

## DEDICATED - MODULAR TRAINING SYSTEM

**Mário Rui Gomes**

Instituto de Engenharia de Sistemas e Computadores (INESC)  
Rua Alves Redol 9 - Apt 13069  
1000 LISBOA PORTUGAL  
Instituto Superior Técnico  
Departamento de Engenharia Electrotécnica e de Computadores

**José M. Velez**

Instituto de Engenharia de Sistemas e Computadores (INESC)  
Rua Alves Redol 9 - Apt 13069  
1000 LISBOA PORTUGAL

**J. M. Brisson Lopes**

Instituto Superior Técnico  
Departamento de Engenharia Mecânica  
1000 LISBOA PORTUGAL

## SUMÁRIO

Neste artigo descreve-se a implementação em UNIX de um sistema modular e distribuído para o Ensino Assistido por Computador. É descrito em algum detalhe a implementação das camadas de software de mais baixo nível e da *Learning Material*.

## 1. INTRODUÇÃO

Numa Europa Unida, a formação avançada, tanto ao nível universitário como ao nível profissional, é actualmente encarada numa perspectiva supra-nacional. Os projectos europeus, consubstanciados em programas como, entre outros, os programas Erasmus, Delta, Commett, Impact e Race, têm constituído uma das vias pelas quais a União Europeia tem avançado no sentido da definição de uma estratégia nestas matérias. A isto também não é alheia a celebração de acordos entre os Doze tendo como objectivo a



definição e homogeneização de estatutos profissionais, como o estatuto europeu do Licenciado em Engenharia e do Engenheiro.

Para que sejam atingidos os importantes e complexos objectivos de tais acordos, quer em termos curriculares, quer em termos de intercâmbio e intermutabilidade de professores e alunos dos vários países, são imprescindíveis largos meios financeiros que, infelizmente, os programas referidos não têm podido disponibilizar. Para avaliar da disparidade dos recursos financeiros bastará comparar os recursos disponibilizados pela União Europeia para o desenvolvimento tecnológico de Circuitos Integrados ("Indústria do VLSI") e da formação profissional ("Indústria do FSE") com os reduzidos recursos disponibilizados para os anteriores programas. Em Portugal, os investimentos comunitários na "Indústria do VLSI" têm produzido importantes resultados em termos científicos, tecnológicos e financeiros, ao passo que os financiamentos à "Indústria do FSE" têm tido, em muitos casos, impactos negativos bem conhecidos nas actividades das Universidades portuguesas.

Ultrapassando estas condicionantes, o Prof. J. L. Encarnação, director do centro de computação gráfica de Darmstadt, Alemanha, decidiu dar um importante impulso com o objectivo de criar as infra-estruturas humanas e materiais necessárias à implementação de curricula europeus. Assim, no que depois ficou conhecida como "Declaração de São Marcos", num seminário realizado no referido palácio, lançou um desafio a várias Universidades e Institutos portugueses para, em conjunto com congéneres alemães, tentarem alcançar esses objectivos. Na mesma reunião foram definidas as estratégias iniciais e perspectivados os recursos financeiros necessários, que atingiam alguns milhões de contos.

O projecto DEDICATED (DEvelopment of a new DIMension in european Computer Aided TEaching), assim formado, submeteu então uma proposta ao Programa Delta com o fim de obter os financiamentos necessários para a aquisição das infra-estruturas de formação dos Centros Locais, LTC (Local Topic Center), nomeadamente equipamentos, a operacionalização da rede de LTCs e dos próprios LTCs (incluindo comunicações) e o desenvolvimento de todo um vasto conjunto de software dedicado denominado MTS (Modular Training System).



Infelizmente, o financiamento efectivamente concedido foi muito inferior ao solicitado e totalmente desadequado aos objectivos iniciais. Foi então necessário proceder a uma redefinição dos objectivos imediatos sem que tal implicasse o desvirtuar do projecto. Tal foi obtido pela redução drástica da funcionalidade do MTS e da diminuição do número de LTCs bem como da sua dimensão. A redefinição da estratégia, na qual o Prof. Encarnação teve mais uma vez um papel fundamental, implicou o alargamento do prazo temporal do projecto, sendo objectivos imediatos apenas a implementação e teste de protótipos operacionais interligados. O sucesso do projecto DEDICATED seria depois empregue na obtenção de meios financeiros mais adequados aos objectivos iniciais.

Esta comunicação apresenta as características fundamentais do Modular Training System, cujo desenvolvimento foi confiado à responsabilidade de um grupo de parceiros alemães (IGD-FhG, THD-GRIS e CLS) e portugueses (INESC e IST-DEM).

O MTS é uma estrutura baseada no conceito de Cliente/Servidor, simétrico e independente da plataforma informática, funcionando sobre MS-Windows (da responsabilidade do IGD-FhG) e UNIX (da responsabilidade conjunta do INESC e do IST-DEM) com o qual comunica um interpretador de uma linguagem descritora dos cursos a leccionar (da responsabilidade da CLS).

Assim, foi possível preservar o essencial dos objectivos iniciais que consistiam no desenvolvimento de um Sistema Autor distribuído, funcionando sobre as redes internacionais de alto ou baixo débito (ISDN) e independente da plataforma.

Este artigo descreve a arquitectura do MTS e a sua implementação, particularmente sobre a plataforma UNIX. É dada particular relevância aos aspectos Multimédia e ao desenvolvimento de uma API comum ao ambiente MS-Windows e ao ambiente UNIX, e que será utilizado pelo interpretador de cursos (CI) e pelos objectos intermédios que descrevem os conteúdos didácticos. É também apresentado o trabalho realizado no sentido da persistência e neutralidade dos objectos manipulados pelos cursos.

A utilização e alguns dos resultados obtidos utilizando a plataforma existente para o desenvolvimento de métodos para a avaliação de conhecimentos são também apresentados.



Nas conclusões serão apresentadas as perspectivas de evolução do sistema tendo em conta vários cenários e a experiência adquirida no desenvolvimento do MTS.

## 2 - ARQUITECTURA DO SISTEMA

O projecto DEDICATED visa, como já foi referido, implementar as infraestruturas e desenvolver o software necessários ao ensino assistido por computador (EAC) promovendo a cooperação entre os diversos países da CEE. Essa cooperação pretende ser efectuada a vários níveis, sendo um dos mais significativos a troca de experiências e de *know-how* sob a forma de material didático. Para isso foi prevista a instalação de 3 *Local Training Centers* (LTCs) situados em Portugal, França e Grécia. Com estes LTCs pretende-se criar as infraestruturas para dois tipos de utilizadores: os criadores de cursos e os alunos que os vão receber. Ambos irão utilizar a infraestrutura informática instalada nestes centros assim como o software desenvolvido para apoiar o desenvolvimento e utilização dos cursos (*Modular Training System, MTS*).

Pretende-se futuramente, promover o intercâmbio de cursos entre os diversos LTCs, de modo a criar uma verdadeira rede europeia de Ensino Assistido por Computador (EAC), ligada em tempo real através de uma infraestrutura de telecomunicações, seja utilizando os serviços da rede Internet, numa primeira fase, seja através da utilização dos serviços da emergente rede RDIS europeia. Deste modo, quando um aluno precisa de um curso sobre uma determinada matéria tem acesso a tudo quanto está disponível nos vários LTCs sem estar restrito ao que existe no seu centro local.

Deste modo, e em resumo, o projecto Dedicated tem os seguintes objectivos:

- Desenvolvimento do MTS (software de base)
- Instalação de uma infraestrutura europeia de LTCs
- Prototipagem e teste de cursos

Um dos requisitos do MTS é a independência de uma plataforma de hardware específica e a fácil portabilidade para qualquer novo sistema que seja necessário suportar. Esta necessidade surgiu porque se colocou logo à partida o cenário dos vários LTC's

serem compostos por um grupo heterogénio de máquinas, dominado principalmente por equipamentos UNIX/X11/Motif e DOS/Windows interligados por uma rede local (LAN) com um *gateway* para o exterior.

Deste modo a arquitectura encontrada de uma forma consensual para o MTS, é composto por várias camadas de software, cada uma fornecendo serviços à superior que fossem independentes do hardware utilizado e com uma grau de abstracção cada vez maior, de modo a facilitar a tarefa dos programadores finais do MTS e, em última análise, do professor que pretender desenvolver cursos utilizando o MTS. Deste modo consegue-se uma total independência do código desenvolvido para todas as camadas, exceptuando a primeira que é a que fornece os serviços de interface com o hardware específico de cada uma das máquinas.

As camadas de software que compõem o MTS são em número de 3 e descrevem-se de seguida:

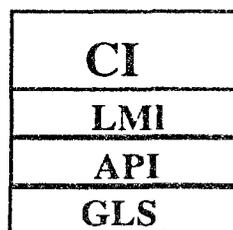


Figura 1

- O CI (*Course Interpreter*), ou interpretador de cursos, executa o *script* que descreve um determinado curso. Esse *script* define totalmente o curso indicando, entre outras coisas, que texto aparece no écran, onde e com que forma, que imagens são apresentadas e em que momento, que sons são produzidos e como se efectua a navegação dentro das várias páginas que compõem o curso.

- A LMI (*Learning Material layer*) é constituída por uma biblioteca de objectos de alto nível (LMCLib) que podem ser utilizados pelo professor para definir um curso. São exemplos de elementos desta biblioteca os objectos que implementam exames, objectos de demonstração de determinados conceitos, como um editor de CAD, entre outros. Mais

adiante descreve-se em detalhe a implementação dos objectos *exame* usados para a avaliação de conhecimentos.

- A GLS (*Generic Learning Suport*) fornece todos os serviços de baixo nível que, em principio, têm uma forte dependência do sistema operativo. Esses serviços são, entre outros, todo o *output* gráfico para o écran quer na forma *raster* (imagens paradas ou video) quer na forma vectorial (primitivas gráficas para desenho: rectas, círculos, etc.), texto, acesso a ficheiros e bases de dados, timers e primitivas de mais alto nível como botões, hot-zones, dialogs, etc.

Por outro lado, pretendeu-se também separar a GLS do resto do MTS, comunicando os dois sub-sistemas através de mensagens enviadas através de *sockets*. Deste modo, consegue-se uma maior versatilidade do sistema, podendo o CI e a LMC serem executados numa máquina de maior desempenho, tornando-se necessário correr apenas o GLS na máquina usada pelo utilizador do sistema.

Na figura 1 está esquematizada a interligação possível entre os diversos módulos num LTC heterogéneo com máquinas UNIX e PC compatíveis.

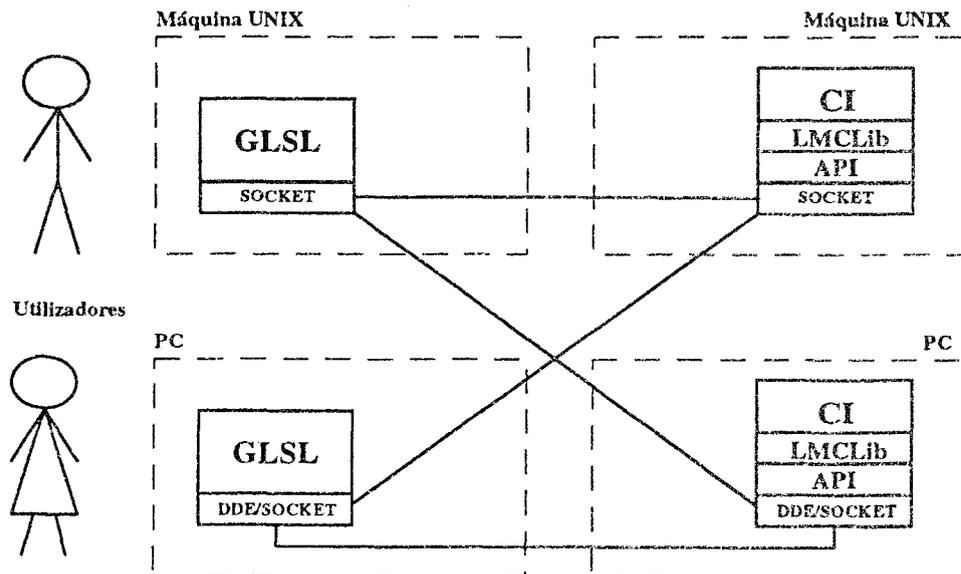


Figura 2

### 3- IMPLEMENTAÇÃO DA GLSI

Para a implementação da *Generic Learning Support layer* em UNIX optou-se por usar o sistema gráfico de janelas X11, com a biblioteca *Motif*. Desta forma garante-se a portabilidade deste sistema para um grande número de máquinas, bastando para isso que elas disponham do X11 e do *Motif*, que estão a tornar-se normas *de facto* no mundo UNIX. Neste momento o GLS UNIX já foi testado com sucesso em máquinas *SGI Indigo* (Silicon Graphics Indigo) e *Sun Sparc*.

O GLS tem uma estrutura interna *event driven*, não só no que diz respeito à interacção com o sistema gráfico e com o gestor de janelas como com os níveis superiores do MTS com os quais comunica através de mensagens enviadas pelo mecanismo de *sockets* UNIX. A cada evento (mensagem) que chega das camadas superiores, corresponde normalmente uma acção a realizar pelo GLS: desenhar uma primitiva gráfica, colocar uma imagem numa janela ou dar início a uma sequência de vídeo em tempo real. Cada mensagem que é enviada corresponde normalmente a uma acção efectuada pelo utilizador que pode ser a activação de um botão ou de uma *hot-zone*, a selecção de uma opção num menu ou a introdução de texto como resposta a uma questão. Pode facilmente verificar-se que utilizando este mecanismo de comunicação baseado em mensagens se consegue tornar a camada GLS independente das superiores que, num caso extremo, nem necessitam de estar a ser executadas na mesma máquina, podendo uma ser um PC e a outra um sistema UNIX.

As camadas superiores do MTS utilizam este mecanismo de mensagens de uma forma indirecta, estando encapsulado no API (*Application Programming Interface*). Deste modo, para invocar uma primitiva suportada pela GLS basta chamar uma função que se encarregará de converter os respectivos parâmetros numa (ou mais) mensagens. Da mesma forma, as camadas de software superiores são notificadas da ocorrência de acções pela GLS através do *polling* da fila de eventos pendentes ou por um mecanismo de mais alto nível de *callback*. Para utilizar esse serviço basta fazer o registo de uma *callback* que ficará associada a uma determinada primitiva. Quando houver qualquer interacção com esse elemento, a mensagem proveniente da GLS é convertida pelo API numa chamada à (ou às) função de *callback* que estiver registada para esse elemento. Se não houver nenhuma função registada para um dado elemento, qualquer interacção com ele não terá efeito algum. Evidentemente que para este mecanismo de *callback* funcionar

correctamente, o controlo do programa deve ser passado para uma função (*DdcMainLoop*) que fará a gestão de todo este mecanismo que acabámos de descrever.

Vamos fazer agora um apanhado das principais funcionalidades oferecidas pela GLS. Em termos de primitivas gráficas vectoriais estão disponíveis linhas, arcos de circunferência, linhas poligonais, rectângulos, círculos e elipses. Os atributos das primitivas gráficas podem ser alterados de forma individual, e vão desde a cor até ao tipo e largura de linha. O texto é vectorial, podendo ser definido com o formato RTF, tornando assim compatíveis todos os processadores de texto que utilizam este formato para a descrição de um texto formatado. As imagens paradas podem ser lidas em vários formatos, sendo presentemente suportados os seguintes: GIF, BMP, PCX, PPM e TIFF. Em termos de menus pode criar-se uma hierarquia qualquer baseada em grupos e em items: uma barra de menus contém vários grupos que, por sua vez, podem conter outros subgrupos e/ou items. Os items (opções) dos menus podem ser activadas ou tornadas inactivas, sendo o seu tipo *normal* ou *toggle* (memorizam o estado corrente: seleccionado ou desseleccionado). Uma das primitivas mais usadas, especialmente para implementar os mecanismos de navegação, é o botão. Um botão pode ser activado ou desactivado, sendo accionado pelo utilizador através de um click dentro da sua área. As *hot-zones* têm um funcionamento semelhante ao dos botões, com a diferença de que não têm representação gráfica.

Outros dos objectos suportados são as janelas. Estas podem ter qualquer tamanho, posição ou cor e ser posteriormente movidas ou sofrer alterações dos atributos. Também foram previstos alguns objectos de mais alto nível. Nomeadamente *sliders*, para permitir a introdução de valores numéricos, dialogos simples com o utilizador para mostrar mensagens e objectos vocacionados para a introdução de texto.

As primitivas oferecidas pela GLS têm o mesmo aspecto e funcionalidade qualquer que seja a máquina utilizada, conseguindo-se deste modo reduzir o tempo de aprendizagem de utilização do sistema quando se muda de sistema.

Em termos de objectos multimédia são suportados audio e video, havendo a possibilidade de efectuar comunicação ponto a ponto entre máquinas diferentes através da rede local tanto em video como em audio. Para ultrapassar os problemas postos pela utilização de um ambiente com máquinas heterogéneas, a comunicação tanto de vídeo



como de audio é efectuada com um formato normalizado na rede, sendo convertida para o formato nativo de cada uma das máquinas por um servidor local. Deste modo consegue-se obter o maior desempenho de cada uma das máquinas pois pode-se utilizar o *hardware* dedicado que cada uma possui.

Uma outra característica importante do sistema MTS é a implementação dos objectos, simples ou compostos, pertencentes à *Learning Material Layer* sob a forma uma estrutura de árvore única, tal como a apresentada na Figura 3.

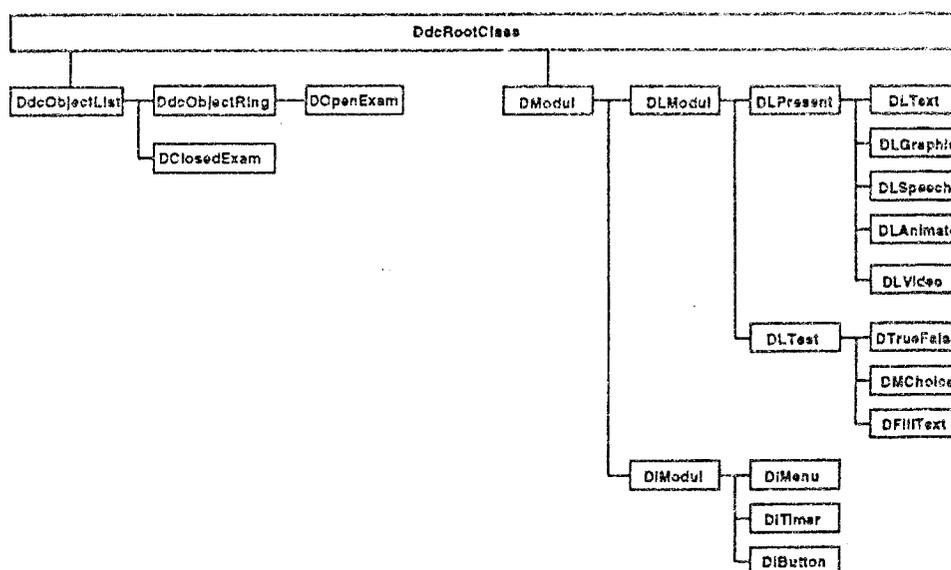


Figura 3

A estratégia de implementação em árvore única foi ditada pela necessidade de obter um procedimento coerente e transparente para a persistência dos objectos constituintes da Learning Material layer. De facto, a inserção ou extracção de um objecto para ou de memória persistente é efectuada (em C++) por uma simples instrução de inserção ou de extracção, o que torna estas operações extremamente simples e transparentes. Por outro lado, dado que no mesmo centro DEDICATED (LTC) devem poder coexistir e cooperar plataformas de "hardware" distintas, o arquivo de objectos em memória persistente tem que ser realizado de uma forma tal que os objectos guardados em ficheiros possam ser lidos ou escritos a qualquer momento por qualquer das plataformas de "hardware" suportadas pelo sistema.

Os objectos são assim arquivados de uma forma neutra única em ficheiros. Desta estratégia resultam ainda os benefícios da portabilidade dos objectos e da sua transferência entre os vários centros (LTCs) constituintes da rede DEDICATED sem necessidade de efectuar qualquer conversão.

#### 4 - LEARNING MATERIAL LAYER e AVALIAÇÃO

É no nível intermédio (Learning Material layer) que se situam os objectos que o aluno verá durante a frequência de um curso. Estes objectos são de dois tipos: objectos contendo material de aprendizagem (LMOs) e objectos intrínsecos (IMOs).

Os objectos intrínsecos (IMOs) destinam-se a criar as condições necessárias para a apresentação e interacção do aluno com os objectos de conteúdo didáctico. Exemplos de objectos intrínsecos são janelas, botões e menus. Como é natural, os objectos intrínsecos podem ser hierarquizados, como é o caso de menus constituídos por um número variável de botões.

Os objectos contendo material de aprendizagem (LMOs) são objectos básicos criados pelos autores dos cursos e representam os objectos elementares multimedia dos tipos fundamentais (texto, imagem, som, animação, etc.) implementados nos cursos. O seu conteúdo e apresentação podem variar, ao passo que os objectos intrínsecos mantêm sempre a mesma aparência. Vários LMOs podem ser agrupados para constituir objectos multimedia em que, por exemplo, uma imagem é acompanhada por um fundo musical.

Todos estes objectos, LMOs e IMOs, não são responsáveis pela sua apresentação, mas constituem-se como objectos de controlo genérico dos objectos que serão efectivamente apresentados no contexto de apresentação do DEDICATED, implementado através do GLS. Tudo o que os LMOs e IMOs fazem é, através da API anteriormente apresentada, requerer ao GLS a efectiva apresentação dos seus conteúdos, deixando ainda ao GLS o encargo de implementar a interacção de baixo nível com o utilizador.

Entre os objectos do tipo LMO encontram-se os objectos que implementam a avaliação. Esta constitui uma importante componente de qualquer tipo de ensino. É através dela que se procede à determinação do grau de compreensão e maturidade do aluno em relação às



matérias leccionadas. No projecto DEDICATED, a avaliação toma ainda uma posição mais relevante pois a exposição de novas matérias é condicionada pelo grau de aproveitamento do aluno nas matérias já expostas, dado que o sistema realiza ajustamentos dos graus de dificuldade e abstracção dos temas a expor em função do aproveitamento anterior do aluno.

O sistema DEDICATED necessita ainda da avaliação para levar o aluno a recapitular partes da matéria que o sistema detecte (via avaliação) terem sido deficientemente adquiridas, ou mesmo para obrigar o aluno a voltar atrás no curso caso o sistema detecte deficiências graves.

Por outro lado, num ensino apoiado em meios telemáticos, será frequente o caso do aluno que se encontre a frequentar um curso à distância ou sem apoio do monitor do curso. Em tais situações é de todo conveniente que o aluno possa, por si próprio, avaliar os seus conhecimentos.

No sistema DEDICATED, a avaliação baseia-se na apresentação de perguntas de vários tipos a que o aluno deve responder. Porque o sistema é modular, estas perguntas poderão aparecer isoladamente à medida que a matéria é exposta ou em grupos no fim de cada unidade de ensino. O sistema permite aos autores dos cursos o emprego de uma ou de ambas as estratégias.

Naturalmente, as perguntas empregam objectos como texto, som, imagem, etc. Estes últimos objectos existem já na árvore de objectos, mas há que lhes dar coerência. Para tal existem objectos do tipo LMO (classe DLModul) representando perguntas. Os objectos de tipo LMO dividem-se então em dois tipos distintos, tal como a Figura 3 apresenta. Os vários tipos (ou classes) de objectos básicos, como texto e imagem, foram agrupados numa subclasse de LMOs, a subclasse de objectos didácticos de conteúdo apresentável, DLPresent. Os objectos do tipo de pergunta, que fazem uso daqueles, foram agrupados numa outra subclasse de LMOs, a classe DLTest.

A análise efectuada e a posterior experimentação com um protótipo mostraram que, dos diversos tipos de perguntas, os seguintes são os mais frequentes:



a) Verdadeiro ou Falso - uma frase é assinalada como verdadeira ou falsa (ou ainda, "não sei"). Por frase entende-se uma frase multimédia. Uma frase multimédia é uma pergunta ou uma resposta que emprega mais do que um tipo de media como, por exemplo, o texto de uma pergunta que é acompanhado por uma imagem ilustrativa e por uma sequência sonora. Verificou-se ainda que cada pergunta deveria ser passível de ser apresentada de mais do que uma forma.

b) Resposta Múltipla - é apresentada uma frase contendo uma pergunta conjuntamente com (normalmente 4 ou 5) frases de resposta em que só uma delas está totalmente correcta, duas aparentam ser correctas e as restantes estão totalmente erradas. As frases são frases multimédia. Estas perguntas devem ser preparadas por forma a ser possível fazer uma pergunta de várias formas e ter um número de respostas maior do que aquele que é efectivamente apresentado. As respostas devem ainda poder ser apresentadas numa sequência aleatória.

c) Pergunta Múltipla Verdadeiro ou Falso - neste tipo de perguntas são simultaneamente apresentadas várias perguntas do tipo verdadeiro ou falso. O número de subperguntas a apresentar é variável e deverá ser aleatoriamente escolhido de entre um número maior de subperguntas disponíveis.

d) Frases a Completar - uma ou mais frases em que faltam alguns itens (palavras, por exemplo) são apresentadas para que o aluno as complete. O número de frases a apresentar é variável e as frases deverão ser aleatoriamente escolhidas de entre um número maior de frases disponíveis.

e) Pergunta de Resposta Livre - o aluno responderá (normalmente por meio de um texto de tamanho limitado à partida) a uma pergunta. A avaliação deste tipo de perguntas tem que ser efectuada à posteriori pelo monitor/professor do curso.

A cotação das perguntas é efectuada segundo o critério em que são atribuídos  $n$  pontos por uma resposta certa, retirados  $m$  pontos quando a resposta está errada ( $m$  pode ser 0) e, finalmente, retirados  $k$  pontos (em que  $k$  pode ser 0) se o aluno escolheu não responder à pergunta.



Nas perguntas múltiplas do tipo verdadeiro ou falso e nas perguntas com frases para completar, cada uma das subperguntas ou items em falta é cotado segundo o esquema anterior em que  $n$ ,  $m$  e  $k$  podem variar livremente de subpergunta ou item para subpergunta ou item. Globalmente, são atribuídos  $p$  pontos à pergunta se todas as respostas estiverem correctas. Se não, é atribuída uma cotação proporcional ao quociente entre os pontos acumulados pelo aluno e o número máximo de pontos que poderia ter acumulado caso tivesse respondido correctamente a todas as subperguntas ou items.

O processo apresentado para a atribuição da cotação às respostas reflecte o procedimento adoptado quando várias perguntas são agrupadas constituindo testes (ou exames). Estes contituem uma outra classe de objectos da Learning Material Layer. Os tipos de testes implementados são:

a) Testes abertos - durante a realização do teste, o aluno pode analisar qualquer das perguntas e responder às mesmas pela ordem que mais lhe convier, podendo modificar respostas anteriormente seleccionadas. Este tipo de testes é implementado sobre uma colecção de objectos organizada em anel e necessita de instrumentos de navegação tais como "avance para a próxima pergunta" ou "acabei de responder". Estes testes podem ainda estar limitados no tempo pelo emprego de um "timer". Quaisquer tipos de perguntas podem ser empregues nestes testes.

b) Testes Fechados - a ordem das perguntas é fixa nestes testes e o aluno só poderá passar de uma pergunta para a seguinte se tiver respondido satisfatoriamente à pergunta corrente. Cada uma das perguntas pode ter que ser respondida dentro de um tempo limite. As perguntas dos tipos Verdadeiro ou Falso e de Resposta Múltipla não se ajustam a este tipo de testes mas poderão ser empregues se o número de vezes que o aluno a elas pode responder e corrigir a resposta fôr limitado.

Na fase inicial do projecto DEDICATED foi desenvolvido um protótipo experimental em que foram implementados testes do tipo aberto com perguntas dos tipos Verdadeiro ou Falso e de Resposta Múltipla, ambos com frases multimédia. Este protótipo tinha como finalidade avaliar da concepção e implementação dos conceitos e objectos. A avaliação efectuada permitiu corrigir a concepção inicial e a implementação posteriormente realizada.

## 5 - CONCLUSÕES

Dos protótipos implementados é possível extrair algumas conclusões confirmando muitas das metas propostas pelos objectivos iniciais.

O primeiro, e talvez o mais importante, é a portabilidade do código que se verificou ser praticamente total. Este resultado é ainda verdade quando aplicado ao *look and feel* do produto, descontadas as particularidades de cada uma das plataformas.

Verificou-se também que a especificação funcional era adequada aos fins propostos. A experiência obtida durante o desenvolvimento dos protótipos, como foi o caso da avaliação, permitiu ainda complementar a especificação funcional.

Outro objectivo relevante atingido foi a obtenção da persistência de objectos em forma neutra que permite obter esta persistência independentemente da plataforma utilizada e ainda possibilita a distribuição dos cursos sem qualquer necessidade de conversão.

Estão assim lançados os alicerces para uma verdadeira rede de Ensino Assistido por Computador a nível europeu.

## 6 - AGRADECIMENTOS

Os autores pretendem aqui deixar os seus agradecimentos a todos os parceiros do projecto DEDICATED pela sua valiosa contribuição.

Um especial agradecimento ao Prof. Encarnação fica aqui registado.



## 7 - BIBLIOGRAFIA

[1] J. M. Velez; B. Tritsch, "Revised MTS Specification", *DELTA Deliverable 11, Project D2014 DEDICATED*, DELTA Commission of the European Community, August 1993.

[2] B. Tritsch, Ch. Hornung, "DEDICATED - Learning on Networked Multimedia Platforms", IFIP WG 3.2 Working Conference, Visualisation in Scientific Computing: Uses in University Education, July 1993

[3] Brisson Lopes, Gomes, Graf, Knierriem, Lindner, Tritsch, Velez, Zysk, "Functional Specification of the Modular Training System", *DELTA Deliverable 3, Project D2014 DEDICATED*, DELTA Commission of the European Community, August 1992.

