-se afirmar que os acessos a essa base poderiam ser mais rápidos, caso fossem feitos diretamente ao banco de dados. Porém, por questão de segurança, os acessos foram feitos pelos objetos CORBA a um servidor remoto, pois esses objetos têm implementado todos os métodos necessários para acessar a base de dados do sistema. Outra vantagem, além da segurança, é a reutilização e a padronização dos códigos de acesso à base de dados, facilitando futuras manutenções.

Outro aspecto relevante neste trabalho é o fato de que a maioria das aplicações *Web* baseadas em dados, sofrem com a baixa velocidade de comunicação. Isso se dá pelo fato de que cada requisição do cliente ao servidor precisa conter todas as informações de estado, que

são necessárias para satisfazer tal requisição. Portanto, o servidor precisa executar uma nova consulta à base de dados, ao invés de reutilizar os conjuntos de resultados previamente estabelecidos.

No WebCELLP tal fato não ocorre porque, utilizando-se CORBA, pode-se fazer com que as aplicações *Web* melhorem o desempenho e o tempo de resposta ao cliente, reduzindo-se o número de requisições que cada cliente faz ao servidor e a quantidade de trabalho que o servidor precisa executar.

Uma outra solução para realizar a integração do CELLP que foi citada neste artigo, seria reescrever todo o código da aplicação, porém com certeza seria muito mais trabalhosa.

Os comentários finais que devem ser pontuados com relação a esta proposta são com relação a possíveis dificuldades que podem ser encontradas para replicá-la:

- Fatoração dos módulos do CELLP em objetos distribuídos: os módulos do CELLP operavam isoladamente, com nenhum grau de integração. Precisou-se então fatorá-los em objetos CORBA, deixando-os inclusive prontos para serem reutilizados;
- Configuração do sistema operacional: para possibilitar a comunicação do CORBA com o Linux® foi preciso configurar algumas variáveis de ambiente do Linux®. Portanto, num projeto como este é necessária a participação de profissionais que dominem tanto a área de sistemas operacionais quanto a de redes de computadores;
- Configuração do Kylix®: foi preciso também incluir alguns arquivos do Kylix® no Linux®, para que fosse possível a comunicação do CORBA com o servidor de aplicação Kylix.

6. CONSIDERAÇÕES FINAIS

Um ganho proporcionado por este trabalho foi a mudança na forma de ensino da disciplina Sistemas Móveis Celulares, dentro do curso de Engenharia Elétrica da Universidade Federal do Pará. Ao longo dos últimos anos, esta disciplina vem sendo ministrada de forma teórica e totalmente presencial, o que limitava em parte um aprendizado mais abrangente. Com o desenvolvimento do WebCELLP, surgiu a possibilidade de transformar parte da carga horária teórica em prática, além da possibilidade de torná-la semi-presencial. Segundo a legislação vigente, pode-se ter até 20% da carga horária em regime não presencial, o que, via de regra, ainda não é utilizado de maneira eficiente em cursos de Engenharia.

Assim, atualmente, estão sendo utilizadas duas abordagens: presencial e a distância. Na parte presencial é apresentado o conteúdo teórico da disciplina, como já vem acontecendo ao longo dos anos; e na parte a distância será utilizado o sistema WebCELLP, como ferramenta de apoio, para que o aluno possa aplicar os conhecimentos adquiridos em sala de aula, através da criação de um projeto de planejamento de sistema móvel celular.

Neste artigo foram apresentados alguns pontos relevantes a aplicações distribuídas e integração de sistemas legados, procurando viabilizar a interoperabilidade entre ambientes heterogêneos, o que se constitui em um problema do estado-da-arte de corporações de diferentes setores e, em especial, a área de educação.

Através do projeto e da implementação do WebCELLP, foi possível se efetivar uma solução viável para problemas de integração de sistemas legados heterogêneos, aplicando-se a estratégia baseada em middleware, no domínio de ensino de engenharia elétrica.

Além da utilização do sistema, há alguns outros ganhos indiretos na utilização dos princípios de engenharia de software para desenvolver sistemas de software. Por exemplo, percebeu-se que através do padrão de projeto MVC, facilmente identificaram-se as classes e instâncias participantes, seus papéis, colaborações e a distribuição de responsabilidade. A utilização desse padrão tornou a aplicação menos complexa e agilizou a fase de

implementação, pois o padrão MVC é ideal para ser utilizado em aplicações com três camadas, como o WebCELLP. A abordagem utilizada permite uma facilidade maior no aumento da escalabilidade dos módulos do sistema, além de facilitar a adição de outros aplicativos que podem interagir com o WebCELLP. Além disso, a abordagem pode ser generalizada para outros aplicativos disponíveis em diferentes instituições ou mesmo dentro do DEEC da UFPA.

REFERÊNCIAS BIBLIOGRÁFICAS

BROWN S. *et al.* **Professional JSP 2nd Edition**. [s.l.]: Wrox, 2001.

CAVALCANTE, A. M. *et al.* Software Educacional para Dimensionamento de Sistemas Móveis Celulares. In: X SIMPÓSIO BRASILEIRO DE MICROONDAS E OPTOELETRÔNICA, 2002, Recife. **Anais**. Recife: UFPE, 2002. p. 389-393.

COULOURIS, George; DOLLIMORE, Jean; et al. **Distributed Systems: Concepts and Design**. England: Pearson Education, 2001.

CUMMINS, F. A. **Integração de Sistemas: arquiteturas para integração de sistemas e aplicações corporativas**. Rio de Janeiro: Campus, 2002.

EMMERICH, W. Software Engineering and Middleware: A Roadmap. In: The Future of Software Engineering – 22nd Int. Conf. On Software Engineering, 5, 2000. ACM Press, 2000. p. 117-129.

MAIA, Carmem. Educação a distância no Brasil na Era da Internet. in: ——— (Coord.). **ead.br**. São Paulo: Anhembi Morumbi, 2000.

MIRANDA, A. M. L. **Integração de Sistemas Legados para aprendizado a Distância: Estudo de Caso em Planejamento de Sistemas Móveis Celulares**. 2003. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica, Universidade Federal do Pará, Belém.

OMG, Object Management Group. Common Object Request Broker Architecture: Core Specification Version 3.0, OMG Document, December 2002. Disponível em: <http://www.omg.org/technology/documents/corb_spec_catalog.htm>. Acessado em: 09/02/2003

ENVIROMENT FOR DISTANCE LEARNING: SPECIFIC STUDIES IN PLANNING MOBILE CELLULAR SYSTEMS

Abstract: *A challenging offering a continuous education is the universal support software available to learning, to be used within curricula structures in engineering courses. An alternative is to release as tools in available sites in the web, so that they are available in any places at any time. Nevertheless, it is difficult to join in a site several softwares which were developed in different languages, operating systems and computer architecture, always providing binary incompatibility within the softwares. This work presents an approach proposal to providing an interoperability within softwares, in order to permit a integration*

within the several solutions already available, intra or inter institutions. The proposal presents how one can perform this integration applying technologies as UML, MVC, CORBA e SOAP for developed softwares for mobile cellular systems planning. This integration solution can be extended to any knowledge branch, and not only to engineering, due to a generic strategy for the integration of legacy systems/software and access universal.

Key-words: *Distance education, Integration of Systems, Distributed Objects, CORBA, Mobile cellular systems.*