

UNIVERSIDADE DE ÉVORA
DEPARTAMENTO DE INFORMÁTICA



**INTERFACE INTELIGENTE PARA O CADERNO
ESCOLAR ELECTRÓNICO**

Luís Manuel Rodrigues Vital Alexandre

Mestrado em Engenharia Informática

Dissertação orientada por:

Professor Doutor Salvador Luís de Bethencourt Pinto de Abreu

Évora

2010

UNIVERSIDADE DE ÉVORA
DEPARTAMENTO DE INFORMÁTICA



**INTERFACE INTELIGENTE PARA O CADERNO
ESCOLAR ELECTRÓNICO**

Luís Manuel Rodrigues Vital Alexandre

Mestrado em Engenharia Informática

Dissertação orientada por:

Professor Doutor Salvador Luís de Bethencourt Pinto de Abreu



185 658

Évora

2010

INTERFACE INTELIGENTE PARA O CADERNO ESCOLAR ELECTRÓNICO

Nesta dissertação descrevemos a construção de uma interface inteligente para uma ferramenta que pretende ser uma alternativa eficaz aos tradicionais cadernos e livros utilizados nas salas de aula, dado que fornece apoio tanto na recolha como na organização da informação - O Caderno Escolar Electrónico. Esta interface é capaz de prever qual a tarefa mais provável que o utilizador irá efectuar no sistema, com base num conhecimento prévio das acções já efectuadas, assim como no tempo e percurso efectuado até determinada altura. Esta adaptabilidade visa a melhoria da interacção entre o sistema e o seu utilizador, reduzindo assim o número de interacções explícitas necessárias para efectuar uma determinada tarefa. Para conseguir prever as tarefas mais prováveis em determinada situação, o sistema recorre a algoritmos de classificação, baseados em árvores de decisão e redes de Markov/Modelos "Escondidos" de Markov. Os utilizadores com necessidades especiais, são o principal alvo deste projecto, mas o sistema poderá também dar um grande contributo a outros utilizadores em idade escolar, dado que esta ferramenta favorece a adopção de um processo metódico para recolha e organização de notas em contexto de sala de aula. Por ter sido desenvolvido com recurso a técnicas de desenho para sistemas interactivos, encontra-se totalmente adaptado aos utilizadores alvos.

INTELLIGENT INTERFACE FOR THE ELECTRONIC SCHOOL NOTEBOOK

In this Master Thesis, we describe the construction of an intelligent interface for a tool that intends to be an effective alternative to the traditional books and exercise books used in classrooms, as it provides support for both collecting and organizing information - The Electronic School Notebook. This interface has the ability to predict the most likely task that a user is about to perform, based on prior knowledge of action already made in the system, combined with action time frame. This adaptive capability will reduce the number of explicit interactions to accomplish a certain task. In order to predict the most likely tasks in a certain context, the system uses machine learning algorithms, based on decision trees and Markov chains/Hidden Markov Models. The users with special needs, are the main target of this project, but the system could give a major contribution to other users in school. This tool promotes a methodic process of collecting and manage the school notes. Developed with techniques for interactive systems, this tool is totally adapted to its users.

Dedicatória

Dedico esta dissertação à minha esposa, Fernanda, por todo o apoio, incentivo, amor, compreensão e companhia ao longo de todos estes anos.

À Barbara e José Luís, meus filhos, principal fonte de energia e inspiração para o meu dia-a-dia.

Agradecimentos

Uma dissertação de mestrado é um trabalho solitário, algo a que qualquer investigador está destinado. No entanto, um trabalho como o proposto nesta tese, nunca chegaria a bom porto sem a preciosa colaboração de inúmeras pessoas, que sem qualquer tipo de condição me apoiaram e me ajudaram, muitas vezes retirando tempo das suas actividades profissionais, para responder às minhas solicitações. Sem estes contributos, todo o trabalho em torno desta dissertação não teria sido possível.

Ao Professor Doutor Salvador Luís de Bethencourt Pinto de Abreu, orientador desta dissertação, agradeço o apoio, a partilha do saber e de experiências, bem como as valiosas contribuições para o trabalho. Gostaria de voltar a salientar todo o apoio prestado e encorajamento, quer para esta dissertação, quer para os artigos escritos no âmbito da mesma, bem como o desafio lançado para a continuação da minha ligação a esta instituição.

Ao Gonçalo, que mesmo à distância ajudou a diminuir as longas noites de trabalho e que sempre esteve disponível para ajudar.

Ao Luís Garcia, pela ideia, por todo o apoio dado e encorajamento quando o desespero começava a retirar o discernimento.

Aos membros do Laboratório de Sistemas de Informação e Interactividade do Instituto Politécnico de Beja, pelos preciosos contributos e ideias dadas para este projecto.

À Professora Francisca Alface e ao Professor Paulo Coelho, pelas rápidas respostas às minhas solicitações, pelas interrupções das actividades lectivas que eu provoquei e finalmente pelo seu profundo empenho neste projecto.

À Laura que mesmo sem saber, tanto contribuiu para este projecto.

Finalmente, à minha esposa por todo o apoio e encorajamento que me deu, pela força que teve de ter para muitas vezes suportar sozinha a gestão da nossa casa e a educação dos nossos filhos.

Conteúdo

CAPÍTULOS

1	INTRODUÇÃO	1
2	O ESTADO DA ARTE	5
2.1	O Caderno Escolar Electrónico	5
2.2	Sistemas <i>Context-aware</i>	6
2.3	<i>Context-aware</i>	8
2.3.1	Dimensões do Contexto	10
2.4	Paradigmas sobre Adaptabilidade e Previsibilidade	11
2.4.1	Interfaces Adaptativas e Preditivas	12
2.4.2	Interfaces Atentas e Sugestivas	15
3	ARQUITECTURA DO CE-e ADAPTATIVO	17
3.1	Estrutura da Base de Conhecimento	17
3.2	Os Algoritmos	19
3.2.1	Modelos "Escondidos" de Markov	22
3.3	As Tecnologias Utilizadas	27
3.3.1	Linguagem de Programação C#	27
3.3.2	XML - eXtensible Markup Language	29
3.3.3	SQLite	30
3.4	A Codificação	31
3.4.1	Treino do Classificador do Motor de Inferência	32
3.4.2	Utilização das Sugestões do Motor de Inferência	34
4	TESTES DA APLICAÇÃO	38
4.1	Os Utilizadores	38
4.2	As técnicas	39
4.3	Os Resultados	40

4.3.1	Resultados do Utilizador 1	41
4.3.2	Resultados do Utilizador 2	41
4.3.3	Resultados Globais	42
5	CONCLUSÕES E TRABALHO FUTURO	45
	BIBLIOGRAFIA	49

Lista de Figuras

Figura

2.1	Interface de <i>notetaking</i> do CE-e	6
2.2	Dispositivo do tipo PDA, com mecanismos de reconhecimento do contexto	7
2.3	Assistente para criação de novo documento do MS Word	10
2.4	Interface sugestiva de [17]	14
3.1	Estrutura dos logs do CE-e	18
3.2	Extracto de um ficheiro de <i>logs</i> do CE-e	19
3.3	Aspecto parcial do ficheiro submetido à ferramenta <i>Weka</i>	20
3.4	Matriz de confusão gerada pela ferramenta <i>Weka</i> com os dados da base de conhecimento	22
3.5	Bibliotecas e classes da <i>.Net Framework</i>	28
3.6	Exemplo de um ficheiro XML	29
3.7	Exemplo de um ficheiro XSD	30
3.8	Exemplo de código, em que se mostra o <i>query</i> e posterior inserção na <i>Data Table</i>	32
3.9	Exemplo de código, em que se mostra a conversão de inteiro para binário	33
3.10	Exemplo de código, em que se mostra o arranque da <i>thread</i>	34
3.11	Esquema de caminhos possíveis no sistema	35
3.12	CE-e a efectuar uma sugestão	36
4.1	Gráfico com os resultados globais do utilizador 1	39
4.2	Gráfico com os resultados parciais do utilizador 1	40
4.3	Gráfico com os resultados globais do utilizador 2	41
4.4	Gráfico com os resultados parciais do utilizador 2	42
4.5	Gráfico com os resultados globais	43

Capítulo 1

INTRODUÇÃO

A Declaração de Salamanca [31], proclamada por 88 países e 25 organizações internacionais, defende a integração das crianças com necessidades especiais no ensino regular. Para a concretização deste objectivo é fundamental colocar em prática diversos mecanismos de apoio aos alunos, entre os quais, o recurso a tecnologias em contexto escolar. Várias crianças com dificuldades motoras recorrem a computadores portáteis com aplicações de processamento de texto, desenho gráfico e matemática para o registo das diversas actividades lectivas. Por se tratar de uma tarefa diária são produzidas quantidades substanciais de informação que os alunos necessitam de organizar para acesso futuro. Para apoiar a recolha e organização destes registos o Laboratório de Sistemas de Informação e Interactividade (LabSI²) e o Centro de Paralisia Cerebral de Beja (CPCB) iniciaram o desenvolvimento de uma aplicação de computador que possa constituir uma efectiva alternativa digital ao tradicional caderno escolar [3].

Apesar da ideia inicial do projecto estar associada a alunos com necessidades especiais, considerando os recentes investimentos das escolas portuguesas em meios tecnológicos cremos que o conceito de Caderno Escolar Electrónico (CE-e) pode ser estendido com muitas vantagens a todos os alunos do ensino regular. Esta ferramenta, ao favorecer a adopção de um processo metódico de recolha de notas em sala de aula poderá introduzir benefícios assinaláveis no sucesso escolar [7]. Neste momento existe um protótipo inicial do sistema dotado de um conjunto básico de funcionalidades que possibilitam a criação de vários cadernos, cada qual associado a uma disciplina ou área de estudo, onde o aluno pode ir acrescentando, aula após aula, registos escritos, fórmulas matemáticas, documentos digitalizados e ligações a recursos externos, como por exemplo ao documento com os conteúdos escritos pelo professor durante a aula no quadro interactivo. O protótipo começou a ser utilizado este ano lectivo por uma aluna que se encontra a ser seguida pelo CPCB. O acompanhamento destes casos irá permitir avaliar as funcionalidades do sistema e identificar novas formas de apoio a implementar no caderno

electrónico.

Mas o CE-e poderá prestar um maior apoio a estes alunos, se reconhecer qual a tarefa em curso, ajudando assim os utilizadores a concluir os passos seguintes da mesma. Em algumas situações é difícil determinar as intenções que se encontram por detrás de uma sequência de acções do utilizador, mas noutros casos, quer recorrendo a conhecimento sobre o domínio [12], quer observando o comportamento do utilizador em situações anteriores [22], pode-se conseguir prever qual a operação que pretende realizar. Um caso óbvio será a abertura de uma nova lição no caderno de uma disciplina quando se aproxima a hora dessa aula. Além de prestarem apoio a alunos com dificuldades motoras estes mecanismos também poderão beneficiar alunos com dificuldades cognitivas que com este tipo de ajuda poderão trabalhar de forma mais autónoma em determinadas situações. Este tipo de mecanismos já tem vindo a ser experimentado em algumas ferramentas mais comuns, como por exemplo programas de e-mail [22], com o intuito de libertar o utilizador da execução de tarefas rotineiras, através da delegação da responsabilidade pela sua concretização no sistema. Esta abordagem também já foi experimentada no desenvolvimento de um sistema de apoio ao desenvolvimento de configurações de teclado para o Eugénio¹ [28, 14].

Assim, este projecto tem como objectivo, implementar e avaliar uma interface inteligente para o CE-e que forneça um maior nível de suporte a alunos com necessidades especiais na realização de diversas tarefas com esta aplicação. Estas ferramentas necessitam de conter modelos da aplicação e do utilizador [12] para que seja possível identificar as intenções deste último e assim fornecer ajuda na realização de uma tarefa no sistema. A utilização de informação sobre o contexto também poderá contribuir para um aumento da eficácia destes sistemas [8].

Assim, na primeira fase deste projecto procedeu-se à criação de uma base de dados que contém o conhecimento sobre as várias tarefas realizadas pelo utilizador no CE-e, as diferentes acções que as constituem e o contexto onde foram realizadas. Para a caracterização do contexto armazenou-se um conjunto de informação temporal como a data, dia da semana e hora. Paralelamente, assitiu-se a aulas por forma a obter um maior conhecimento sobre o contexto. Estes dados foram recolhidos no decurso das aulas, com o aluno a utilizar o caderno electrónico para o registo das várias actividades lectivas.

¹ Sistema de comunicação Alternativa e Aumentativa, desenvolvida pelo LabSI² em parceria com o Centro de Paralisia Cerebral de Beja o Instituto de Engenharia de Sistemas e Computadores Investigação e Desenvolvimento em Lisboa (INESC-ID).

Esta base de dados e de conhecimento foi utilizada na fase seguinte do projecto, onde foram criados e avaliados mecanismos de apoio ao utilizador na realização de diversas tarefas no CE-e, que entraram em linha de conta com todas as dimensões do contexto. É justamente aqui que acenta o *core* do desenvolvimento deste projecto, dado que se o sistema conseguir interpretar correctamente todas as dimensões do contexto, poderá evoluir de um conjunto de regras *hardcoded* que fornecerão um conjunto limitado de adaptações, para um sistema que com a sua contínua utilização conseguirá aprender e criar novas adaptações ao contexto, mais adequadas ao perfil do utilizador. Esta "inteligência" do sistema só será possível com recurso a algoritmos de aprendizagem automática. Por forma a não sobrecarregar o sistema, todos os *logs* serão efectuados com recurso à tecnologia SQLite².

A codificação do CE-e foi efectuada com recurso à *.NET framework* e à linguagem de programação *C#*, no entanto os algoritmos de aprendizagem automática poderão ser codificados com recurso a outras tecnologias, mais eficazes para este tipo de processamento de dados e integrados no sistema através de *DLLs*. Este conjunto de dados terá obrigatoriamente de estar associado à componente temporal, resultando assim os dados a tratar num conjunto de variáveis multidimensionais, que serão então alvo dos algoritmos de aprendizagem, que darão ao sistema a "inteligência" e as adaptações ao contexto mais adequadas ao momento.

Com este projecto pretende-se aliviar o utilizador de tarefas rotineiras criando um conjunto de mecanismos de suporte ao utilizador e adaptação ao contexto. Com estes mecanismos pretende-se atingir dois objectivos: (i) Eliminar um conjunto de tarefas que retirem rendimento a utilizadores sem necessidades especiais; (ii) Aliviar os utilizadores com necessidades especiais, de uma sobrecarga de acções que não contribuem para o correcto acompanhamento das aulas, dado que os atrasam e distraem dos conteúdos das mesmas. Todos estes mecanismos de suporte e adaptação ao contexto serão testados antes da sua efectiva implementação, através de testes em contexto real de aula. Numa primeira fase estes poderão ser acompanhados e numa segunda fase serão feitos inquéritos que ajudarão a perceber a efectiva utilidade do sistema. Intrinsecamente ligada a este projecto, está a ideia de melhorar a interface do Caderno Escolar electrónico, com vista a dotá-la de uma maior flexibilidade.

A base de conhecimento anteriormente referida, é essencial para se conseguir atingir

² Sistema Gestor de Base de Dados (SGBD), que não necessita de um servidor activo no sistema.

os resultados propostos. Assim definiu-se que o arranque do trabalho teria de passar inevitavelmente pela construção desta base em simultâneo com uma pesquisa bibliográfica, na qual se procuraria sistemas que utilizassem sistemas ou tecnologias idênticas de adaptação ao contexto.

O passo seguinte passou pela codificação do sistema, como já existia um protótipo este trabalho de codificação teve de passar pela introdução de novas funcionalidades e reescrita do código existente. Todo este processo estendeu-se por cerca de três meses. Após a codificação, o sistema foi colocado em avaliação. Esta avaliação foi efectuada por um conjunto de utilizadores que avaliaram, em contexto real de utilização, as adaptações ao contexto sugeridas pela Interface Inteligente do Caderno Escolar Electrónico.

Finalmente o último mês de trabalho foi dedicado à apreciação da avaliação, conclusões e definição do trabalho futuro.

Foi definido como objectivo final desta tese a construção de uma Interface Inteligente para o Caderno Escolar Electrónico que possa antecipar as acções do utilizador, sugerindo a acção mais provável que este realizará no sistema, através do realce do botão ou *widget* que despoleta essa acção. Por forma a ajudar ainda mais o utilizador, o botão ou *widget* será realçado e ser-lhe-á atribuído o focús, por forma a que com um simples toque na tecla "enter" ou "return" a acção possa ser despoletada.

O presente trabalho foi objecto de apresentação nas conferências Inforum'2010 - Simpósio de Informática, no *track* "Sistemas Inteligentes", que decorreu em Braga na Universidade do Minho nos dias 9 e 10 de Setembro e na conferência Interacção'2010 - 4ª Conferência Nacional em Interacção Humano-computador, que decorreu na Universidade de Aveiro entre 13 e 15 de Outubro de 2010.

Capítulo 2

O ESTADO DA ARTE

Hoje em dia vivemos rodeados de sistemas computacionais (e.g. desktops, laptops, PDAs, entre outros). Vivemos numa época em que todos os aspectos da nossa vida são influenciados ou determinados por decisões ou acções efectuadas por sistemas computacionais.

No entanto estes sistemas poderão prestar um maior apoio se o conseguirem prestar de uma forma automática e autónoma aos seus utilizadores. Este apoio só é possível se o sistema conseguir determinar ou reconhecer o contexto, em que tanto a máquina como o seu utilizador, estão enquadrados [6].

Se o Caderno Escolar Electrónico conseguir reconhecer o contexto em que se enquadra, poderá prestar um grande apoio aos utilizadores, em particular os utilizadores com necessidades especiais. Estes utilizadores quando inseridos no ensino regular terão de conseguir suportar e acompanhar os ritmos dos alunos sem necessidades especiais e assim todas as ajudas que o Caderno Escolar Electrónico lhes possa prestar, serão sem dúvida nenhuma um grande contributo e um grande alívio à sobrecarga a que são expostos.

2.1 O Caderno Escolar Electrónico

No início do desenvolvimento desta tese, já existia um protótipo funcional do Caderno Escolar Electrónico [3]. Este protótipo funcional teve uma fase de testes com utilizadores e, nesta altura estava a ser utilizado por alguns alunos, com necessidades especiais, que eram seguidos no Centro de Paralisia Cerebral de Beja e que estavam inseridos no ensino regular.

A experiência adquirida com a utilização, obtida através destes alunos, forneceu-nos as ideias e orientou-nos na formulação das bases deste trabalho. Demonstrou-nos ainda as

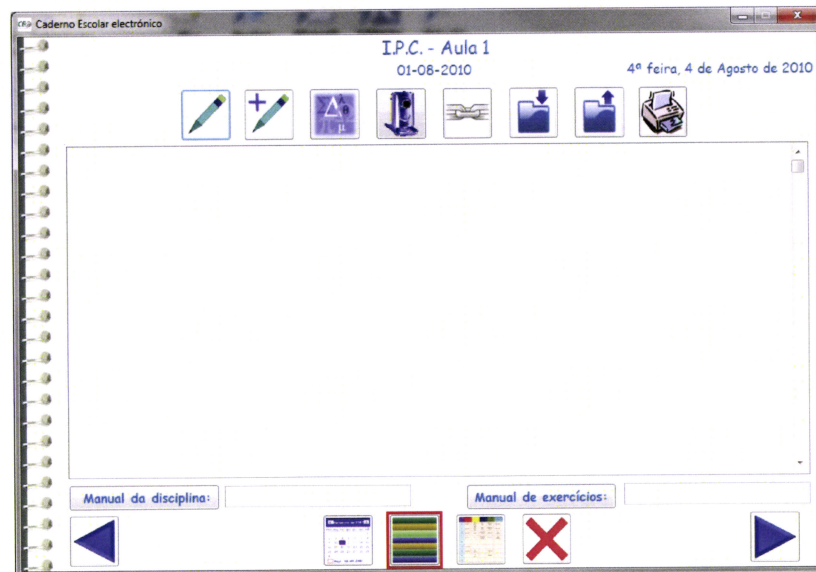


Figura 2.1: Interface de *notetaking* do CE-e

dificuldades, que este tipo de utilizadores incorrem, quando inseridos no ensino regular. A maior destas dificuldades é sem dúvida devido ao facto de terem de acompanhar um ritmo de aula, imposto por alunos sem qualquer tipo de dificuldade.

A interface do CE-e foi desenhada e construída de acordo com metodologias de desenho para sistema interactivos, sendo o seu desenho centrado no utilizador. Por isto, vários protótipos de baixa fidelidade foram construídos e avaliados, quer por eventuais utilizadores, quer por técnicos do Centro de Paralisia Cerebral de Beja. Ambos deram importantes contributos, para a usabilidade do sistema, contribuindo assim para que as interfaces do CE-e, fossem limpas e usáveis por utilizadores com necessidades especiais em contexto de aula.

2.2 Sistemas *Context-aware*

Muitos investigadores têm referido a importância e os benefícios que um conhecimento do contexto pode trazer à usabilidade dos sistemas informáticos [8], podendo desta forma aumentar e enriquecer a Interação Pessoa-computador.

A principal ideia por detrás do *context-aware* ou *context-awareness*¹, resulta da assumpção que os computadores podem "sentir" e reagir com base nas alterações do

¹ Conhecimento ou consciência do contexto envolvente.

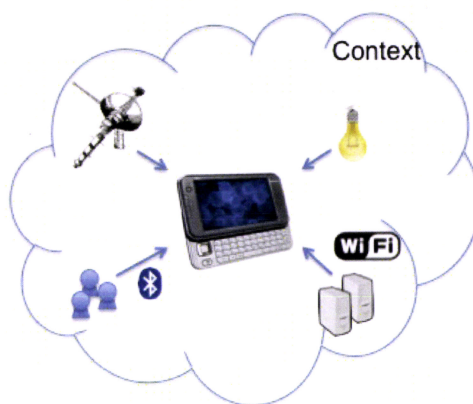


Figura 2.2: Dispositivo do tipo PDA, com mecanismos de reconhecimento do contexto

ambiente envolvente. Para que esta reacção seja possível os sistemas computacionais terão de possuir um conjunto de informação proveniente de estímulos, informação anterior que em conjunto com regras pré-estabelecidas, permitirão ao sistema reagir com uma consequência lógica e útil à situação do momento [29, 30].

Um sistema *context-aware*, pode e deve tentar fazer suposições sobre a envolvente do utilizador. Para Dey [11] o contexto é "qualquer informação que possa ser utilizada para caracterizar uma qualquer situação de uma entidade".

Inicialmente o contexto foi apenas entendido como sendo a localização, mas nos últimos anos Dey propôs que a localização não fosse entendida como o contexto da situação, mas como um dos componentes do contexto. Dey propôs ainda que o contexto e uma adaptação ao mesmo, pudesse ser utilizada para a construção de interfaces adaptativas e inteligentes, nas quais a interacção seja o mais implícita possível.

Muitos dos estudos e tentativas de adaptação ao contexto têm passado pela utilização de dispositivos de localização por satélite. Estes sistemas apenas utilizam uma pequena componente de um relativamente grande universo de possíveis adaptações ao contexto. Um sistema que apenas utiliza a localização para se adaptar ao contexto, não pode ser considerado um sistema *context-aware*, devendo ser considerado um sistema *location-aware* [5]. As aplicações com maior divulgação neste campo são os dispositivos de navegação por GPS utilizados nos veículos automóveis e os dispositivos do tipo guia turístico, normalmente dispositivos do tipo PDA ou *smartphone*. Estes sistemas efectuem uma adaptação dos conteúdos que exibem em função do local onde se encontram (Figura 2.2).

O *Context-awareness* é maioritariamente utilizado, para além dos sistemas de localização por satélite, em sistemas de computação ubíqua². Neste tipo de sistemas torna-se possível a utilização de sensores e outros dispositivos que captam com precisão o contexto, podendo assim propor e efectuar alterações de luminosidade ou da abertura das janelas em edifícios, por forma a minimizar os consumos energéticos.

2.3 *Context-aware*

No entanto um verdadeiro sistema *context-aware* terá obrigatoriamente de correlacionar outras variáveis, para além da posição (**onde**) e a identidade (**quem**), só assim poderá efectuar uma verdadeira contextualização do ambiente onde se encontra, podendo assim efectivamente executar tarefas que se adequem à realidade que o utilizador está a viver [12].

Apesar da definição de contexto continuar a ser alvo de muitas investigações, é certo que para um sistema poder obter uma "consciência" do contexto terá obrigatoriamente de entrar em linha de conta, para além do **onde** e de **quem**, com as seguintes variáveis:

- **Quando** - A grande maioria das aplicações que possuem alguma adaptação ao contexto, não contextualizam a passagem do tempo. A excepção pode ocorrer apenas e só para o registo do tempo que um determinado utilizador ou pessoa passa num determinado local. O grande interesse que possa existir no tratamento da variável **quando**, reside no grande interesse que as pequenas mudanças que ocorrem no tempo, podem possuir como elo fundamental para interpretar a actividade e rotinas humanas. Por exemplo se uma determinada pessoa apenas efectua curtas visitas a determinadas páginas Web, isto poderá significar que as mesmas são pouco interessantes para o mesmo. Assim se for possível estabelecer uma linha temporal que se relacione com acções, podem facilmente ser detectados estados ou acções que antecipem uma determinada acção ou que por outro lado violem o padrão que é habitual seguir;
- **O quê** - A percepção e interpretação da actividade humana é algo que se apresenta como extremamente difícil. Apesar desta dificuldade, sistemas ou aplicações com adaptação ao contexto terão obrigatoriamente de incorporar sistemas que interpretem a actividade humana, só desta forma poderão efectivamente fornecer ou sugerir acções ou informações úteis ao utilizador,

² É um modelo de interacção pessoa-computador em que a computação está integrada nos objectos do dia-a-dia.

enriquecendo desta forma a interacção com a máquina, aliviando o utilizador de acções que não acrescentam valor ao seu trabalho. A estratégia mais óbvia para vencer a dificuldade que esta variável apresenta, será a incorporação da informação ou da aplicação que o utilizador está a utilizar no ambiente virtual. O exemplo clássico da utilização deste tipo de informação são os *cookies* que inúmeras páginas Web utilizam para descrever e guardar as preferências e as actividades dos utilizadores de sistemas Web. No entanto poderemos ter maiores e melhores resultados se conseguirmos interpretar qual a intenção do utilizador no mundo real, esta sim apresentando-se como algo bastante mais complexo;

- **Porquê** - Se a variável **o quê** se apresenta como extremamente ambiciosa e difícil de captar, a variável **porquê** ainda se apresenta como mais difícil, porque poderemos ter muito melhores resultados na adaptação ao contexto se conseguirmos perceber **o que** alguém está a fazer e **porque** o faz.

Para o sistema em causa (Caderno Escolar Electrónico), não poderemos utilizar sistemas que causem intrusão, assim dispositivos como sensores que possam medir certos valores do utilizador ou do ambiente estão à partida descartados, devendo apenas ser utilizados dispositivos de captação que não acrescentem complexidade (aparente) ao sistema.

Os sistemas computacionais que todos nós utilizamos no nosso dia-a-dia estão construídos para efectuar tarefas de uma forma rotineira, ou seja marcam claramente o seu início e fim, bem como o início e o fim de todas as sub-tarefas que possam lançar durante o seu funcionamento, senão vejamos: Os processadores de texto (Figura 2.3), estão ajustados para iniciar o seu funcionamento com um documento em branco, um *template* ou com um documento previamente iniciado. As tarefas subsequentes serão a introdução de texto, a sua formatação, a sua impressão e o guardar das alterações feitas antes de fechar o documento. Este tipo de aplicações não possuem capacidade de adaptação, por exemplo a um sistema de controlo de versões em que o texto é reaproveitado de versões anteriores de acordo com uma linha temporal.

Podemos então dizer que uma aplicação ou sistema que pretenda efectuar uma real adaptação ao contexto terá de entrar em linha de conta com os seguintes factores:

- As experiências do dia-a-dia terão obrigatoriamente de ser captadas e armazenadas para análise, para que em futuros acessos o sistema possa sugerir e ajudar o utilizador na sua utilização;
- As actividades do dia-a-dia não estão organizadas como a maioria dos sistemas

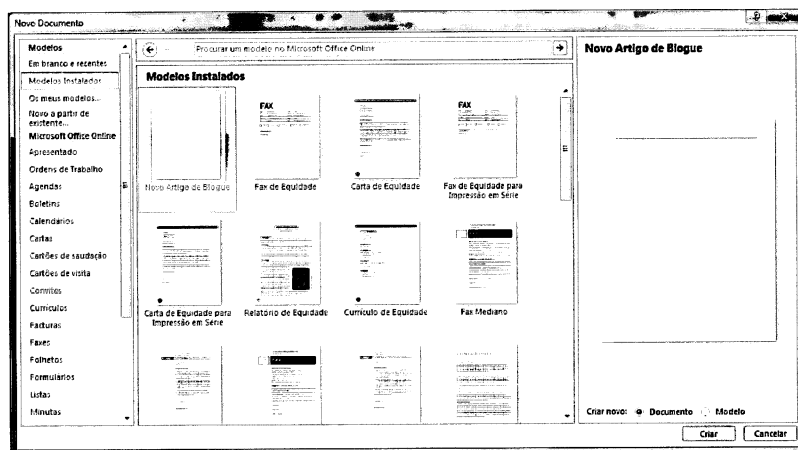


Figura 2.3: Assistente para criação de novo documento do MS Word

computacionais que utilizamos, ou seja, raramente têm um início, um meio e um fim, assim qualquer sistema *context-aware* terá de possuir uma grande flexibilidade e simplicidade de acções para que rapidamente se adapte a novas realidades, que resultem da troca de atenção do utilizador entre actividades concorrentes (resolução de um exercício e a consulta do manual da disciplina);

- O tempo é um factor de grande importância na caracterização da relação entre a actividade humana e a máquina;
- Diversos modelos de informação terão de ser associados e correlacionados para que possa existir um verdadeiro reaproveitamento da informação relativa a decisões passadas.

2.3.1 Dimensões do Contexto

Tal como já foi referido, só se consegue contextualizar eficazmente o ambiente circundante se conseguirmos definir as variáveis do contexto, mas para além das variáveis do contexto teremos que obrigatoriamente trabalhar com as dimensões deste mesmo contexto [5, 8].

As dimensões do contexto poderão ser várias, no entanto e segundo diversos autores, as mais comuns são a localização, as rotinas e fala, sendo também estas as que mais se adequam à realidade do Caderno Escolar Electrónico.

Assim, de uma forma macro, o Caderno Escolar Electrónico poderia utilizar ou prever a utilização das seguintes dimensões:

- **Localização** (onde) - Obrigaria à utilização de um dispositivo de localização por satélite (e.g. receptor de GPS), no entanto estes sistemas não funcionam dentro de casa e assim para o sistema efectivamente funcionar o sistema teria de funcionar durante as deslocações, o que não seria muito prático. Como o sistema pretende ser uma ajuda em contexto de aula esta dimensão ficará salvaguardada se o sistema reconhecer o dia e a hora (analisado no próximo ponto);
- **Hora e dia** (tempo) - Não sendo algo demasiado complexo, terá de ser analisado convenientemente para que o sistema possa retirar todas as potencialidades desta dimensão. Será muito útil para uma das adaptações obvias ao contexto - Abertura do caderno mais indicado para a aula de uma determinada hora;
- **Horário escolar** - Em conjunto com o ponto anterior poderá facilmente permitir ao sistema a abertura do caderno mais indicado para uma determinada hora ou determinada disciplina ou área de estudo;
- **Rotinas de utilização do sistema** - As rotinas de utilização do sistema podem ser conseguidas através do registo das acções efectuadas associadas ao tempo, que poderiam ser utilizadas das seguintes formas:
 - * Utilização de algoritmos de aprendizagem (e.g. Árvores de Decisão, Vector Machines, Redes de Markov, entre outros) que permitam ao sistema evoluir o seu tipo de adaptação consoante as opções mais escolhidas pelo utilizador;
 - * Simplesmente para recolha de dados e futuras evoluções do sistema;
- **Fala** - Poderá ser uma grande mais valia na adaptação ao contexto, dado que se o sistema conseguir identificar palavras, expressões ou frases chave poderá em conjunto com outra informação proveniente do tempo e/ou do horário escolar, sugerir ou efectuar acções efectivamente úteis. Tal como no ponto anterior, as formas de utilização são idênticas.

2.4 Paradigmas sobre Adaptabilidade e Previsibilidade

Os primeiros sistemas computacionais tinham como objectivo a resolução de problemas de índole científica ou comercial. Nos dias de hoje todos nós vivemos rodeados de sistemas informáticos, uns mais visíveis outros mais ubíquos, sendo a interacção cada vez mais intensa, obrigando por isso a cada vez mais acções por parte do utilizador.

Ainda nos primeiros sistemas computacionais, a interface era algo sem grande importância no desenvolvimento de um sistema computacional, sendo esta uma mera ferramenta

para fazer a ligação entre a máquina e o utilizador, mas proporcionalmente ao aumento da complexidade dos sistemas, também as interfaces tiveram de evoluir, tornando-se por vezes altamente complexas ou demasiadamente complicadas ou recheadas de um sem número de *widgets*, que não favorecem a interacção nem favorecem o entendimento das funcionalidades que o sistema e a respectiva interface têm a oferecer.

Alguns paradigmas foram sendo estudados e investigados para repor o nível de performance do utilizador no sistema. Entre outros, as interfaces adaptativas e preditivas, as interfaces atentas e os sistemas Web adaptativos, são abordagens que incorporam ideias e conceitos semelhantes aos que se pretendem implementar neste projecto, sendo por este motivo melhor explicados nos pontos seguintes, em busca de ideias pertinentes para incorporação na Interface Inteligente para o Caderno Escolar Electrónico.

2.4.1 Interfaces Adaptativas e Preditivas

"Uma interface adaptativa é um sistema computacional interactivo que melhora a sua capacidade de interacção com o utilizador, com base em experiências parciais de interacção com esse mesmo utilizador." (Langley, 2006)

Dois dos problemas que mais afectam a performance dos utilizadores de sistemas computacionais são a grande quantidade de informação que as interfaces possam apresentar e o facto de terem de tomar decisões em curtos espaços de tempo [25].

Apesar de nos últimos anos terem sido dados passos significativos no que toca à Interacção Pessoa-computador, muitos sistemas computacionais ainda apresentam interfaces de utilizador, complicadas e desajustadas. Esta situação resulta do facto de em muitos destes sistemas a interface não ser tida como parte do sistema, sendo vista como um mero pacote de software do sistema cuja única função é interligar o sistema e o utilizador. Assim, muitas das interfaces não integram no seu desenho princípios de desenho centrado no utilizador, nem estão suficientemente integradas com o sistema, não proporcionando assim uma boa interacção do utilizador com a máquina. Desta desadequação resulta uma não optimização da experiência e do desempenho do utilizador.

Para ultrapassar este problema foram criadas metodologias de desenho de interfaces,

nas quais as interfaces são desenhadas à medida do utilizador ou então o desenho tenta ser o mais adequado possível a todos os possíveis utilizadores do sistema, mesmo que para atingir o grau de adequação óptimo, se tenha de em utilização configurar alguns aspectos da interface.

Mas o facto de adoptarmos uma metodologia de desenho centrado no utilizador, não nos garante que a interface possa ser igualmente usável por todos os utilizadores, nomeadamente se esses utilizadores forem portadores de algum tipo de deficiência. Uma possível resposta para este problema poderá ser a construção de interfaces adaptativas.

A ideia base em que assentam as interfaces adaptativas é muito simples: "Em vez de ser o utilizador a adaptar-se ao sistema, adapta-se este ao utilizador."

Apesar de assentar em cima de um pressuposto, os problemas inerentes são em grande número e complexidade. Uma interface adaptativa tem de se basear em modelos cognitivos e no ambiente circundante [25]. Estes modelos tentam explicar o nível de perícia e experiência do utilizador, através de parâmetros do tipo; comandos efectuados, taxas de erro, velocidade de digitação, entre outros.

Outro grande problema reside no diálogo, ou seja, o diálogo entre a interface e o utilizador deve ser o mais específico possível para o utilizador (em causa). Finalmente, a interface tem obrigatoriamente de ser pensada como um componente do sistema e não como uma mera peça que interliga o sistema e o utilizador.

Uma interface adaptativa pode efectuar as adaptações de duas formas:

- A primeira hipótese de adaptação é uma adaptação relativa ou balizada, ou seja, é efectuada uma adaptação limitada deixando nas mãos dos utilizadores a readaptação da interface, no caso da adaptação efectuada pela interface não ser satisfatória;
- A segunda hipótese é deixar totalmente nas mãos da interface a responsabilidade pela adaptação.

A segunda hipótese é por muitos considerada como a verdadeira adaptabilidade, sendo no entanto muito difícil de atingir. Uma adaptação totalmente dinâmica apenas conseguirá ser atingida à custa de algoritmos de aprendizagem automática (do acrónimo Inglês, *machine learning*).

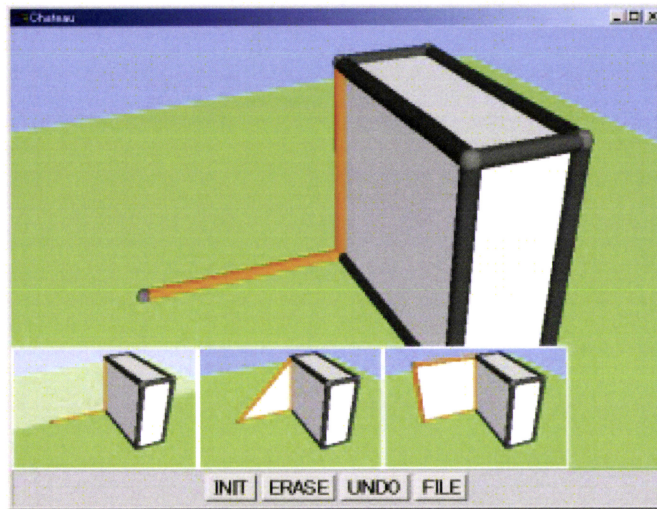


Figura 2.4: Interface sugestiva de [17]

O conceito por trás da aprendizagem automática pode ser por si só considerado como adaptativo, porque os algoritmos deste tipo baseiam-se em informação anterior para tomarem decisões sobre dados actuais.

Contudo uma interface adaptativa também tem os seus problemas e entre outros poderemos referir o facto de o utilizador não conseguir criar um modelo mental do sistema, se a aparência do sistema sofrer muitas e frequentes alterações. Outro potencial problema poderá ser a perda de controlo, ou a potencial perda de controlo que o utilizador poderá sentir. Por último e talvez o maior problema das interfaces adaptativas, que acaba por ser o custo e a complexidade da implementação.

Para que as adaptações sejam úteis, o sistema deve ser capaz de:

- Conhecer, distinguir e caracterizar os vários utilizadores do sistema. Esta distinção é passível de ser efectuada através da definição e construção de modelos de utilizadores. Um modelo de utilizador é um conjunto de informação mantida pelo sistema que caracteriza um utilizador [25, 13];
- Conhecer o tipo de interacção em utilização e a forma de dialogo entre o utilizador e o sistema, para poder realizar uma adaptação que seja realmente útil;
- Conhecer o domínio da tarefa, porque o objectivo final de um utilizador é concluir uma determinada tarefa ou conjunto de tarefas e se o objectivo reside

no facto de termos uma interface que consiga realizar o maior número possível de adaptações, esta deve ser capaz de inferir a partir das acções do utilizador quais os seus objectivos no sistema;

- Conhecer o sistema, porque as adaptações que sejam efectuadas pela interface, devem estar dentro dos limites e capacidades do sistema.

2.4.2 Interfaces Atentas e Sugestivas

Outro paradigma sobre a adaptabilidade e previsibilidade, deriva da atenção e da sugestão. Neste caso as interfaces passam a estar atentas e automaticamente sugerem acções aos utilizadores.

Uma interface atenta estabelece automaticamente prioridades e de acordo com as mesmas apresenta ao utilizador a informação ou as opções mais adequadas ao contexto e ao momento de utilização do sistema, para que os recursos, quer do utilizador quer do sistema sejam optimizados, não sobrecarregando assim nenhum dos intervenientes na interacção [32]

Tal como nas interfaces adaptativas, as interfaces atentas baseiam as suas decisões num conjunto de informações e modelos sobre decisões anteriormente efectuadas pelo utilizador, estados anteriores e estados futuros.

No entanto as interfaces atentas podem-se subdividir em vários subtipos, de uma forma geral esta divisão é efectuada de acordo com a capacidade que as interfaces têm de monitorizar o estado de interacção e atenção do utilizador para com o sistema e a interface.

Nos sistemas cuja interacção é efectuada através de janelas, a atenção do utilizador é captada e comunicada ao sistema de uma forma **explicita** através da organização das janelas, por parte do utilizador de uma forma manual. Por outro lado sistemas que possuam mecanismos sensíveis, tais como sistemas de *eye tracking* para seguirem as acções do utilizador e perceberem onde a atenção do utilizador está centrada, são classificados como sistemas **implícitos**. Uma interface atenta tenderá a utilizar na sua estrutura uma combinação destes dois tipos de interfaces.

Por outro lado temos as interfaces sugestivas (Figura 2.4), que apenas fornecem ao utilizador pistas sobre a próxima possível acção do utilizador, através do realce de certos

componentes da interface [17]. Um utilizador ao interagir com uma interface sugestiva, apenas tem de escolher uma das opções em destaque ou pode ignorar todas as sugestões caso nenhuma corresponda à opção desejada. Por norma as sugestões são geradas por um sistema normalmente apelidado de motor de sugestão, que constantemente observa o estado do sistema, gerando então um conjunto de sugestões que correspondam aos padrões mais próximos do estado actual. Uma interface sugestiva pode ser vista como sendo um sistema de interacção por gestos, ou seja, ao invés de actualizar o sistema automaticamente, ao ser descoberto um padrão idêntico, a interface apresenta um conjunto de sugestões e pede ao utilizador que escolha uma destas.

Uma interface sugestiva é uma extensão às interfaces preditivas e está a ser encarada por muitos investigadores como a interface do futuro, para a grande maioria dos sistemas, nomeadamente os sistemas WIMP (Windows, Icons, Menus and Pointing Devices).

Capítulo 3

ARQUITECTURA DO CE-e ADAPTATIVO

A interface do protótipo funcional do CE-e foi desenhada de acordo com uma metodologia de desenho para sistemas interactivos com o focus no utilizador, sendo por isso bastante simples e intuitiva [3]. No entanto como o CE-e pretende fornecer o máximo de ajuda possível em tarefas de notetaking e de organização da informação, pretende-se que a interface possa aliviar, o máximo possível, o utilizador de tarefas rotineiras e repetitivas, o que poderá ser particularmente útil para utilizadores com graves problemas motores.

Assim, com base na experiência própria, na opinião de técnicos e de outros peritos, optamos por uma interface do tipo sugestivo, que apresentará ao utilizador a hipótese mais provável, de acordo com os padrões que a base de conhecimento apresente.

3.1 Estrutura da Base de Conhecimento

Tal como foi referido anteriormente, para sistema poder prever ou sugerir a acção mais provável para um determinado contexto, tem de se socorrer de conhecimento passado e modelos de interacção. Para a construção desta base de conhecimento, recorre-se frequentemente a sistemas que registam as acções do utilizador *logging*.

Há já algum tempo que investigadores que trabalham em sistemas de Comunicação Aumentativa e Alternativa (CAA), tentam medir o impacto dos avanços tecnológicos no acto de comunicar com recurso a este tipo de sistemas. Assim, de forma a facilitar a análise dos dados recolhidos, com base em sistemas de *logging*, foi definido um formato padrão que permite efectuar a análise aos dados de uma forma sistemática, o *Universal Logging Format* [21].

A estrutura do ficheiro de *logs* proposta em [21], apresenta três partes distintas:

id	name	type	notnull	def_value	pk
0	ID	INTEGER	0		1
1	date	NVARCHAR(40)	0		0
2	weekDay	NVARCHAR(20)	0		0
3	time	INTEGER	0		0
4	form	NVARCHAR(50)	0		0
5	discipline	NVARCHAR(50)	0		0

Figura 3.1: Estrutura dos logs do CE-e

- Cabeçalho que especifica o conteúdo e o formato dos registos;
- Corpo que é composto por n linhas, cada uma representativa dos dados relativos a um *log*;
- Secção de análise, sendo esta opcional, onde se podem colocar algumas estatísticas dos *logs*.

Os *logs* têm uma componente temporal, um *output* ou resultado, a acção efectuada, o tipo de dispositivo utilizado que originou o *log*, o tipo de comunicação ou escrita que estava a ser desenvolvida, o contexto (no caso de uma ferramenta CAA), que palavras antecederam o *log* e uma indicação de qual a página do sistema onde foi despoletado o *log*.

Com base neste formato, construímos uma estrutura de *log* que contém os seguintes campos: data, dia da semana, hora, página do sistema, disciplina activa e acção (Figura 3.1).

Com base em [16] e nas funcionalidades do XML, decidimos utilizar um sistema que permite a construção de *queries* ou interrogações à base de conhecimento. Assim o registo dos *logs* é efectuada com recurso a uma base de dados relacional, com um motor SQLite.¹ Este sistema apresenta como grandes vantagens o facto de não necessitar de ter um servidor de base de dados activo no sistema, eliminando assim qualquer pré-requisito para o registo dos *logs* e o facto de permitir construção de interrogações elaboradas em SQL *standard*.

Tal como já foi referido, os *logs* só teriam validade se fossem recolhidos em contexto real de aula e assim o CE-e foi colocado em testes em duas escolas. Os *logs* foram recolhidos de equipamentos portáteis pertencentes a alunos com paralisia cerebral, apenas afectados na sua parte física e integrados no ensino regular.

¹ Documentação disponível em: <http://www.sqlite.org/>

ID	Date	WeekDay	Time	Form	discipline	Event
1	26-02-2010	Friday	11:9:58	FormAgenda		iniciar_aplicacao
2	26-02-2010	Friday	11:10:56	FormAgenda		vista_de_livros
3	26-02-2010	Friday	11:11:6	BooksForm		abrir_caderno_Matemática
4	26-02-2010	Friday	11:11:14	NotesForm	Matemática	vista_livros
5	26-02-2010	Friday	11:11:16	BooksForm		abrir_caderno_Língua Portuguesa
6	26-02-2010	Friday	11:11:20	NotesForm	Língua Portuguesa	aceder_manual_disciplina
7	26-02-2010	Friday	11:12:21	NotesForm	Língua Portuguesa	aceder_manual_exercicios
8	26-02-2010	Friday	14:6:56	FormAgenda		iniciar_aplicacao
9	26-02-2010	Friday	14:7:11	FormAgenda		vista_de_livros
10	26-02-2010	Friday	14:7:16	BooksForm		abrir_caderno_Estudo do Meio
11	26-02-2010	Friday	14:7:22	NotesForm	Estudo do Meio	aceder_manual_disciplina
12	26-02-2010	Friday	14:56:21	NotesForm	Estudo do Meio	terminar_aplicacao
13	02-03-2010	Tuesday	9:18:29	FormAgenda		iniciar_aplicacao
14	02-03-2010	Tuesday	9:18:38	FormAgenda		vista_de_livros
15	02-03-2010	Tuesday	9:18:44	BooksForm		abrir_caderno_Matemática
16	02-03-2010	Tuesday	9:18:50	NotesForm	Matemática	nova_nota_de_texto
17	02-03-2010	Tuesday	9:26:49	NotesForm	Matemática	vista_livros
18	02-03-2010	Tuesday	9:26:53	BooksForm		abrir_caderno_Língua Portuguesa

Figura 3.2: Extracto de um ficheiro de *logs* do CE-e

Estes alunos, tal como já foi referido, possuem paralisia cerebral. Mas, a sua componente cognitiva não se encontra afectada de qualquer forma, assim a grande dificuldade que enfrentam numa sala de aula tradicional é o ritmo da aula, dado que a sua componente física encontra-se afectada, não permitindo que os mesmos possam acompanhar devidamente a aula e as matérias lecionadas. Nesta situação o CE-e torna-se uma ferramenta imprescindível para estes alunos, dado que não conseguem manejar os tradicionais instrumentos de escrita - o papel e o lapís.

Situando-se numa faixa etária compreendida entre os 8 e os 11 anos, estes utilizadores contribuíram de sobremaneira para os desenvolvimento do CE-e. A análise aos *logs* de utilização, permitiu retirar conclusões sobre as decisões de implementação que deveriam ser tidas em consideração durante a fase de análise e durante a codificação e implementação do sistema.

3.2 Os Algoritmos

Há já algum tempo que investigadores da área da Interação Pessoa-computador vêm a desenvolver estudos sobre a adaptabilidade de interfaces aos seus utilizadores, através de sistemas baseados em aprendizagem automática, tanto ao nível das interfaces tradicionais, como ao nível das interfaces Web [13, 19, 25].

Antes de se iniciar a real implementação do sistema, os dados foram sujeitos a ferramentas de aprendizagem automática, para se perceber qual, ou quais os algoritmos que poderiam trazer melhores resultados.

```

Dataset: C:\e_laure_2010-03-24
  Attribute: Day      {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}
  Attribute: Hour     numeric
  Attribute: Form     {UserFeedback, Form, FormAgenda, BookForm, BookForm, ScheduleViewForm, NoteForm}
  Attribute: Discipline {Áudio do Ensino, "Espelho do Rio", "Matemática", "Língua Portuguesa", ""}
  Attribute: Event    {"Formatar texto alinhamento centro", "Formatar texto cor", "Formatar texto highlight", "Inserir texto tamanho fonte",
  "Formatar texto sublinhado", "Formatar texto bold", "Novo evento", "Navegar vista configuracoes", "Editar nota texto",
  "Cancelar escolha nota texto", "Configurar manual disciplina", "Adicionar nota texto", "Adicionar disciplina", "Cancelar nota texto",
  "Formatar texto itálico", "Abrir manual disciplina", "Download ficheiro", "Adicionar hiperligacao", "Iniciar nova aula",
  "Abrir manual exercicio", "Terminar aplicacao", "Iniciar aplicacao", "Navegar vista livros", "Abrir caderno",
  "Escolha impressao objecto", "Cancelar escolha impressao objecto", "Impressao nota-livro", "Navegar aula anterior",
  "Navegar form agenda", "Navegar aula posterior", "Adicionar nota texto"}

data
  Thursday,10031,"FormAgenda","", "Navegar vista livros"
  Thursday,10034,"BookForm","", "Abrir caderno"
  Thursday,10049,"BookForm","Espelho do Rio", "Navegar vista livros"
  Thursday,10050,"BookForm","", "Abrir caderno"
  Thursday,10050,"NoteForm","Língua Portuguesa", "Iniciar nova aula"
  Thursday,10050,"NoteForm","Língua Portuguesa", "Adicionar hiperligacao"
  Thursday,10055,"FormAgenda","", "Navegar vista livros"
  Thursday,10055,"BookForm","", "Abrir caderno"
  Thursday,10090,"FormAgenda","", "Navegar vista livros"
  Thursday,10091,"BookForm","", "Abrir caderno"

```

Figura 3.3: Aspecto parcial do ficheiro submetido à ferramenta *Weka*

A aprendizagem automática divide-se em dois grandes tipos: Supervisionada e Não Supervisionada [4]; Como os logs recolhidos para a constituição da base de conhecimento, são constituídos pela componente tempo, por uma componente contexto e finalmente pela componente acção, ou seja, o objectivo que o utilizador tinha em mente, estamos perante um cenário de aprendizagem supervisionada em que a acção do utilizador será a classe. A aprendizagem supervisionada pode-se ainda dividir em problemas de classificação ou de regressão, sendo para o problema em causa uma questão de classificação, ou seja, partindo de uma base de conhecimento pretende-se obter a classe a que possam pertencer novos exemplos (novas acções), ou seja, conseguir prever a hipótese mais provável, dentro de um problema multiclasse.

Na posse de registos de utilização reais, já era possível começar a trabalhar na componente de "inteligência" do sistema, podendo assim dota-lo de uma interface sugestiva, que poderia provar a viabilidade da interface inteligente para o Caderno Escolar electrónico. Este tipo de testes é conhecido por prova de conceito.

A ferramenta Weka [15] foi escolhida para testar a base de conhecimento em busca de padrões que pudessem justificar a real implementação do sistema. Esta ferramenta apresenta implementações da grande maioria dos algoritmos de aprendizagem automática, nomeadamente, árvores de decisão. Foi sobre as árvores de decisão que recaiu a primeira escolha, dado que o sistema pretendido terá de trabalhar em tempo real, efectuando consultas à base de conhecimento e apresentando de seguida a hipótese que mais se adequa ao contexto em causa e à componente temporal do momento.

Os primeiros testes efectuados, com recurso a uma árvore de decisão C4.5 [26], demonstraram que poderiam ser atingidos resultados com uma percentagem mínima de acertos na ordem dos 50%, valor que justificava o esforço de implementação, dado

que o sistema possui, cerca de 40 classes. No entanto destas 40 opções que o sistema possui, dependendo do ecrã onde o utilizador se encontra, apenas são apresentadas entre 3 a 17 opções, o que em termos médios nos dará a cada momento 7 opções possíveis, sendo que 50% de 7 dará aproximadamente 4. Uma percentagem de acertos de 4 em 7 é um resultado bastante encorajador.

A árvore de decisão C4.5, foi desenvolvida por Ross Quilan [26] e, é um melhoramento da sua predecessora, a árvore ID3. O algoritmo C4.5 constrói uma árvore de decisão utilizando o conceito de entropia². Esta entropia é obtida através de um conjunto de treino, através do qual o classificador é treinado. Os dados que submetemos a uma árvore C4.5, são tratados como amostras e cada amostra tem um conjunto de características e uma classe associada e, em cada nó o algoritmo escolhe o atributo que melhor se enquadra nas características da classe em causa. A escolha deste atributo é efectuada através da diferença na entropia, normalmente conhecida por ganho³. O conceito de aprendizagem, basea-se no tipo "aprendizagem por indução" ou seja, qualquer hipótese que se aproxime correctamente do objectivo num determinado conjunto (treino), também deverá aproximar-se correctamente do mesmo objectivo num outro conjunto de dados não observado (teste) [24].

Outros algoritmos de classificação foram testados em busca de melhores resultados ou de resultados que confirmassem os obtidos com recurso a árvores de decisão, nomeadamente algoritmos baseados em redes de Markov, mais especificamente modelos "escondidos" de Markov.

Este último algoritmo acabou por ser o escolhido, para construir o motor de inferência do sistema, facto que será mais à frente explicado.

O motor de inferência do sistema lidará com um conjunto pequeno de atributos, como se pode ver na figura 3.3 o motor terá apenas quatro atributos que necessita de analisar, dado que o quinto atributo é o atributo classe, que servirá para treinar o classificador. No entanto este classificador terá de lidar com um sistema multi-classe, sendo as classes em grande número. A matriz de confusão construída pela ferramenta Weka (figura 3.4) demonstra de uma forma bastante clara o número e dimensão das classes, sendo este um problema para o classificador, dado que a árvore resultante possuirá um grande número

² É uma medida de incerteza, associada a uma variável aleatória, que quantifica, na forma de um valor provável, a informação contida num determinado conjunto de dados.

³ Medida não simétrica, que resulta da diferença entre duas distribuições probabilísticas.

sistema poderá estar, tendo apenas um conhecimento probabilístico de onde o mesmo deveria estar [27].

Um Modelo "Escondido" de Markov, é normalmente constituído por um conjunto finito de estados, geralmente associados a uma distribuição probabilística, num espaço multidimensional, encontrando-se próximo do estado provável do sistema dentro de uma distribuição probabilística, em que o estado seguinte depende de n estados passados. Foi esta definição genérica que serviu como base de decisão, para a utilização, nesta fase do projecto, deste tipo de algoritmo, isto porque, no caso do CE-e todas as próximas acções poderão depender de n acções passadas, sendo certo que a próxima acção mais provável será uma de m possíveis.

De uma forma menos genérica temos:

- N , que representa o número de estados possíveis do modelo/sistema;
- M , que representa o número de observações;
- Λ , que representa o conjunto das várias probabilidades de transição entre estados $\Lambda = \{a_{ij}\}$, onde:

$$a_{ij} = p \{q_{t+1} = j | q_t = i\}, 1 \leq i, j \leq N,$$

onde q_t indica o estado actual.

As probabilidades de transição entre estados, devem satisfazer todas as restrições dos modelos estocásticos,

$$a_{ij} \geq 0, 1 \leq i, j \leq N,$$

e também,

$$\sum_{j=1}^N a_{ij} = 1, 1 \leq i \leq N$$

Na definição de Modelo "Escondido" de Markov, as probabilidades de transição entre estados são definidas (como vimos anteriormente) por:

$$a_{ij} = p \{q_{t+1} = j | q_t = i\},$$

O que implicitamente assume, que o estado seguinte depende apenas do estado corrente, no entanto o que normalmente acontece e acontece também no caso do Caderno Escolar

Electrónico, é que o estado seguinte pode depender não só do estado actual, como de um variado número de estados passados k , o que altera a definição das probabilidades de transição entre estados para:

$$a_{i_1 i_2 \dots i_k j} = p \{ q_{t+1} = j | q_t = i, q_{t-1} = i_2, \dots, q_{t-k+1} = i_k \}, 1 \leq i_1, i_2, \dots, i_k, j \leq N.$$

Contudo, este algoritmo possui três problemas, que conforme o tipo de problema que estamos a tentar resolver, merecem a nossa atenção, que são:

- **O Problema de Avaliação**, em que dado um modelo λ e uma sequência de observações $O = o_1, o_2, \dots, o_n$, qual é a probabilidade destas observações serem geradas pelo modelo $p \{ O | \lambda \}$;
- **O Problema da Descodificação**, em que dado um modelo λ e uma sequência de observações $O = o_1, o_2, \dots, o_n$, qual é a sequência de estados, mais provável, no modelo que produziu as observações;
- **O Problema da Aprendizagem**, em que dado um modelo λ e uma sequência de observações $O = o_1, o_2, \dots, o_n$, como poderemos ajustar os parâmetros $\{ \Lambda, B, \pi \}$ do modelo, de modo a maximizar $p \{ O | \lambda \}$, onde:
 $B = \{ b_j(k) \}$, é a distribuição probabilística em cada um dos estados, sendo:

$$b_j(k) = p \{ o_t = v_k | q_t = j \}, 1 \leq j \leq N, 1 \leq k \leq M;$$

e, $\pi = \{ \pi_i \}$, sendo:

$$\pi_i = p \{ q_i = i \}, 1 \leq i \leq N;$$

Como no caso do Caderno Escolar Electrónico, queremos treinar o classificador para que este seja utilizado em tarefas de reconhecimento de padrões, teremos de resolver este problema.

Na sua essência, o problema da aprendizagem reside na forma como os parâmetros do modelo λ devem ser ajustados, para que o conjunto de treino seja representado da melhor forma possível. Escusado será dizer, que podem existir diversos critérios de aprendizagem, dependendo do tipo de aplicação ou modelos que estamos a utilizar.

Existem, principalmente, dois critérios de optimização:

- Critério da Probabilidade/Semelhança Máxima;
- Critério da Máxima Informação Mútua.

Como o Critério da Máxima Informação Mútua, não aplica o conceito de "discriminação"⁴, não sendo assim de grande interesse para o problema em causa - reconhecimento de padrões, optamos então pelo Critério da Probabilidade/Semelhança Máxima.

O Critério da Máxima Semelhança, tenta maximizar a probabilidade de uma determinada sequência de observações O^w , pertencentes a uma determinada classe w , dado um modelo λ_w pertencente à classe w e com os parâmetros do mesmo modelo. Esta probabilidade é a semelhança máxima das observações. De uma forma matemática, temos:

$$L_{max} = p\{O^w|\lambda_w\},$$

Se considerarmos apenas uma classe a cada momento, podemos simplificar a fórmula da seguinte forma:

$$L_{max} = p\{O|\lambda\}$$

Contudo não existe uma maneira conhecida de analiticamente resolver o modelo $\lambda = \{\Lambda, B, \pi\}$, por forma a maximizar a probabilidade L_{max} , no entanto podemos escolher os parâmetros, por forma a que pelo menos localmente esta probabilidade seja maximizada. Para o efeito, teremos de utilizar um procedimento iterativo, como o algoritmo *Baum-Welch*.

O algoritmo *Baum-Welch*, também conhecido como algoritmo *Forward-backward* (para a frente e para trás), utiliza uma técnica de contagem de ocorrências dos argumentos, de acordo com a seguinte fórmula:

$$Q(\lambda, \bar{\lambda}) = \sum_q p\{q|O, \Lambda\} \log [p\{O, q, \bar{\lambda}\}],$$

sobre o complementar do modelo ($\bar{\lambda}$). Uma característica particular deste algoritmo é a convergência garantida.

Para podermos descrever este algoritmo, precisamos de definir variáveis auxiliares:

- A primeira variável que precisamos de definir, é a probabilidade de estarmos perante um estado i , num tempo t e num estado j num tempo $t + 1$. Matematicamente teremos o seguinte:

$$E_t(i, j) = p\{q_t = i, q_{t+1} = j|O, \lambda\},$$

⁴ Particionamento de amostras com padrões semelhantes



o que é o mesmo que termos:

$$E_t(i, j) = \frac{p\{q_t=i, q_{t+1}=j, O|\lambda\}}{p\{O|\lambda\}},$$

utilizando as variáveis *Forward* e *backward*, poderemos então expressar esta probabilidade da seguinte forma:

$$E_t(i, j) = \frac{\alpha_t(i)a_{ijt+1}(j)b_j(o_{t+1})}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i)a_{ijt+1}(j)b_j(o_{t+1})} \quad (i)$$

- A segunda variável é denominada a variável "à posteriori", que podemos definir da seguinte forma:

$$\tau_t(i) = p\{q_t = i | O, \lambda\},$$

que é a probabilidade de estarmos perante um estado i , num tempo t , dada uma sequência de observações e um modelo. Assim, as variáveis *forward* e *backward* podem ser expressadas, matematicamente, da seguinte forma:

$$\tau_t(i) = \left[\frac{\alpha_t(i)_t(i)}{\sum_{i=1}^N \alpha_t(i)_t(i)} \right],$$

assim a as duas variáveis relacionam-se da seguinte forma:

$$\tau_t(i) = \sum_{j=1}^N E_t(i, j), 1 \leq i \leq N, 1 \leq t \leq M \quad (ii)$$

Neste momento é então possível descrever o processo de aprendizagem do algoritmo *Baum-Welch*, no qual os parâmetros do Modelo "Escondido" de Markov são actualizados, de forma a maximizar a quantidade $p\{O|\lambda\}$.

Partindo do pressuposto que o modelo inicial $\lambda = (\Lambda, B, \pi)$, poderemos então calcular α e β , com as seguintes fórmulas recursivas:

$$\beta_t(i) = \sum_{j=1}^N t_{t+1}(j)a_{ij}b_j(o_{t+1}), 1 \leq i \leq N, 1 \leq t \leq T - 1$$

e,

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^N \alpha_t(i)a_{ij}, 1 \leq j \leq N, 1 \leq t \leq T - 1$$

O E e τ podem ser calculados utilizando as fórmulas vistas em (i) e (ii). Finalmente estamos em condições de actualizar os parâmetros do modelo com as chamadas fórmulas de re-estimação:

$$\bar{\pi}_i = \tau_1(i), 1 \leq i \leq N$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} E_t(i, j)}{\sum_{t=1}^{T-1} \tau_t(i)}, 1 \leq i \leq N, 1 \leq j \leq N$$

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T \tau_t(j)}{\sum_{t=1}^T \tau_t(j)}, 1 \leq j \leq N, 1 \leq k \leq M$$

3.3 As Tecnologias Utilizadas

Nesta secção fazemos uma breve apresentação das tecnologias utilizadas no desenvolvimento do sistema.

3.3.1 Linguagem de Programação C#

A linguagem de programação C# (lê-se C Sharp), criada pela Microsoft é inspirada nas melhores ideias e pormenores de algumas das linguagens de programação mais bem-sucedidas dos últimos anos, nomeadamente a linguagem C, C++ e Java. A possibilidade de esconder funções de baixo nível e ponteiros, ou pelo contrário, utilizar estas ferramentas de uma forma explícita, confere-lhe uma flexibilidade e poderes muito grandes [18].

Outra vantagem desta linguagem, é o pormenor de ser baseada na *.Net framework*. Esta *framework* é composta por quatro partes distintas: (i) *Common Language Runtime*; (ii) Um conjunto de bibliotecas e classes (Figura 3.5); (iii) Um conjunto de linguagens de programação, onde se inclui o C#; e (iv) O ambiente *asp.net*. Esta *framework* tem três objectivos principais:

- Tornar as aplicações desenvolvidas para ambientes Windows muito mais flexíveis, seguras e fiáveis;
- Simplificar o desenvolvimento de aplicações;

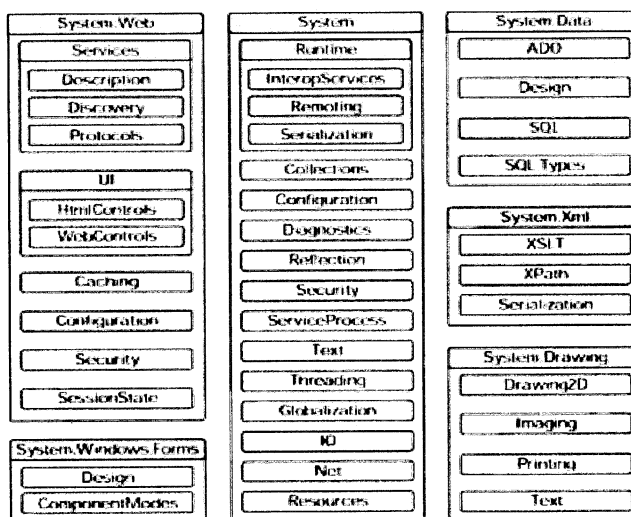


Figura 3.5: Bibliotecas e classes da *.Net Framework*

- Fornecer um conjunto de bibliotecas que funcionassem em várias linguagens.

O desenvolvimento de aplicações Windows com recurso à *.Net Framework*, é baseado em *Windows Forms*, sendo estas um "cruzamento" entre as formas do Visual Basic 6 e as formas do Visual C++. Apesar muito parecidas com as suas predecessoras, as formas da *.Net Framework*, que o C# utiliza, são no entanto totalmente orientadas a objectos e baseadas em classes.

Estas novas formas suportam para além dos tradicionais controlos, como é caso dos botões e caixas de texto, suportam ainda controlos do tipo ActiveX⁵. A segurança também não foi esquecida, de tal forma que se pode definir muitas medidas de segurança, tais como a capacidade que uma aplicação tem de escrever em disco.

Finalmente o *Common Language Runtime* (CLR), consiste num runtime unificado que pode ser utilizado pela *.Net Framework* e todas as linguagens suportadas por esta. O CLR fornece capacidades de gestão de memória, segurança, interoperabilidade e tratamento de erros. Antes do CLR, as linguagens de programação tinham embutido o seu próprio runtime environment, para além do compilador, o que não permitia que uma linguagem pudesse utilizar o que de melhor outras linguagens pudessem oferecer, dado que o runtime era unicamente seu.

⁵ É uma *framework* para definição para a definição de componentes de software, conhecidos como controlos, independentes da linguagem de programação em que são implementados e que se destinam a realizar uma determinada função específica.

```

<?xml version="1.0" encoding="UTF-8" ?>
<disciplines xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <discipline>
    <discp_name>Matematica</discp_name>
  </discipline>
  <discipline>
    <discp_name>F. Civica</discp_name>
  </discipline>
  <discipline>
    <discp_name>Est. Meio</discp_name>
  </discipline>
  <discipline>
    <discp_name>Inglês</discp_name>
  </discipline>
  <discipline>
    <discp_name>L. Port.</discp_name>
  </discipline>
  <discipline>
    <discp_name>Leitura</discp_name>
  </discipline>
  <discipline>
    <discp_name>Expressoes</discp_name>
  </discipline>
</disciplines>

```

Figura 3.6: Exemplo de um ficheiro XML

3.3.2 XML - eXtensible Markup Language

O XML (eXtensible Markup Language), é uma forma muito simples e flexível de formatação de texto, derivada do SGML⁶. Originalmente foi desenhada para fazer face às grandes quantidades de dados que começaram a circular nos sistemas de informação e, também para permitir portabilidade dos dados e da informação, num grande número de tecnologias, principalmente na Internet.

A especificação do XML é definida pelo grupo de trabalho da W3C⁷, e pode ser definida como um conjunto de regras para codificar informação numa linguagem que seja facilmente descodificada pela máquina, sendo ao mesmo tempo interoperável.

O XML possui linguagens de descrição, como se pode ver na Figura 3.6, nas quais são descritas as estruturas de um documento XML. Uma das linguagens de descrição mais utilizada é o XML Schema Definition (XSD). Como se pode ver na Figura 3.7, o XSD é baseado em XML, fornecendo diversas funcionalidades, como o relacionamento de dados dentro de um documento XML, a construção de tipos próprios derivados de tipos básicos e principalmente consegue suprir as limitações da primeira linguagem de descrição para o XML, o DTD (Document Type Definition).

⁶ Standard Generalized Markup Language (ISO 8879:1986), é uma metalinguagem através da qual se pode descrever e definir linguagens de marcação para documentos.

⁷ O World Wide Web Consortium é o organismo internacional encarregue do desenvolvimento de standards para a Web. É liderado pelo inventor da Web, Tim Berners-Lee e têm como missão extrair todo o potencial possível da Web.


```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!--Definition of simple elements-->
  <xs:element name="discp_name" type="xs:string"/>

  <!--Definition of complex types-->
  <xs:element name="discipline">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="discp_name"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="disciplines" >
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element ref="discipline"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Figura 3.7: Exemplo de um ficheiro XSD

O DTD, já existia antes do aparecimento do XML, sendo utilizado desde o surgimento do SGML, para a definição de regras sobre quais as etiquetas que podem ser utilizadas num documento.

3.3.3 SQLite

O SQLite é uma biblioteca de software que implementa um motor de base de dados relacional, independente, sem servidor activo no sistema e sem configurações, estando o seu código fonte sob domínio público.

O SQLite é desenvolvido em Linguagem de programação C e ao contrário de muitas outra base de dados SQL, não tem um servidor activo separado. O sistema lê e escreve directamente em ficheiros normais. É perfeitamente interoperável e independente da plataforma.

O funcionamento do SQLite é bastante simples, em vez do sistema correr num processo servidor separado, a livreria SQLite é embebida, tornando-se assim parte integral do software aplicativo, utilizando chamadas a funções para interagir com o SGBD, sendo assim a latência do acesso à base de dados muito reduzida.

O SQLite consegue armazenar todas as funcionalidades de um SGBD relacional, tais

como tabelas e índices, num único ficheiro independente da plataforma na máquina na qual reside o software aplicacional.

3.4 A Codificação

A codificação do motor de inferência desenrola-se em torno de dois algoritmos, uma árvore C5, que deriva da árvore C4.5 [26] e, um algoritmo baseado em modelos "escondidos" de Markov. Apesar do primeiro ter apresentado melhores resultados nos testes realizados na ferramenta *Weka*, a codificação do sistema avançou com recurso ao segundo algoritmo. Esta decisão foi motivada pela urgência de possuímos um protótipo pronto antes do início do ano escolar, para que se pudesse comprovar a real utilidade da Interface Inteligente do Caderno Escolar electrónico.

Apesar do motor de inferência deste primeiro protótipo da Interface Adaptativa do Caderno Escolar electrónico, ter sido implementado com recurso ao algoritmo baseado em modelos "escondidos" de Markov, temos como objectivo final a implementação do sistema com recurso ao algoritmo baseado na árvore de decisão C4.5/C5, dado que esta é mais rápida e ligeiramente mais eficaz na procura de padrões e na classificação em classes.

Ao nível da codificação, estas implementações seguem metodologias diferentes, enquanto que a árvore de decisão será implementada através da concepção de uma DLL⁸ baseada em código disponibilizado em open source por Quilan [26], a segunda seguiu uma implementação por classes integradas no código fonte do CE-e.

Assim, foi possível implementar e testar o sistema em busca de erros ou problemas de implementação, em tempo útil, antes do início do ano escolar no ensino básico, onde o sistema foi efectivamente colocado à prova.

Após a prova de conceito ter sido realizada, a DLL baseada na árvore de decisão de [26], continuará a ser desenvolvida. Este esforço justifica-se pelo facto da DLL ser independente do sistema e portátil, permitindo assim a sua reutilização noutros sistemas, para além de ser mais fácil a construção de novas versões do CE-e, com e sem interface inteligente, sem que esta situação origine a construção de duas versões,

⁸ *Dynamic-link library* (Biblioteca de ligação dinâmica) ou DLL, é a implementação feita pela *Microsoft* para o conceito de bibliotecas partilhadas nos sistemas operativos *Microsoft Windows* e OS/2.

```

148 public /*DataTable*/ void readFromLogFile()
149 {
150     DataColumn col;
151     DataTable dt = new DataTable();
152
153     //First column - Sequences
154     col = new DataColumn();
155     col.DataType = System.Type.GetType("System.String");
156     col.ColumnName = "Sequences";
157     dtFinal.Columns.Add(col);
158
159     //Second column - Label
160     col = new DataColumn();
161     col.DataType = System.Type.GetType("System.String");
162     col.ColumnName = "Label";
163     dtFinal.Columns.Add(col);
164
165     //Third column - States
166     col = new DataColumn();
167     col.DataType = System.Type.GetType("System.String");
168     col.ColumnName = "States";
169     dtFinal.Columns.Add(col);
170
171     connection.Open();
172
173     command.CommandText = "SELECT * FROM Actions";
174     reader = command.ExecuteReader();
175     dt.Load(reader);

```

Figura 3.8: Exemplo de código, em que se mostra o *query* e posterior inserção na *Data Table*

incompatíveis.

Com a introdução do motor de inferência no protótipo funcional do CE-e, o sistema teve de sofrer algumas alterações, não visíveis, que apenas se reflectiram no seu código fonte, para que os vários componentes da sua interface possam ser alterados dinamicamente por indicação do motor de inferência da interface.

A adaptação da estrutura interna do sistema, tal como já foi referido, teve de sofrer algumas alterações, assim, para além do registo de logs que já se encontrava em funcionamento há algum tempo com recurso a um SGBD relacional (SQLite 3), o passo seguinte foi adaptar o sistema para que no seu arranque conseguisse carregar os dados contidos na base de dados e posteriormente, os conseguisse utilizar na sugestão de acções. Nos próximos pontos descreveremos em pormenor estes passos.

3.4.1 Treino do Classificador do Motor de Inferência

Para que o motor de inferência pudesse efectivamente sugerir a acção mais provável que o utilizador efectuará no sistema, o motor teria de ser capaz de no arranque do sistema ler toda a informação contida na base de dados, ou seja, todos os *logs* de utilização.

Assim, no arranque do sistema o motor de inferência consulta a base de dados, através

```

private string intToBinary(Int64 Decimal)
{
    // Declare a few variables we're going to need
    Int64 BinaryHolder;
    char[] BinaryArray;
    string BinaryResult = "";

    while (Decimal > 0)
    {
        BinaryHolder = Decimal % 2;
        BinaryResult += BinaryHolder;
        Decimal = Decimal / 2;
    }

    // The algorithm gives us the binary number in reverse order (mirrored)
    // We store it in an array so that we can reverse it back to normal
    BinaryArray = BinaryResult.ToCharArray();
    Array.Reverse(BinaryArray);
    BinaryResult = new string(BinaryArray);

    return BinaryResult;
}

```

Figura 3.9: Exemplo de código, em que se mostra a conversão de inteiro para binário

de um normalíssimo *query* em SQL e em seguida trata o resultado deste *query*, passando todas as linhas para uma estrutura em memória para que possa dar início à próxima fase do treino do classificador. Nesta fase ainda não se pode dizer que o sistema se encontre já a treinar o classificador, pode-se sim, dizer que se encontra numa fase de pré-processamento dos dados, dado que após o *query*, as linhas do mesmo são inseridas numa *data table*, como se pode ver na figura 3.8.

O passo seguinte passa pela conversão dos *logs* para formato binário (Figura 3.9), dado que o classificador funciona com dados em formato binário. Após esta transformação, o classificador inicia então o processo de treino, lendo todas as ocorrências e respectivas classes, ou seja, todas as acções anteriormente efectuadas pelo utilizador no sistema. Nesta fase, o classificador efectua uma série de cálculos por forma a atribuir a cada registo, que é composto pelos dados de cada linha do ficheiro de *logs*, um valor probabilístico, ao qual a classe (opção escolhida pelo utilizador), fica associada. Estes cálculos são efectuados a partir dos valores binários e, é por este motivo que existe a necessidade de numa primeira fase pré-processar os *logs*, por forma a transformar, quer os valores numéricos, como os valores não numéricos em binário. Sem esta transformação não seria possível efectuar os cálculos por forma a achar o valor probabilístico associado a cada registo.

Este valor probabilístico, ao qual cada registo e respectiva classe ficam associados, serve como valor referência, ou seja, todos os restantes registos e classes cujo valor se enquadre na mesma gama de valores, ficará associado a uma determinada classe. Nesta fase o

```

public MainForm()
{
    InitializeComponent();
    timer1.Start();

    lc = new LogClass();

    t = new Thread(new ThreadStart(lc.readFromLogFile));
    t.Start();

    //lc.readFromLogFile();
    staticLC = lc;

    this.progressBar1.Value = 0;
}

```

Figura 3.10: Exemplo de código, em que se mostra o arranque da *thread*

classificador é treinado, criando um conjunto de padrões criados a partir do conjunto de treino, que no caso do Caderno Escolar Electrónico Adaptativo, são os valores que foram registados no ficheiro de *logs* do sistema. Desta forma todos os registos, cujo valor se enquadre num determinado intervalo probabilístico, será associado a uma determinada classe, a classe essa que corresponde à maioria das acções efectuadas pelo utilizador que correspondem ao padrão em causa.

Assim, quando o sistema for chamado a efectuar sugestões, enquadrará o contexto no momento, através do cálculo do mesmo valor probabilístico, no que mais se aproxime de entre os intervalos padrão, encontrados durante a fase de treino, conseguindo assim sugerir a acção que mais se adequa ao contexto em que o utilizador se enquadra.

Todo este processamento é efectuado durante o arranque do sistema, o que poderá tornar o arranque mais lento do que seria espectável, por este facto foi criada uma barra de progressão que informa o utilizador do progresso da operação. Toda esta operação é efectuada numa *thread* à parte (Figura 3.10).

3.4.2 Utilização das Sugestões do Motor de Inferência

Tal como foi referido no final do ponto anterior, no momento em que o sistema for chamado a realizar sugestões, comparará o valor probabilístico encontrado para o contexto do momento com os intervalos probabilísticos encontrados durante a fase de treino. A sugestão de acções é efectuada em todos os ecrãs do sistema, nos quais exista mais do que um caminho possível a ser seguido tal como a figura 3.11 mostra.

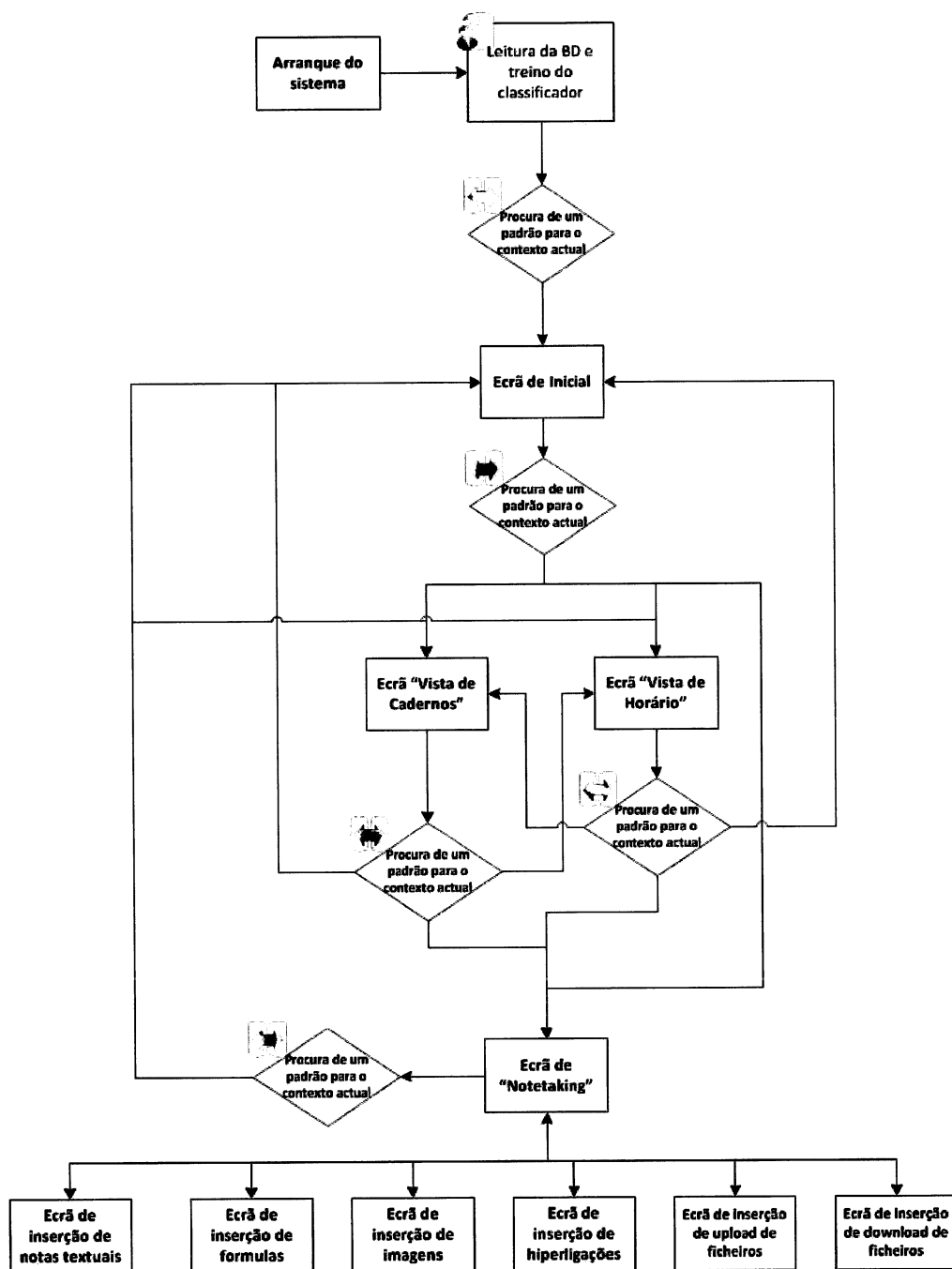


Figura 3.11: Esquema de caminhos possíveis no sistema

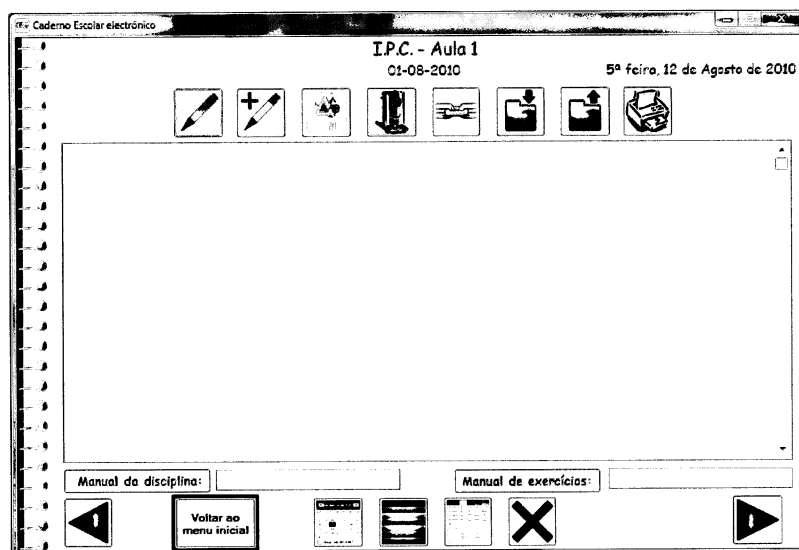


Figura 3.12: CE-e a efectuar uma sugestão

Esta sugestão tem obrigatoriamente de entrar em linha de conta com o contexto do momento, assim, no momento em que um ecrã é invocado, existe também uma chamada ao sistema para determinar o dia da semana e a hora, para depois em conjunto com o ecrã (invocado) e a disciplina activa (se aplicável), ser realizada outra chamada ao sistema, desta vez ao motor de inferência, para que este determine qual a acção mais provável a ser realizada no sistema.

A sugestão efectuada pelo motor de inferência é realçada através de uma mudança de cor ou através da colocação de uma moldura no botão que acciona a opção sugerida. Mas como o objectivo do sistema é aliviar ao máximo o utilizador, este realce é acompanhado da atribuição do focus (o objecto em causa passa para primeiro plano), o que permite ao utilizador despoletar a acção com um simples toque nas teclas "enter" ou "return".

Mas, o sistema tinha também como objectivo associar um conjunto de regras fixas, às sugestões efectuadas pelo motor de inferência. Estas regras encontram-se associadas ao horário escolar do utilizador em conjunto com o dia da semana e hora. Assim, sempre que se aproxime a hora de início de uma aula, ou sempre que se sobreponham uma regra e uma sugestão do motor de inferência, em caso de divergência, a regra tem uma maior prioridade, sendo a sugestão desta a efectuada, em detrimento da sugestão efectuada pelo motor de inferência. Na figura 3.12 podemos ver o Caderno Escolar Electrónico Adaptativo a efectuar uma sugestão.

No entanto, tal como referido anteriormente, a interface adaptativa do CE-e é do tipo sugestivo, ou seja, todas as sugestões que o motor de inferência propõe não são automaticamente accionadas, ou sequer obrigam o utilizador a segui-la. Desta forma, o utilizador é livre de escolher outra opção caso a sugestão não vá de encontro ao que tenha em mente.

De uma forma genérica, o utilizador tem duas hipóteses de interagir com o sistema:

- (1) Verifica a sugestão do motor de inferência e aceita-a, com um clique do rato ou premindo as teclas "enter" ou "return";
- (2) Verifica a sugestão do motor de inferência e não a aceita, escolhendo de seguida a opção desejada.

À parte das sugestões do motor de inferência, os *logs* de utilização continuam a ser registados pelo sistema, sempre com a opção efectivamente escolhida pelo utilizador, quer esta corresponda ou não à sugestão efectuada pelo sistema. Assim, o motor de inferência vai aprendendo à medida que o número de utilizações do sistema aumenta. Consequência directa deste continuo registo dos *logs* é a melhoria do poder de sugestão do motor de inferência, porque a cada nova sessão que um utilizador inicie, o classificador vai ser treinado com todo o histórico de utilizações.

Capítulo 4

TESTES DA APLICAÇÃO

Após a codificação do sistema, o nosso principal objectivo era efectuar os testes que efectivamente demonstrassem o real valor da interface "inteligente" do Caderno Escolar Electrónico e se a mesma efectivamente conseguia atingir os pressupostos que tenham sido idealizados para esta.

4.1 Os Utilizadores

Os testes para aferir a qualidade da interface "inteligente", para terem validade teriam de ser efectuados por utilizadores, numa situação a mais próxima possível da realidade.

Assim, resolvemos utilizar um utilizador em situação simulada de sala de aula e um outro utilizador em contexto real de aula. Relativamente ao primeiro utilizador, o mesmo realizou as operações no Caderno Escolar Electrónico, de uma forma simulada, no entanto seguiu uma estratégia, que simulava completamente um contexto real escolar, dado que o horário escolar do sistema foi configurado com base num horário existente, tendo periodicamente sido efectuadas operações no sistema.

Ao contrário do segundo utilizador, este não possuía qualquer tipo de deficiência, conseguindo efectuar de uma forma totalmente autónoma todas as tarefas.

Também ao contrário do segundo utilizador, este utilizador é um adulto com conhecimento de informática na óptica do utilizador, com alguma destreza no acesso à internet e na utilização de processadores de texto e folhas de cálculo.

O segundo utilizador realizou operações no Caderno Escolar Electrónico em contexto real de sala de aula, com supervisão parcial por parte do professor titular da sala,

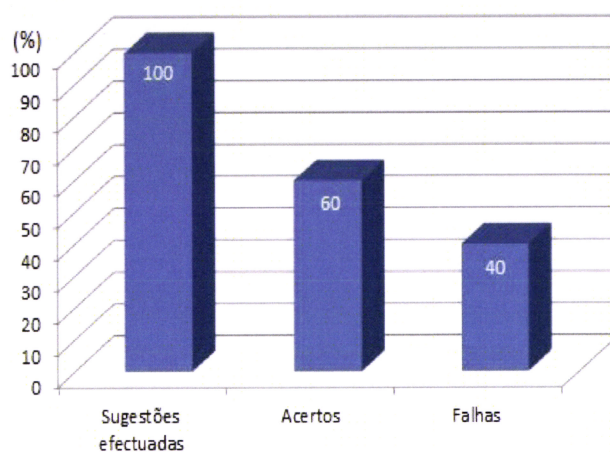


Figura 4.1: Gráfico com os resultados globais do utilizador 1

servindo esta supervisão para garantir a real utilização do sistema e qual o nível de acertos conseguido pelo sistema.

Este utilizador possui alguns problemas físicos, provocados por paralisia cerebral, situando-se assim dentro do principal grupo alvo do projecto.

O professor titular, para além de garantir a realização das tarefas e de registar o número de acertos, no que toca às sugestões efectuadas pelo sistema, registou também as sugestões incorrectas e qual a real intenção do utilizador.

4.2 As técnicas

Tal como foi referido já anteriormente, os testes efectuados, foram repartidos entre uma sala de aula e um contexto escolar simulado. Desta forma tentamos garantir uma grande abrangência nos resultados dos testes.

É sabido, por experiência própria, que os testes de utilização em contexto real de sala de aula, com miúdos do 1º ciclo do ensino básico, nem sempre reflectem a totalidade dos trabalhos efectuados na sala de aula, ou seja, como é o professor que coordena as actividades lectivas, normalmente indica tarefas, a serem efectuadas no Caderno Escolar Electrónico, apenas de uma determinada área, "esquecendo" outras, o que é compreensível, dado que os programas são exigentes e as turmas são grandes não permitindo grandes veleidades ou as perdas de tempos que estes testes implicam, devido

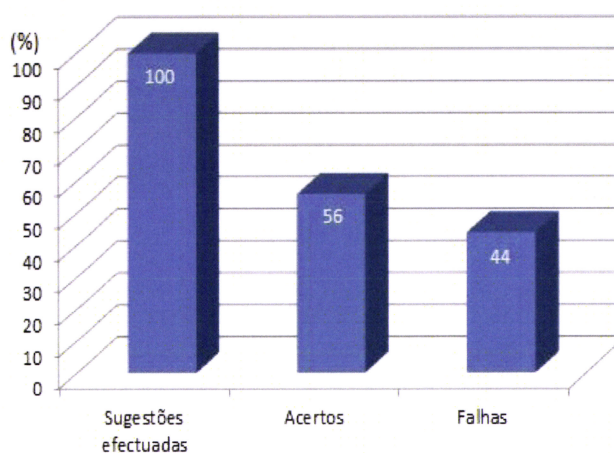


Figura 4.2: Gráfico com os resultados parciais do utilizador 1

ao registos das sugestões e acções realizadas, entre outras.

Assim, o utilizador simulado tentou efectuar um leque de operações, quão abrangente quanto possível. Desta forma e recorrendo a um ambiente totalmente simulado, mas o mais próximo possível da realidade, dado que o horário escolar configurado no sistema era real e as acções efectuadas cobriam todo o período de actividades lectivas e alguns períodos que normalmente são passados em casa pelos alunos, mas que serviram para testar uma eventual realização de trabalhos de casa, por parte do utilizador.

Desta forma, podemos dizer que a uma maior previsibilidade de tarefas efectuadas, por parte do primeiro utilizador, tentamos juntar uma maior aleatoriedade nas acções realizadas, por parte do segundo utilizador, ainda que o contexto fosse simulado.

4.3 Os Resultados

Nesta secção vamos apresentar os resultados dos testes, individualmente para cada utilizador e de seguida uma comparação dos resultados médios dos dois utilizadores. Para facilitar a leitura e percepção dos resultados, em todos os gráficos são apresentados valores médios para cada 100 sugestões efectuadas.

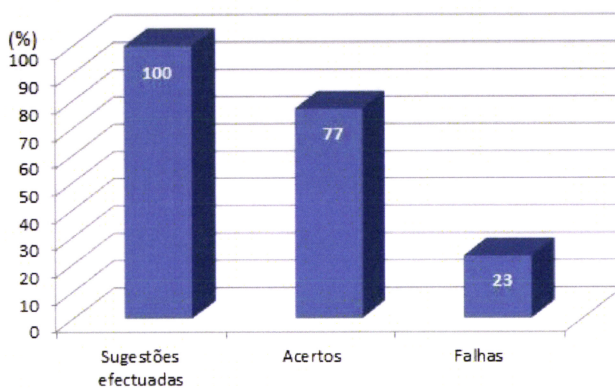


Figura 4.3: Gráfico com os resultados globais do utilizador 2

4.3.1 Resultados do Utilizador 1

No gráfico da Figura 4.1, apresentamos os resultados globais para o utilizador 1 (utilizador em situação simulada), ou seja todas as sugestões efectuadas pelo sistema, sendo estas provenientes do motor de inferência ou provenientes das regras fixas.

Este gráfico mostra-nos que para cada 100 sugestões efectuadas pelo sistema, 60 corresponderam à real intenção do utilizador, o que resulta numa percentagem de de acertos de 60%, uma percentagem que vem confirmar os resultados obtidos com a ferramenta de aprendizagem automática na altura em que se analisaram os *logs*. Relembro que este utilizador tem uma maior abrangencia de opções escolhidas.

No entanto para testar a real eficácia do sistema, retiramos dos dados em análise as sugestões (certas) efectuadas por intermédio das regras. O gráfico da Figura 4.2 mostra que a percentagem de acertos desce ligeiramente para cerca de 56%, o que demonstra que mesmo sem as sugestões efectuadas pelas regras, a percentagem de acertos continua a ser bastante aceitável.

4.3.2 Resultados do Utilizador 2

À semelhança do que fizemos para o utilizador 1, vamos de seguida apresentar os mesmos resultados para o utilizador 2. Assim, apresentamos os resultados globais para o utilizador 2 (utilizador em contexto real).

Analisando o gráfico da Figura 4.3, temos que para cada 100 sugestões efectuadas o motor de inferência do sistema conseguiu uma percentagem de acertos na ordem dos

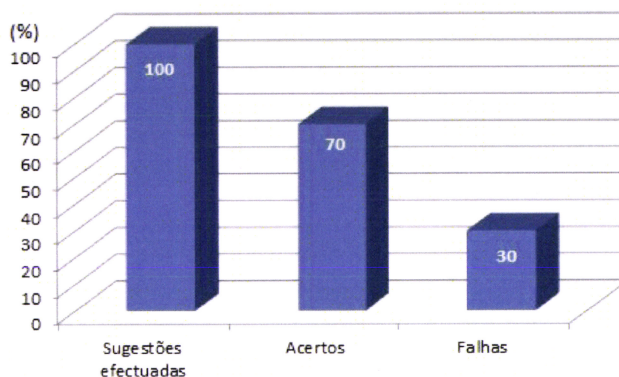


Figura 4.4: Gráfico com os resultados parciais do utilizador 2

77%, ou seja a percentagem de acertos conseguida nos testes com este utilizador superou a alcançada nos testes com o utilizador 1. Este resultado deve-se principalmente a um facto já referido, ou seja, o menor número de tarefas executadas por este no sistema, o que resulta num menor número de variáveis a analisar pelo sistema, logo existindo uma maior probabilidade de acerto na acção pretendida pelo utilizador.

Também à semelhança do que fizemos para o utilizador 1, quisemos saber como se comportaria o motor de inferência, se não tivesse disponível as regras, que tal como já foi referido, resultam em sugestões certas. Assim e, um pouco na sequência dos resultados do utilizador 1, quando retiramos as regras a percentagem de acertos também desceu ligeiramente, ou seja, dos 77% de acertos obtidos anteriormente, obtivemos 70% (Figura 4.4), um resultado também bastante aceitável. Esta descida deve-se ao motivo já relatado para o utilizador 1, mas também se deve ao menor número de sugestões efectuadas pelo motor de inferência, cerca de 40% menos.

Relembro que este utilizador, muito jovem, é possuidor de paralisia cerebral e, todas as tarefas que efectuou foram sob supervisão do docente titular da turma.

4.3.3 Resultados Globais

De uma forma global, podemos então dizer que os resultados obtidos são satisfatórios, dado que como se pode ver no gráfico da Figura 4.5, a média de acertos entre os dois utilizadores, utilizando as regras atinge um valor médio de 68,5 acertos em cada 100 sugestões efectuadas, enquanto se retirarmos as regras atingimos um valor médio de acertos de 63. De uma forma praticamente directa retiramos um valor médio

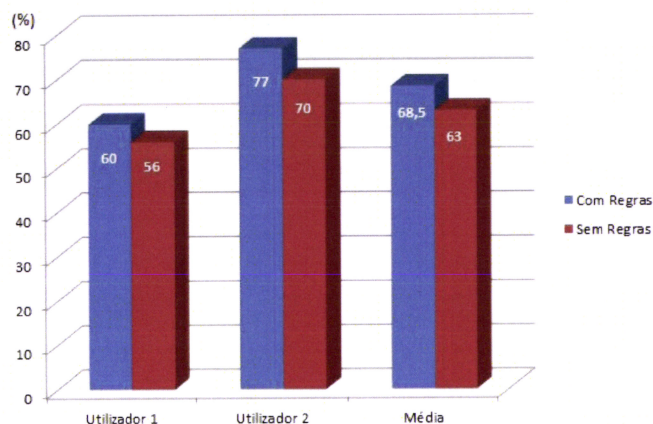


Figura 4.5: Gráfico com os resultados globais

geral para os acertos a rondar os 65%.

Este valor é encorajador, no entanto pode ainda ser algo enganador e não reflectir a realidade, senão vejamos:

- O período de testes é relativamente curto; Será necessário um período mais alargado de testes para podermos aferir com um relativo grau de certeza, se estes valores reflectem o real número de acertos, ou se por outro lado estes resultados se encontram abaixo, ou acima, do que o motor de inferência pode produzir;
- Os resultados do utilizador 2, como já foi referido, foi supervisionado e apenas algumas tarefas foram efectuadas, particularmente as tarefas que envolvem escrita, mas as do utilizador 1 também podem sofrer algum desajuste dado que as tarefas efectuadas podem estar afectadas por vícios de utilização, isto porque nós somos humanos e somos "animais" de hábitos e executamos tendencialmente as mesmas coisas, da mesma forma.

Apesar de tudo pensamos que estes resultados, preliminares, são encorajadores e poderão servir como base de trabalho para testes com mais utilizadores, que possam abranger um maior número de alunos, espalhados por várias escolas e por vários níveis de ensino.

Não entanto não é fácil obter a colaboração e apoio dos docentes do ensino básico, nomeadamente nos ciclos mais adiantados. Até ao momento tivemos a preciosa colaboração de dois docentes do 1º ciclo do ensino básico que nos ajudaram e colaboraram incondicionalmente, muitas vezes em detrimento do bom funcionamento

das suas próprias aulas. Mas os programas de ensino, vão sendo cada vez mais exigentes à medida que se avança nos níveis e, não poderemos poder contar com o mesmo género de colaboração, isto também porque muitas vezes os docentes não estão devidamente alertados para este tipo de problemas, dado que muitas vezes os alunos não são inseridos no ensino regular, sendo afastados para turmas e escolas especiais.

Capítulo 5

CONCLUSÕES E TRABALHO FUTURO

Nesta dissertação propusemos e apresentamos uma interface adaptativa para uma ferramenta que pretende ser uma alternativa aos tradicionais cadernos escolares, para estudantes com necessidades especiais, em particular aqueles que apenas possuem problemas físicos, que lhes restringe a mobilidade e a habilidade para manipular os tradicionais mecanismos de input.

Actualmente existe um protótipo do CE-e com a interface inteligente, totalmente funcional e testado, capaz de efectuar sugestões ao utilizador, aliviando-o assim na execução de tarefas rotineiras.

A interface "inteligente" do CE-e faz sugestões ao utilizador com recurso à análise de uma base de conhecimento, que contem informação sobre acções anteriormente efectuadas no sistema. A análise é efectuada com base em técnicas de aprendizagem automática, através de algoritmos baseados em Modelos "Escondidos" de Markov.

Com vista à construção de uma base de conhecimento que conseguisse responder às questões colocadas durante a fase de análise e decisão, o mecanismo de registo de logs passou por uma base de dados relacional, com recurso à tecnologia SQLite, dado que a interrogação de bases de dados em SQL é uma prática bastante eficiente. De igual forma o registo de logs num sistema deste tipo é bastante simples e eficiente, para um nível de utilização realista, não apresentando qualquer tipo de sobrecarga, quer para o Caderno Electrónico, como para o sistema operativo.

Como se pretende que o sistema vá aprendendo ao longo do tempo, o ficheiro de *logs* poderá crescer indefinidamente tornando a sua análise pelo motor de inferência da interface do CE-e, muito lenta. Para ultrapassar este potencial problema, o ficheiro *logs* poderá ver o seu crescimento limitado. Esta limitação no crescimento, poderá

ser regulada com recurso ao tempo, sendo os *logs* com datas posteriores a trinta dias (efectivamente registados no mesmo) eliminados. Se verificarmos que este período é demasiado curto, ou longo demais, esta regra será reajustada.

No entanto, a hipótese de existir uma limitação temporal dos *logs*, pode ainda sofrer alterações dado que estamos a analisar qual a opção que trará maiores probabilidades de acertos.

Se por um lado tivermos um sistema que apenas conhece os últimos trinta dias de utilização, este terá um conhecimento temporalmente limitado, mas esse conhecimento poderá incidir sobre as opções que o utilizador tem utilizado com maior frequência, neste período limitado de tempo, podendo assim conseguir uma maior quantidade de acertos, dado que as opções que um utilizador invoca hoje com mais frequência, poderão ser totalmente opostas às que invocará daqui a um, dois ou mais meses.

Por outro lado e, partindo da hipótese que os *logs* podem crescer indefinidamente, o sistema terá uma visão global de todas as acções e escolhas efectuadas pelo utilizador, o que poderá trazer maiores probabilidades de acerto. Esta decisão só poderá ser tomada, após um período de utilização e testes de duração não inferior a seis meses, que demonstrará qual o melhor tipo de base de conhecimento, "com esquecimento" ou "sem esquecimento".

Uma eventual solução para o problema, do crescimento da base de dados, motivado pela recolha contínua de logs, poderá passar pelo seu processamento em offline, o que levará à construção de um índice de acções, à semelhança do que se passa num sistema de recuperação de informação em bases de texto [23], ou seja, poderia ser criado um sistema semelhante a um *Webservice*, que sem causar intrusão efectuará uma ligação remota a um servidor *Web*, para onde enviaria o ficheiro da base de dados, o qual seria processado e posteriormente enviado para o computador do utilizador, evitando assim tempos de espera por parte do utilizador, reduzindo também a carga de processamento na máquina local do utilizador.

No entanto esta solução não é única e, num mundo cada vez mais virado e ligado à Internet, a hipótese de criar um Caderno Escolar Electrónico baseado num sistema Web, é algo que não pode ser descartado, podendo inclusive os actuais problemas com que nos deparamos, deixar de o ser com um sistema que "corra" a partir de um servidor

remoto. Esta mudança de paradigma tem vantagens, no entanto não nos podemos esquecer do facto que o utilizador teria de estar obrigatoriamente ligado à Internet, algo que efectivamente não poderá ser considerado como um problema grave, dado que o parque escolar encontra-se a ser dotado que equipamentos tecnológicos e ligações de banda larga à Internet.

Um Caderno Escolar Electrónico "Inteligente Web" aliviaria a necessidade do utilizador transportar consigo um computador portátil "a sério", podendo a interacção com o sistema efectuar-se a partir, por exemplo, de um netbook.

Outro objectivo, inerente a esta proposta, era a criação de um conjunto de regras fixas, que em caso de não correspondência entre a mesma e o resultado obtido pelo motor de inferência, estas sobreporiam-se à sugestão efectuada pelo motor de inferência. Como exemplo deste tipo de regras temos o horário escolar que poderá fornecer informações sobre qual a tarefa mais provável a executar no sistema, de acordo com a componente tempo fornecida pelo sistema operativo. Estas regras foram também totalmente implementadas, com base no horário escolar, podendo assim o sistema efectuar, validar ou corrigir as sugestões efetuadas.

O mecanismo de sugestão coloca em realce a opção mais provável. Este realce está aliado a uma tecla de atalho, o que evita um movimento de rato para selecção ou validação da opção sugerida pelo motor de inferência. Com esta medida, aliviou-se ainda mais o esforço do utilizador, deixando mais tempo para que o mesmo realize tarefas que acrescentam valor ao trabalho do mesmo.

Relativamente aos testes efectuados, voltamos a realçar a colaboração que nos foi prestada por alguns docentes do 1º ciclo do ensino básico. De igual forma temos fé que o futuro nos permita alargar os testes a mais escolas e mais níveis de ensino, só assim poderemos continuar a desenvolver o Caderno Escolar Electrónico e aferir a sua real utilidade.

Por último gostaríamos de voltar a frisar, que este trabalho foi presente em duas conferências, sendo uma destas mais voltada para a inteligência artificial (Inforum'2010) [2] e a outra especificamente direccionada para a problemática das interfaces e da interacção dos humanos com os computadores (Interacção'2010) [1]. Em ambas os resultados e a aceitação do trabalho foram bons, tendo sido referido a novidade e

interesse do trabalho, dado que mistura duas áreas do conhecimento das ciências da computação, as interfaces e todas as questões inerentes à interacção, com a aprendizagem automática.

O CE-e pretende ser uma alternativa aos tradicionais cadernos escolares. Com a introdução de uma interface inteligente que se sugere a opção mais adequada ao contexto do momento, os estudantes com dificuldades físico-motoras, terão ao seu dispor uma ferramenta que os ajudará ainda mais na organização de todas as tarefas escolares que envolvam escrita e na organização de todos os materiais escolares. A criação desta interface inteligente para o CE-e, que consegue prever qual a acção mais provável para um determinado contexto, veio reforçar o principal objectivo do CE-e: Ajudar e aliviar os estudantes com necessidades especiais de tarefas rotineiras, que não acrescentam mais-valias ao trabalho que têm de efectuar, de forma a ajudar na sua integração no ensino regular.

Bibliografia

- [1] Luis Alexandre and Salvador Abreu. Caderno escolar electrónico adaptativo - resultados preliminares. Interacção'2010 - 4ª Conferência Nacional Interacção Humano-computador, 2010.
- [2] Luís Alexandre and Salvador Abreu. Development of an adaptive interface for the electronic school notebook. INForum'2010 - Simpósio de Informática, 1:733–744, 2010.
- [3] Luís Alexandre, Luís Garcia, and Luís Bruno. Development of an electronic scholar notebook for students with special needs. In DSAI2007, Vila Real, Portugal, 2007.
- [4] Ethem Alpaydin. Introduction to Machine Learning. MIT Press, 2004.
- [5] Leon Barnard, Ji Soo Yi, Julie A. Jacko, and Andrew Sears. Capturing the effects of context on human performance in mobile computing systems. 2006.
- [6] Mark Lawrence Blum. Real-time context recognition. Master's thesis, Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology Zurich (ETH), 2005.
- [7] Joseph R. Boyle and Mary Weishaar. The effects of strategic notetaking on the recall and comprehension of lecture information for high school students with learning disabilities. Learning Disabilities Research & Practice, 16(3), pages 133–141, 2001.
- [8] Nicholas A. Bradley and Mark D. Dunlop. Toward a multidisciplinary model of context to support context-aware computing. Human Computer Interaction, 20:403 – 446, 2005.
- [9] Nills Dahlbäck, Arne Jönsson, and Lars Ahrenberg. Wizard of oz studies - why and how. Department of Computer and Information Science, Sweden, 2003.
- [10] Brian D. Davison and Haym Hirsh. Predicting sequences of user actions. In AAAI/ICML 1998 Workshop on Predicting the Future, 1998.
- [11] Anind K. Dey. Understanding and using context. Personal Ubiquitous Computing, 5:4 – 7, 2001.
- [12] Alan Dix, Janet Finlay, Gregory Abowd, and Russell Beale. Human Computer Interaction, 3rd Edition. Prentice Hall, 2004.
- [13] Enrique Frias-Martinez, Sherry Y. Chen, and Xiaohui Liu. Survey of data mining approaches to user modeling for adaptive hypermedia. IEEE Transactions on Systems, Man and Cybernetics - Part c: Applications and Reviews, 36, NO. 6:734 – 749, 2006.

- [14] Luís Garcia. Conceção, implementação e teste de um sistema de apoio à comunicação aumentativa e alternativa para o português europeu. Master's thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico, 2003.
- [15] S. R. Garner. The waikato environment for knowledge analysis. In New Zealand Computer Science Research Students Conference, pages 57 – 64, Hamilton, New Zealand, 1995. University of Waikato.
- [16] Marcos André Gonçalves, Ganesh Panchanathan, Unnikrishnan Ravindranathan, Aaron Krowne Edward A. Fox Filip Jagodzinski, and Lillian Cassel. The xml log standard for digital libraries: analysis, evolution, and deployment. Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries, 1:312 – 314, 2003.
- [17] Takeo Igarashi and John F. Hughes. A suggestive interface for 3d drawing. Computer Science Department, Brown University.
- [18] Jason Beres Pierre Boutquin Jeff Ferguson, Brian Patterson and Meeta Gupta. C# Bible. Wiley Publishing, Inc., 2002.
- [19] Pat Langley. Machine learning for adaptive user interfaces, volume 1303/1997 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006.
- [20] Yue-Shi Lee and Show-Jane Yen. Neural-based approaches for improving the accuracy of decision trees. Data Warehousing and Knowledge Discovery, 2454/2002:125–140, 2002.
- [21] Gregory W. Lesh, Bryan J. Moulton, Gerard Rinkus, and D. Jeffery Higginbotham. A universal logging format for augmentative communication. 2000.
- [22] Pattie Maes. Agents that reduce work and information overload. Communications of the ACM 37(7), 1994.
- [23] C.D. Manning, P. Raghavan, and H. Schütze. An Introduction to Information Retrieval. Cambridge University Press, 2008.
- [24] Tom M. Mitchell. Machine Learning. McGraw-Hill Science/Engineering/Math, 1997.
- [25] Anthony F. Norcio and Jaki Stanley. Adaptive human- computer interfaces: A literature survey and perspective. IEEE Transactions on Systems, Man and Cybernetics, 19, No.2:399 – 408, 1989.
- [26] Ross Quilan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
- [27] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 2:257 – 286, 1989.
- [28] Nelson Rodrigues. Desenvolvimento de mecanismo de interação predictiva para aumentar o desempenho de tarefas. Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Beja, 2007. Projecto de Licenciatura.
- [29] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94), 1:89 – 101, 1994.

- [30] B.N. Schilit and M.M. Theimer. Disseminating active map information to mobile hosts. IEEE Network, pages 22 – 32, 1994.
- [31] UNESCO. The salamanca statement and framework for action on special needs education, 1994.
- [32] Roel Vertegaal. Designing attentive interfaces. ETRA'02 New Odeans Louisiana USA, 1:23 – 30, 2002.

