

# SISTEMAS DE REGRAS APLICADOS A INQUÉRITOS “ONLINE”

Paulo Urbano

*Faculdade de Ciências da Universidade de Lisboa  
C6, Piso 3, Campo Grande 1749-16, Lisboa, Portugal  
E-pub@di.fc.ul.pt*

David Rodrigues

*Instituto Superior das Ciências do Trabalho e da Empresa  
Av. das Forças Armadas 1649-026 Lisboa, Portugal  
[m4467@iscte.pt](mailto:m4467@iscte.pt)*

## RESUMO

Neste artigo descrevemos uma aplicação dos Sistemas Baseados em Regras a um programa de inquérito “online” inteligente, o De.:SID. Incorporámos no programa três bases de conhecimento baseadas em regras que controlam três componentes vitais num inquérito: (1) as dependências entre as várias perguntas, ou seja a própria estrutura e a lógica de ramificação do inquérito; (2) a detecção de inconsistências entre as várias respostas e (3) a decisão quanto à selecção da próxima pergunta a ser colocada. Estes componentes permitem que não se imponha uma ordem pré-determinada na colocação das perguntas, dotando o sistema de uma flexibilidade e capacidade de adaptação às respostas do utilizador, permitem uma fácil e livre navegabilidade ao longo das várias perguntas e a possibilidade da revisão das respostas sem perder a coerência do inquérito. Para além disso, beneficiamos das virtudes geralmente associadas às regras: a separação e consequente exposição da lógica associada ao inquérito e a facilidade de mudar essa mesma lógica sem alterar os outros elementos do programa.

## PALAVRAS-CHAVE

Sistemas baseados em Regras, Inquérito “online” inteligente, Bases de conhecimento.

## 1. INTRODUÇÃO

No âmbito do projecto “Design como recurso estratégico empresarial: estudo dos impactos do Design”, acolhido na Faculdade de Arquitectura da Universidade Técnica de Lisboa financiado pela FCT e com os parceiros Centro Português de Design e Associação Portuguesa de Designers, a equipa multidisciplinar tinha como um dos objectivos a construção de um inquérito “online” que clarificasse o papel do Design na Indústria Transformadora Nacional (ITN). À partida poderíamos seleccionar uma entre duas estratégias para a construção do inquérito “online”; ou utilizar uma das ferramentas existentes no mercado (por exemplo, SurveyMonkey, Ideascale, Questionpro, Questionform, etc..) que permitem construir e gerir inquéritos “online”, ou construir um sistema inteiramente de raiz. Depois de analisadas as ferramentas decidimos avançar com um inquérito feito de raiz. O motivo porque o fizemos prende-se com o facto de as ferramentas existentes terem algumas limitações, sendo a principal delas, pressuporem uma ordem de colocação de perguntas que é definida à priori e que não é muito sofisticada. Em geral, nestes sistemas, que usam técnicas muito semelhantes, a lógica de ramificação que gere a cadeia de dependências entre as perguntas e as respostas está muito ligada à ordem de colocação das perguntas. Podemos exemplificar com uma regra de ramificação típica: se a pergunta 17 teve a resposta “sim” então salta para a 45 senão continua na 18. A mistura entre a escolha da próxima pergunta a ser colocada e a estrutura também lida dificilmente com modificações. Se quisermos apagar uma pergunta, mudá-la de posição, alterar o seu identificador ou modificar a lógica de ramificação, temos de programar à mão as alterações a outras secções do inquérito. Esta

debilidade é relevante mas, mais importante ainda, estes sistemas comprometem-nos com uma sequência determinada de perguntas retirando-lhe a flexibilidade que achamos necessária. Regressando ao nosso exemplo, a pergunta 18 não fará sentido para quem respondeu *sim* à 17 mas porquê saltar para a 45 e não para outra qualquer que faça sentido e que seja de importância equivalente? Existem perguntas que só fazem sentido no contexto de certas respostas mas podemos não nos querermos comprometer com uma ordem rígida, definida totalmente a priori e para todos os utilizadores. Determinadas respostas ou combinações de respostas podem inibir ou permitir um conjunto de outras perguntas mas não devem impor uma narrativa monolítica do questionário. Achamos que a ordem de colocação das perguntas deve ser claramente separada da lógica de ramificação para benefício da usabilidade e da capacidade de adaptação à mudança por parte dos questionários.

O facto de desejarmos definir e declarar a lógica de ramificação das perguntas não nos deve impedir de estabelecer critérios mais sofisticados para a selecção das perguntas a apresentar. Se quisermos, por exemplo, que a próxima pergunta a ser colocada dependa da história particular das respostas a um inquérito, isso não é possível com estas ferramentas. Não permitem também que possamos preferir perguntas que foram apresentadas um número de vezes menor do que outras que o foram mais vezes—note-se que estamos a pressupor a possibilidade de adiar uma resposta e de apresentar a mesma pergunta ainda não respondida várias vezes. O facto de um utilizador do inquérito ter acabado de fazer login pode ser uma fonte de informação para a selecção da primeira pergunta a apresentar —estamos a pressupor que o utilizador possa responder ao inquérito “online” ao longo de várias sessões. Em resumo, pretendemos que a ordem de colocação de perguntas (a narrativa do inquérito) possa adaptar-se em tempo real ao utilizador e que não seja definida *à priori* nos bastidores.

Outra das funcionalidades que desejávamos era a capacidade de definir as combinações de respostas incoerentes ou inconsistentes e poder avisar o utilizador da sua presença, sem o impedir de continuar a responder ao inquérito. Esta funcionalidade pode também ser utilizada “offline” para tratamento posterior dos dados ou as inconsistências detectadas podem ser o motivo para a selecção de novas perguntas. Para além disso, achámos que seria fundamental que o utilizador pudesse navegar livremente ao longo do inquérito, fornecendo-lhe a capacidade de poder saltar para qualquer pergunta que já foi apresentada, e que pudesse preencher a resposta caso estivesse por responder. Mais, teria de ter a liberdade de saltar directamente para uma pergunta já respondida e de a rever mantendo a coerência do inquérito. Imaginemos que um determinado utilizador, por exemplo, deu uma resposta r16a à pergunta 16 que inibe a pergunta 19, mas se mudar a sua resposta para r16b e esta activar a pergunta 19 então a pergunta 19 passa a fazer parte do relatório válido.

Chegámos à conclusão também que seria importante que o conhecimento sobre a ramificação lógica do inquérito, as inconsistências e os critérios de escolha da ordem de apresentação das perguntas estivessem em módulos completamente separados e independentes da restante parte do programa. Por um lado teríamos a transparência do conhecimento utilizado no inquérito, a possibilidade de o afinar e melhorar, sem pôr em cheque os restantes componentes. Sendo assim, as bases de conhecimento baseadas em regras [Jackson 1997] pareceram-nos serem a ferramenta mais natural para representar a estrutura, as inconsistências e os critérios de selecção das perguntas. A utilização de um motor inferencial para regular o funcionamento do inquérito através da definição de regras e não de árvores de decisão (a estrutura de dados típica das ferramentas de inquéritos) permite uma adaptabilidade do inquérito aos utilizador, permite focar na sequência de respostas, ou seja no perfil que se vai construindo do utilizador e não numa ideia pré-concebida do que o utilizador deveria fazer. Para além disso, um sistema baseado em regras, devido ao seu carácter modular e por ser um módulo separado, permite que os 3 componentes possam ser fácil e rapidamente modificados, alterando simplesmente algumas regras.

Os sistemas baseados em regras têm sido extensamente utilizados em aplicações de autorização de empréstimos bancários e de uso de emissão de cartões de crédito, detecção de fraudes [Winston and Prendergast 1984, Edwards and Connel 1989, Forsyth 1984, Harmon 1985, Waterman 1986]. “Os sistemas baseados em regras representam o conhecimento em termos de múltiplas regras que especificam o que pode ser concluído em diferentes situações. Um sistema baseado em regras consiste num conjunto de factos e de regras SE-ENTÃO, num interpretador que controla qual a regra que é invocada face aos factos existentes numa memória de trabalho” [Giarratano and Riley 2004]. Existem duas maneiras de interpretar as regras, em encadeamento para a frente e em encadeamento para trás. Um motor de inferência é polarizado pelos dados, sendo inicializado com um conjunto de factos tenta derivar novos factos aplicando as regras. O motor de

inferência com encadeamento para trás é orientado pelos objectivos, i.e., tenta provar certas hipóteses e utilizar as regras para o provar [Debenham 1989]. Os sistemas baseados em regras possuem algumas qualidades: a modularidade, que permite um desenvolvimento incremental, a transparência e as facilidades de explicação do raciocínio por detrás das decisões através do simples traço do encadeamento das regras, finalmente o facto de poderem representar e modelizar o conhecimento humano [Giarratano and Riley 2004]. Recentemente, temos vindo a assistir a um grande entusiasmo pelos sistemas de regras com as novas áreas de investigação e de aplicação de Gestão de Regras de Negócio [Ross 1997, Halle 2001, Graham 2006].

Do ponto de vista de quem concebe o inquérito, a arquitectura do D.:SID fornece 3 bases de conhecimento. Na primeira, é definida a estrutura e a lógica de ramificação do questionário; na segunda declara-se que combinação de respostas provoca situações inconsistentes e na terceira, que pensamos ser o aspecto mais inovador do nosso trabalho, estão explicitados os critérios que definem que pergunta é que vai ser seleccionada para ser colocada ao utilizador. Temos no fundo, a possibilidade, não só de definir esses critérios, tornando-os transparentes, como também de os rever e melhorar.

As regras foram definidas em CLIPS que é um software desenvolvido pela NASA que serve para implementar sistemas periciais. O acesso a estes motor é feito através de uma extensão a PHP denominada PHLIPS que funciona como interface entre a aplicação Web e o motor de regras.

Na próxima secção, após descrever o estado da arte no que diz respeito às formas de lidar com a lógica de ramificação dos inquéritos, apresentamos as nossas próprias contribuições; na secção 3 descrevemos como o inquérito D.:SID permite uma livre navegabilidade ao longo das perguntas, seleccionando directamente as perguntas desejadas e possibilitando a revisão das respostas mantendo sempre sua integridade; na secção 4 apresentamos a base de conhecimento associada à detecção das inconsistências; na secção 5, falaremos da base de conhecimento onde são declaradas as regras que decidem a ordem das perguntas; na secção 6 vamos mostrar as diferentes partes que compõem a arquitectura do programa do inquérito; antes de fechar, na secção 7, focaremos na interacção entre o sistema de regras e o servidor Web; finalmente apresentamos as nossas conclusões e apontamos direcções para possíveis melhorias.

## 2. LÓGICA DE RAMIFICAÇÃO

A lógica de salto (“skip”) ou de ramificação (“branching”) é uma característica importante nos sistemas comerciais que produzem inquéritos “online”. Permitem que as pessoas que respondem aos inquéritos não vejam todo o inquérito e não sejam obrigadas a responder a todas as perguntas podendo seguir percursos customizados do questionário. Por exemplo, só fará sentido perguntar para que países uma determinada empresa exporta os seus produtos caso essa empresa seja uma empresa exportadora. Assim, poder-se-á declarar que só serão colocadas certas perguntas dependendo da natureza das respostas a outras perguntas.

Tipicamente, os sistemas comerciais existentes fornecem a capacidade de declarar a lógica da ramificação em que podemos declarar, por exemplo, que se a resposta à pergunta 18 for “sim” então salta para a pergunta 45, ou que independentemente da resposta à pergunta 46, o inquérito dá um salto incondicional continuando na pergunta 67. Esta capacidade de declarar a lógica de salto e de ramificação é algo altamente positivo porque permite que se produzam inquéritos ramificados em que certas perguntas ficam activas ou inibidas de acordo com a combinação de respostas. Estes sistemas permitem apenas declarar uma única pergunta para a qual se salta ou se ramifica na presença de uma determinada resposta. Notem que a forma como tipicamente se declara a lógica da ramificação obriga a que o “designer” do inquérito estabeleça todo a narrativa do inquérito à priori. A pergunta 18 sucede à 17 e a 19 sucede à resposta “sim” da pergunta 18 mas a 25 sucede à um “não” da 18, etc. etc. Mas, sobre a possibilidade de alterar o percurso narrativo (a ordem das perguntas) falaremos mais tarde na secção 4. Queremos apenas realçar o facto da ordem de apresentação das perguntas estar intrinsecamente ligada à ramificação do inquérito, o que para nós não faz sentido sendo uma limitação.

No nosso caso, separamos claramente a lógica de ramificação, ou de salto, da escolha e ordem de apresentação das perguntas. São dois processos completamente separados. A validade das perguntas, i.e., o facto de poder ser colocada, vai depender das respostas a outras questões mas isso não pressupõe uma

sequência pré-determinada de ordenação de perguntas. Podemos declarar quais as dependências entre as perguntas e as respostas, mas ao contrário dos sistemas que descrevemos podemos ter várias perguntas que dependem de uma ou mais respostas ou mesmo da ausência de respostas. Por exemplo, uma regra traduzida em linguagem natural pode ser:

*Regra1: Se temos uma resposta Xa ou Xb à pergunta 12 e uma resposta Ya ou Yb à pergunta 15 então as perguntas 18, 19, 20, 21 e 22 são válidas*

Podemos também ter factos e isso acontece para a série de primeiras perguntas que são sempre válidas e que não dependem de nenhuma resposta. Todas elas podem ser imediatamente colocadas ao utilizador. Por exemplo, podemos declarar:

*As perguntas 1,2,3,4,5 são válidas*

No que diz respeito à regra 1, após as respostas às perguntas 12 e 15, o inquirido não tem de saltar necessariamente para a 18 nem para nenhuma das outras dependentes {19,20,21,22}, mas essas perguntas tornam-se válidas, tornando-se passíveis de ser colocadas e serão em princípio colocadas a não ser que o utilizador interrompa o inquérito ou reveja eventualmente as respectivas respostas de tal modo que iniba essas perguntas.

Temos assim um conjunto de regras e de factos em Clips que exprimem as dependências entre as perguntas e as respostas e que estabelecem a cada momento o território do questionário, ou seja o conjunto de perguntas passíveis de serem colocadas, as perguntas válidas (umas delas respondidas, outras adiadas e outras ainda não colocadas). No reverso da medalha, estas regras impedem de serem colocadas perguntas que não fazem sentido. Também inibem as respostas a perguntas que deixaram de fazer sentido (inválidas) e que foram feitas no passado, mas que devido a um processo de revisão deixaram de fazer sentido. As respostas ficam em “background” mas estão inibidas porque correspondem agora a perguntas inválidas mas que foram válidas no passado.

Queremos também exprimir dependências fortes entre algumas perguntas e respostas que podem condicionar a ordem das perguntas colocadas. Por exemplo, se queremos perguntar se uma companhia exporta (questão 10) e se queremos saber para que países ela exporta (questão 11) faz sentido que estas duas perguntas sejam colocadas em sequência. Para isso, podemos declarar que a questão 11 depende fortemente da resposta positiva à pergunta 10 com o facto seguinte:

(dependência 10 “sim” 11)

indicando que se recomenda que a pergunta 11 se segue naturalmente à resposta “sim” da pergunta 10. Esta declaração não obriga a que a ordem recomendada seja cumprida mas pode ser levada em conta no módulo que selecciona a próxima pergunta a ser colocada.

O módulo de dependências entre as perguntas faz uso da base de conhecimento de dependências (factos e regras) e calcula a cada momento quais as perguntas que são válidas de acordo com o perfil de respostas do utilizador.

### **3. REVISÃO DAS RESPOSTAS E SUAS CONSEQUÊNCIAS**

Uma das críticas aos inquéritos “online” é a incapacidade de o utilizador ter a liberdade de poder navegar livremente no inquérito, podendo escolher perguntas anteriormente colocadas, reanalisá-las e até rever as suas respostas. O utilizador pode ser obrigado a adiar uma resposta por falta de dados suficientes naquele momento e querer regressar à mesma pergunta mais tarde quando tiver recolhido a informação necessária. Ele pode esperar que o módulo de selecção de perguntas lhe coloque essa pergunta mas é muito mais prático ter a liberdade de a seleccionar directamente e gravar a sua resposta. Esta liberdade de andar para trás e para a frente no inquérito pode ser uma fonte de problemas já que as mudanças realizadas numa parte do inquérito podem alterar a cadeia de dependências e invalidar algumas respostas. Basta pensar na alteração da resposta a

uma pergunta situada num nó de ramificação. A resposta a essa pergunta conduziu o inquérito ao longo de um ramo, mas a sua posterior alteração pode obrigar a mudar de ramo, alterando a estrutura do inquérito e invalidando um conjunto de respostas já gravadas.

Nós permitimos que o utilizador viaje para a frente e para trás no inquérito, mas só entre perguntas que são válidas, como é facilmente compreensível. Como é recalculada a cada momento a cadeia de dependências gerando as perguntas que são válidas após cada interacção com o utilizador, o sistema mantém sempre a coerência. Neste caso, sempre que um utilizador responde pela primeira vez a uma pergunta ou quando altera uma resposta o sistema recalcula todas as perguntas que dependem da situação corrente, garantindo sempre um inquérito coerente. Mesmo que existam respostas a perguntas que eram válidas antes de uma revisão e que agora deixaram de ser válidas, elas não contam mais para o estado actual do inquérito. Mantemos essas respostas para o caso de, mais tarde, uma nova revisão poder dar de novo sentido a essas respostas, tornando-as válidas.

Uma solução alternativa, mais sofisticada e mais eficiente consistiria em utilizar um Sistema de Manutenção de Crenças (Doyle-1977), que apenas recalcularia as perguntas que dependem directamente da nova resposta mantendo a coerência. No entanto isso obrigaria a que o Sistema de Regras estivesse activo ao longo das várias sessões, guardando o estado das inferências entre respostas o que não acontece devido à particularidade da interacção entre o servidor “ Web” e o sistema de regras, que será explicado na secção 6. A nossa solução é bastante admissível dada a pequena dimensão do questionário (53 perguntas).

#### **4. REGRAS DE VALIDAÇÃO E DE VERIFICAÇÃO DE INCONSISTÊNCIAS**

Num questionário é normal colocarem-se perguntas cujas respostas são inconsistentes. As perguntas são feitas por vários elementos de uma equipa e nem sempre é possível nem talvez desejável depurar o questionário de modo a que não seja possível obter respostas inconsistentes. Assim, precisamos de um módulo que olhe para as várias respostas e detecte as inconsistências que serão tratadas pelo utilizador. Precisamos também de poder expor, modificar e fazer evoluir de um modo faseado o nosso conhecimento sobre qual a informação que é incompatível. Um sistema baseado em regras pareceu-nos ser a ferramenta ideal para atingir estes objectivos. Por um lado, permite (1) separar o conhecimento das inconsistências e torná-lo independente dos dados e do código do programa; (2) torna-o explícito e acessível e (3) é fácil de modificar e de evoluir sem incomodar os outros componentes que compõem o questionário. No fundo as 3 grandes qualidades dos sistemas de regras. Damos dois exemplos de regras de detecção de inconsistências, no âmbito do inquérito ao design na indústria de transformação. Estas regras estão escritas em CLIPS mas foram interpretadas em linguagem natural para melhor compreensão.

*Regra1: Se a empresa “não olha” para os concorrentes (resposta à pergunta 27) e conhece os contributos do design nos líderes do mercado (resposta à pergunta 29) então inconsistência entre pergunta 27 e a 29*

*Regra2: Se é dito que a empresa é pró-activa antecipando mercados (resposta à pergunta 18) e não dá importância ao mercado e aos consumidores (resposta à pergunta 28) então inconsistência entre a pergunta 18 e 28.*

No nosso caso, optámos por enviar um aviso ao utilizador sempre que provoque uma determinada inconsistência pela primeira vez. O aviso é acompanhado de uma explicação da inconsistência. Não impedimos os utilizadores de continuarem a responder ao inquérito quando não resolvem as inconsistências mas mantemos sinalizadas as perguntas cujas respostas estão a provocar inconsistências. Antes de lacrar o inquérito, o utilizador é de novo avisado que existem inconsistências e pode modificar as respostas às perguntas envolvidas.

## 5. NARRATIVA DO INQUÉRITO: REGRAS PARA SELECIONAR A PRÓXIMA PERGUNTA A SER COLOCADA

Iremos passar a descrever o sistema de selecção da próxima pergunta a ser colocada. Queremos dotar o questionário com a capacidade de ser flexível no que diz respeito à ordem de colocação de perguntas. Queremos escapar-nos da inevitabilidade de uma narrativa monolítica completamente definida a priori. Para isso, podemos definir um conjunto de regras de selecção da próxima perguntas a apresentar ao utilizador. Só pode ser seleccionada uma pergunta que é válida, uma pergunta que pode ser colocada. É importante lembrar que são as regras que geram as dependências, referidas na secção 2, que determinam quais as perguntas que são válidas. A selecção da próxima pergunta a ser colocada pode depender de um conjunto de elementos, uns genéricos a todos os inquéritos, como por exemplo a existência de certas respostas a determinadas perguntas, mas pode também depender de informação que existe apenas num inquérito particular. Vamos dar dois exemplos de regras que utilizámos, traduzindo-as de CLIPS para linguagem natural.

*Regra1: Se o utilizador acabou de fazer login e é o primeiro login então coloca a pergunta 1.*

*Regra2: Se o utilizador acabou de fazer login e não é o primeiro login então coloca uma pergunta válida, caso exista, que já foi apresentada antes e não foi respondida, mas que seja diferente da última colocada e que não tenha sido colocada mais do que 2 vezes.*

Vejam que estas regras pressupõem que as perguntas não sejam respondidas e que se adiem as suas respostas, o que acontece no inquérito que fizemos, e pressupõem também que contemos o número de vezes que isso acontece. Implicam também que conheçamos se o utilizador faz ou não “login” pela primeira vez e qual foi a pergunta que foi colocada pela última vez—essa informação está disponibilizada no nosso inquérito.

Poderíamos facilmente reescrever a primeira regra de um modo diferente caso o inquérito possa começar por qualquer das 5 primeiras perguntas.

*NovaRegra1: Se o utilizador acabou de fazer login e é o primeiro login então coloca qualquer das perguntas válidas pertencentes ao conjunto {1,2,3,4,5}, escolhida aleatoriamente.*

Poderíamos, por exemplo, classificar as perguntas em “pesadas” e “leves” e definir regras que tentem evitar que sejam apresentadas consecutivamente duas perguntas “pesadas”. Ou podemos não colocar as pergunta mais chatas nem no princípio nem no final do inquérito, como é geralmente aconselhado. Notem que embora este componente permita representar o conhecimento de um perito em inquéritos, nós não consultámos nenhum perito particular mas usámos o nosso bom senso na expressão das regras que escolhem a próxima pergunta a ser colocada.

O facto de utilizarmos um sistema de regras tem, de novo, principalmente três vantagens: (1) a transparência das regras ao exporem a lógica que determina a selecção da próxima pergunta; (2) podem ser facilmente alteradas e afinadas; (3) podem ser modificadas independentemente dos outros aspectos, não tendo que para isso mexer no código do programa que gere o inquérito.

## 6. ARQUITECTURA DO QUESTIONÁRIO “ONLINE”

A decisão de optarmos por um sistema baseado em regras para lidar com a parte da lógica do inquérito, obrigou a pensar uma solução que fosse simultaneamente exequível num curto prazo de tempo e ao mesmo tempo pudesse ser generalista e facilmente acedível por todos os utilizadores que iriam responder ao inquérito. Este número rondava os 3000 e fazem parte de uma base de dados da indústria portuguesa, que foi fornecida pela equipa do De.:SID. A primeira decisão que foi necessário tomar teve a ver com a forma de distribuição do inquérito. Ou se criava um sistema centralizado ou se construía uma aplicação que os

utilizadores pudessem fazer download e utilizar a partir dos seus computadores em modo offline enviando no final o inquérito. Atendendo à diversidade de utilizadores foi decidido que se utilizaria a primeira opção optando por criar uma aplicação Web a partir da qual os utilizadores fariam o input das suas respostas. O “browser” e o acesso à internet tornou-se de tal forma ubíquo que imediatamente se apresentou como solução para um processo que se queria distribuído e de largo alcance.

A decisão de optar por uma aplicação Web conduziu invariavelmente a uma arquitectura dividida em módulos. De entre os diversos sistemas de implementação possíveis optou-se por utilizar uma stack LAMP (Linux, Apache, MySQL e PHP) de suporte ao projecto permitindo o seu desenvolvimento a partir daí.

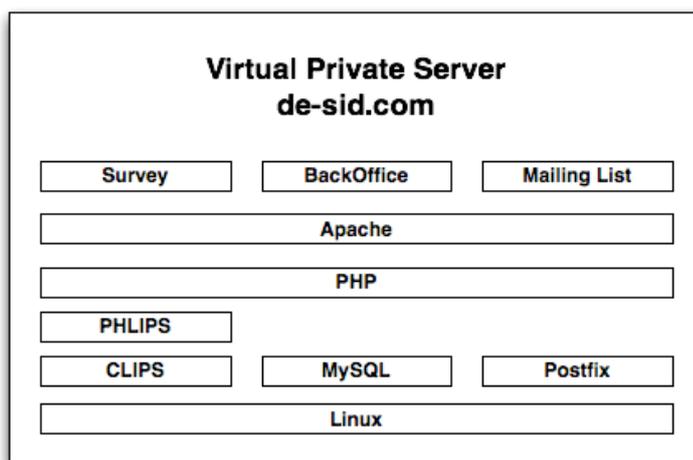


Figura 1 - Diagrama da implementação do de-sid.com

A decisão de optar por um sistema baseado na Web permitiu pensar mais duas aplicações, para além do próprio inquérito. Uma tratou-se de um backoffice para a gestão das bases de dados das empresas, uma vez que se tinha verificado até ao presente que a informação estava dispersa havendo muitas duplicações assim como alguns erros nos registos. A outra foi a criação de implementação de um sistema de mailing list para comunicar mais rapidamente com os utilizadores. Tal levou a que fosse instalado e configurado um servidor de correio electrónico dedicado apenas às tarefas do servidor, tendo recaído a escolha sobre o Postfix como MTA.

O inquérito em si foi desenvolvido em PHP, utilizando preferencialmente a extensão/módulo PHLIPS para fazer a comunicação com o motor de inferência CLIPS. O CLIPS (“C Language Integrated Production System”) é uma ferramenta para o desenvolvimento de “expert systems” desenvolvido pela NASA em 1985 e é ainda hoje um dos mais utilizados por estar no domínio público. Sendo escrito em C necessitava de algum tipo de interface para comunicação com o inquérito, sendo que se optou pelo PHLIPS que funciona como módulo para o PHP.

## 7. INTERACÇÃO COM O SISTEMA DE REGRAS

Como vimos na secção anterior, a implementação inclui o módulo PHLIPS que vai fazer a interacção entre a aplicação Web (o inquérito) e o motor de regras (CLIPS). A interacção entre os dois níveis é feita sempre que é necessário decidir sobre o que deve ser apresentado ao utilizador. Isto acontece de cada vez que o utilizador do inquérito faz login, adia a resposta a uma pergunta, ou grava uma resposta. A cada alteração do perfil do utilizador é necessário transmitir esses dados ao motor de regras que após processar a informação disponível indicará como proceder a seguir.

Assim, a evolução dos acontecimentos, em cada interacção, pode ser sucintamente esquematizada:

1. O utilizador executa uma acção, que altera o seu historial, perfil, etc...
2. Inicializa-se o motor de regras.
3. Carrega-se o conjunto de regras utilizadas no inquérito.
4. Fornece-se ao motor o estado do perfil do utilizador através de um conjunto de factos.
5. Executa-se o processo de decisão do motor de regras.
6. Obtém-se o conjunto de resultados que serão actualizados pela aplicação Web.

A interacção com o sistema de regras é efectuado de cada vez que é preciso decidir sobre o que o utilizador deve fazer a seguir ou sobre o que lhe deve ser apresentado ao nível de inconsistências entre determinadas respostas. Como estamos perante uma aplicação Web, em que a persistência dos dados é assegurada por uma base de dados MySQL, isso implica que a persistência dos dados ficou atribuída à aplicação Web, sendo que o motor de inferência é activado a cada chamada à camada de decisão e sendo necessário portanto passar-lhe de cada vez o estado do utilizador em causa, que corresponde ao número de logins efectuados e às respostas dadas. Naturalmente que a repetição sistemática da activação e inicialização do

motor de inferência advém também da própria arquitectura escolhida, uma vez que não seria possível ter inúmeras instâncias do motor de inferência a correr esperando eventuais acções por parte do utilizador. Assim, a persistência de estados é efectuada do lado da aplicação Web (através da base de dados MySQL) e não através do motor de regras. No futuro será possível explorar outras alternativas de persistência nomeadamente aproveitando as novas tecnologias que explorem o lado do cliente, como sejam o caso do Adobe Air ou Silverlight da Microsoft e também incluir questões essenciais como a mobilidade dos utilizadores, sendo que fará sentido então adaptar o sistema por forma a que este leve em linha de conta, por exemplo, o aspecto da ligação móvel através de aparelhos como os telemóveis. O motor de regras poderá aí ter um papel importante na adaptação do inquérito às particularidades de cada utilizador.

## 8. CONCLUSÃO

Aplicámos tecnologias associada à Inteligência Artificial, nomeadamente os sistemas periciais, aos inquéritos “online” com o objectivo de melhorar a sua usabilidade, flexibilidade e a adaptabilidade, no que diz respeito ao utilizador. No entanto, o facto de descrevermos a componente lógica do inquérito expôs essa lógica tornando-a transparente e tem como consequência a facilidade de modificação do inquérito sem incomodar os outros componentes do programa. Para além disso permitiu-nos também separar e controlar os três aspectos fundamentais do inquérito online, as dependências entre perguntas, a detecção semântica através da análise de conteúdo para controlo de inconsistências e a decisão de selecção de questões a colocar futuramente.

A flexibilidade de um sistema deste tipo revela-se particularmente útil em duas situações. Adaptação do inquérito em tempo real sem que se afecte significativamente a percepção que o utilizador tem do inquérito, e o ajuste do mesmo a cada um dos utilizadores por forma a que cada inquérito seja único e ao mesmo tempo permitindo ainda recolher a informação necessária para os investigadores. Ainda ao nível de implementação esta solução modular apresenta vantagens uma vez que poupa o tempo de desenvolvimento e isola componentes que podem ser colocados em diferentes níveis e tratados independentemente.

O sistema implementado permite ainda que no futuro se possam explorar novas soluções de implementação e distribuição do inquérito centrando ainda mais a parte adaptabilidade do sistema do lado do utilizador em vez de ser do lado do servidor. Um tipo de solução poderá por passar pela criação de um inquérito gerido por “agentes” que seriam distribuídos aos utilizadores sob a forma de uma pequena aplicação e que seriam responsáveis por apresentar a um utilizador o inquérito segundo um perfil local de utilizador, reportando apenas no final, ou em pontos de sincronização pré-definidos, os resultados com o servidor central. Tal tipo de solução é facilmente implementável a partir do sistema actual e poderá no futuro ajudar a obter resultados ao inquérito de forma mais rápida e segura uma vez que num sistema deste género os perfis de utilizador não seriam guardados no servidor central, assegurando assim uma maior privacidade dos seus dados.

## AGRADECIMENTO

Queremos agradecer a todos os elementos do grupo De.:Sid ( <http://www.de-sid.com> ). Queremos também agradecer à Fundação para a Ciência e Tecnologia (FCT) que suporta este projecto com a referência PTDC/AUR/70607/2006 com o título: O Design como recurso estratégico das empresas: um estudo do impacto do Design”.

## REFERÊNCIAS

- Crighton D., 2005. *Adding a Truth Maintenance System to ERA, the Electronic Referee Assistant, to allow backtracking*. Master of Science, University of Edimburgh.
- Debenham, J., K. 1989. *Knowledge Systems Design*. Prentice Hall.
- Doyle, J. 1977. Truth Maintenance Systems for problem solving. *5<sup>th</sup> International Joint Conference on Artificial Intelligence*.
- Edwards, A., Connel, A.A.D. 1989. *Expert Systems in Accounting*. Herfordshire, UK: Prentice Hall International (UK) Ltd.
- Forsyth, R. 1984. *Expert Systems: Principles and Case Studies*. London Chapman and Hall Computing.
- Giarratano, J., Riley, G., 2004. *Expert Systems: principles and programming*. Course Technology.
- Graham, I. 2006. *Business Rules Management and Service Oriented Architecture: a pattern language*. New York: Wiley.
- Halle, B. von ,2002. *Business Rules Applied*. New York: Wiley.
- Harmon, P., King, D., 1985. *Expert Systems: Artificial Intelligence in Business*. New York: Wiley.
- Jackson, P. 1998. *Introduction to Expert Systems*. Addison Wesley.
- Ross, R. G. 1997. *The Business Rule Book*. (2nd ed.). Houston: Business Rule Solutions, LLC.
- Urbano P., Rodrigues, D. 2008. Rule Based Systems Applied to Online Surveys. *Proceedings of IADIS International Conference WWW/INTERNET*. Freiburg.
- Waterman, D. A., *A Guide to Expert Systems*. 1986. Reading, MA: Addison-Wesley.
- Winston, P. H., Prendergast, K.A. (editors) 1984. *The AI Business: Commercial Use of Artificial Intelligence*. Cambridge, MA. The MIT Press.
- CLIPS *Expert System Shell* <<http://www.ghg.net/clips/CLIPS.html>>.
- IdeaScale <<http://www.ideascale.com/>>
- PHLIPS *PHP extension for CLIPS*. <<http://phlips.sourceforge.net/>>
- QuestionPro <<http://www.questionpro.com/>>
- Questionform <<http://questionform.com/>>
- SurveyMonkey <<http://www.surveymonkey.com/>>