

Sistema de Atendimento Telefónico Automatizado dos SDUA (SATA-SDUA)

Carlos Rui Gouveia Carvalhal, Arménio Augusto Arvins Pereira Pinto,
Manuel Agostinho de Oliveira Romeiro

Resumo - Neste artigo apresenta-se um Sistema de Atendimento Telefónico Automatizado, baptizado com a designação SATA-SDUA (Sistema de Atendimento Telefónico Automatizado dos Serviços de Documentação da Universidade de Aveiro), capaz de disponibilizar, via telefone, algumas das funcionalidades acessíveis através do Servidor WWW dos SDUA. Adicionalmente este sistema disponibiliza uma Interface de Gestão, acessível via WWW, que permite ao Administrador configura-lo, geri-lo e obter relatórios da sua utilização. O SATA-SDUA apresenta uma Arquitectura Distribuída baseada no Modelo Cliente-Servidor, permitindo assim que os diferentes processos que o compõem possam ser dispersos por várias máquinas. Foi desenvolvido usando, essencialmente, a Linguagem Java o que o torna portátil, económico e de fácil manutenção.

Abstract - This paper presents an Automated Phone Attendance System, named SATA-SDUA (Automated Phone Attendance System of the Aveiro University Documentation Services), capable to supply, by telephone, some of the functionalities supplied by the SDUA WWW Server. Otherwise, it disposes an Administration Interface, accessible by WWW, which allows the Administrator to configure it, manage it, and to obtain reports about the uses of it. SATA-SDUA presents a Distributed Architecture based on the Client-Server Model, allowing, in that way, the dispersion of its constituent processes by several computers. It was developed using, essentially, the Java Programming Language that turns it portable, economic, and of easy maintenance.

I. INTRODUÇÃO

Apesar dos SDUA (Serviços de Documentação da Universidade de Aveiro) disporem de um sistema de atendimento acessível via WWW, que permite aos seus utentes consultarem informação relevante sobre o funcionamento da Biblioteca e realizarem múltiplas tarefas sobre o seu Registo Bibliotecário (Consulta/Renovação de Empréstimos, Criação/Consulta/Anulação de Reservas, Consulta de Empréstimos com Prazo de Entrega ultrapassado, etc.), verifica-se, no entanto, que os seus funcionários são muitas vezes inquiridos, telefonicamente, pelos utentes com questões quer sobre o funcionamento da Biblioteca quer sobre o seu Registo Bibliotecário, o que provoca uma

grande sobrecarga sobre os funcionários, em detrimento da qualidade do serviço prestado “ao balcão”.

Com o intuito de melhorar esta situação, surgiu a ideia SATA-SDUA (Sistema de Atendimento Telefónico Automatizado dos SDUA), um Sistema de Atendimento Telefónico Automatizado capaz de transpor para o telefone algumas das funcionalidades disponibilizadas pelo Servidor WWW dos SDUA. Este sistema deveria ser tal que:

- A interface com o Utente SDUA fosse implementada por um equipamento terminal telefónico com capacidade de marcação DTMF (Dual Tone Multi-Frequency). Dessa forma a interacção no sentido Utente SDUA→SATA-SDUA seria efectuada através do teclado do telefone e a interacção no sentido contrário através da reprodução de ficheiros áudio.
- Permitisse aos Utes dos SDUA consultar informação relevante sobre o funcionamento da Biblioteca, assim como também realizar as seguintes operações sobre o seu Registo Bibliotecário:
 - Consultar/Renovar Empréstimos.
 - Consultar/Anular Reservas.
 - Consultar Empréstimos cuja data de devolução tenha sido ultrapassada (Multas).
- Implementasse um sistema de segurança através de mecanismos de autenticação dos utilizadores do Sistema (Administrador SATA-SDUA e Utente SATA-SDUA), assim como também através da definição dos seus papéis (atribuição de permissões).
- Implementasse uma interface de gestão, acessível via WWW, que permitisse ao Administrador SATA-SDUA configurar o Sistema, geri-lo e obter relatórios da sua utilização.

II. ARQUITECTURA DO SISTEMA

O SATA-SDUA apresenta uma arquitectura baseada no Modelo Cliente-Servidor como mostra a Figura 1 que apresenta a sua arquitectura e fluxo de dados. Nesta Figura é possível distinguir um conjunto de processos servidores (Servidor SATA-SDUA e Servidor Administrador SATA-SDUA), um conjunto de processos de suporte aos servidores (Gestor de Comunicação, Gestor da Base de Dados de Administração e Gestor da Base de

Dados dos SDUA), um conjunto de Bases de Dados (Base de Dados de Administração e Base de Dados dos SDUA), geridas por dois dos processos de suporte, e finalmente por um conjunto de processos de interacção com os utilizadores (Cliente Administrador SATA-SDUA e Gestor da Placa Voice-Modem). Estes processos e Bases de Dados podem, por sua vez, serem agrupados, em termos funcionais, em 2 sub-sistemas, o Sub-Sistema do Administrador, que engloba todas as componentes destinadas à gestão do sistema, e o Sub-Sistema do Utente, que engloba todas as componentes envolvidas na interacção (em sentido lato) com o Utente SATA-SDUA, conforme apresentado na Figura 1.

O recurso a esta arquitectura, desde que suportada pela tecnologia apropriada, possibilita a sua implementação num ambiente distribuído, distribuindo, assim, os diferentes blocos funcionais do sistema por várias máquinas, o que permite por um lado a distribuição da carga de processamento e a especialização de cada um dos sistemas computacionais e por outro lado a implementação de níveis de segurança suplementares.

A. Processos Servidores

Servidor SATA-SDUA: Este processo desempenha o papel central do Sub-Sistema do Utente, sendo, dessa forma, o responsável pela gestão e interligação entre os restantes processos que o integram. Sempre que o seu processo Cliente, o processo Gestor de Comunicação, emitir um pedido de autenticação ou de consulta/alteração do registo bibliográfico dum Utente SATA-SDUA, este processo iniciará a negociação com os processos Gestor da Base de Dados dos SDUA e Gestor da Base de Dados de Administração no sentido de satisfazer tal pedido e registar a operação efectuada, respondendo adequadamente ao processo chamador.

Servidor Administrador SATA-SDUA: Processo central do Sub-Sistema do Administrador, sendo o responsável pela autenticação do Administrador SATA-SDUA e satisfação dos pedidos de gestão dele emanados.

B. Processos de Suporte aos Servidores

Gestor de Comunicação: Processo de interface entre o Gestor da Placa de Voice-Modem e o Servidor de Comunicação. Obterá do segundo a informação necessária à satisfação do pedido emitido pelo primeiro e elaborará a

resposta que o Gestor da Placa de Voice-Modem apresentará ao Utente SDUA (mensagem de áudio).

Gestor da Base de Dados de Administração: Processo responsável pelo registo, na Base de Dados de Administração, da informação de monitorização de todos os acessos efectuados pelos Utentes SDUA ao sistema.

Gestor da Base de Dados dos SDUA: Este é o Processo responsável pela comunicação com a Base de Dados dos SDUA, implementando dessa forma a interface do sistema com essa Base de Dados exterior ao sistema. Para tal, formulará os quesitos apropriados à obtenção da informação necessária à sua resposta, transmitindo-a de seguida para o processo Servidor SATA-SDUA.

C. Processos de Interacção com os Utilizadores

Gestor da Placa Voice-Modem: Processo de interacção com o Utente SATA-SDUA. É o processo responsável pela monitorização, controlo e gestão da placa Voice-Modem e do telefone DTMF que implementa a interface com o Utente SATA-SDUA. Sempre que houver uma interacção por parte do Utente SATA-SDUA comunicará com o processo Gestor de Comunicação no sentido desse pedido encontrar resposta.

Cliente Administrador SATA-SDUA: Processo de interacção com o Administrador SATA-SDUA. Dessa forma, implementará uma interface gráfica amigável, acessível via WWW, e apropriada às operações de gestão a realizar por este, nomeadamente consulta estatística da utilização do sistema e dos registos de sessão (sequência de operações realizadas pelo Utente SATA-SDUA em cada acesso ao sistema).

D. Bases de Dados

Base de Dados de Administração: Base de Dados na qual serão registados os acessos e operações realizadas, em cada sessão, pelos Utentes SATA-SDUA. Esta informação permitirá ao processo Servidor de Administrador SATA-SDUA elaborar relatórios da utilização do sistema, os quais serão úteis, nomeadamente, na resolução de situações problemáticas.

Base de Dados dos SDUA: Base de Dados com a informação sobre o Arquivo Bibliográfico dos SDUA e os seus Utentes. Apesar de estar representada na Figura 1, trata-se duma Base de Dados exterior ao Sistema.

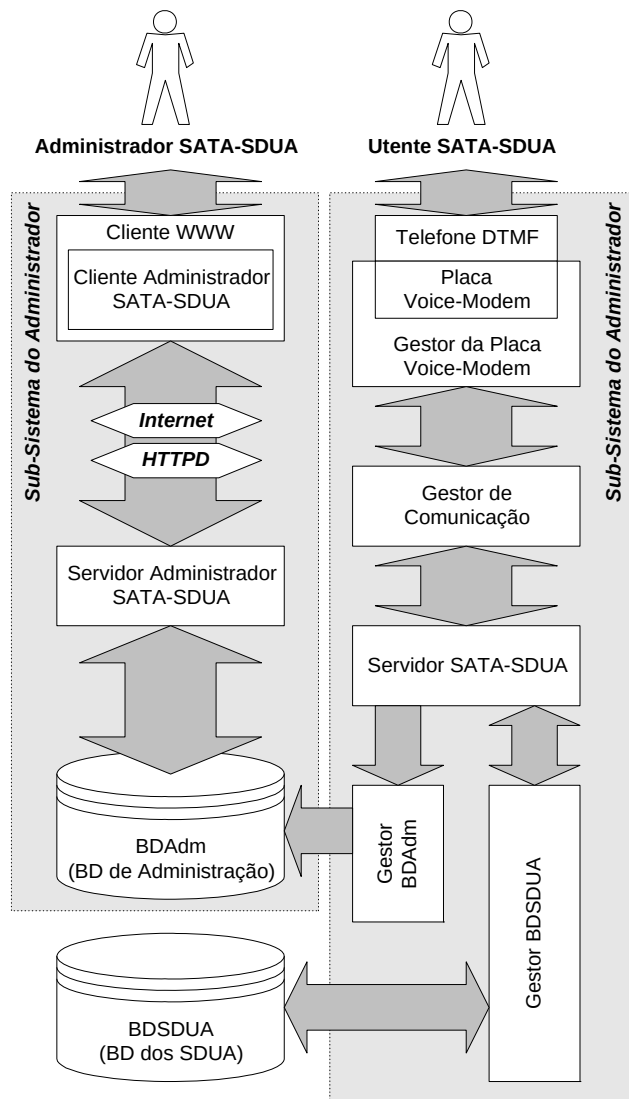


Figura 1: Arquitectura SATA-SDUA.

III. OPÇÕES TECNOLÓGICAS

A implementação da arquitectura SATA-SDUA atrás descrita foi precedida de um estudo sobre as diferentes opções tecnológicas disponíveis, por forma a serem escolhidas aquelas que melhor se adaptavam aos objectivos pretendidos. Dessa forma, houve necessidade de escolher uma linguagem de programação, um mecanismo de comunicação entre processos, um mecanismo de comunicação/controlo duma placa de Voice-Modem e um mecanismo de acesso a Bases de Dados Relacionais.

A. Linguagem de Programação

Quanto à Linguagem de Programação a escolha recaiu sobre a Linguagem Java. As razões que levaram a esta escolha são várias: em primeiro lugar, o sistema seria portátil; em segundo lugar, a complexidade do desenvolvimento seria menor do que no caso, por exemplo, de C/C++, uma vez que a Linguagem Java é

consideravelmente mais simples; finalmente, como os melhores compiladores de Java são gratuitos, seria uma solução mais vantajosa em termos económicos [1]. Dessa forma foi usada a Linguagem Java na implementação de todas as componentes do sistema com excepção da componente de Comunicação/Controlo da Placa de Voice-Modem. A razão para tal prende-se com a indisponibilidade de uma API em Java capaz de comunicar com a Placa Voice Modem. Consequentemente, houve necessidade de dividir a componente de Comunicação/Controlo da Placa de Voice-Modem em dois processos: um desenvolvido em C/C++ que comunica com a Placa Voice-Modem (Gestor da Placa de Voice-Modem), e um desenvolvido em Java que implementa a interface com o resto do sistema (Gestor de Comunicação). [7]

B. Comunicação entre Processos

Os mecanismos de IPC (Inter-Process Communication) considerados foram os seguintes: Sockets, RPC (Remote Procedure Call), CORBA (Common Object Request Brokerage Architecture) e JRMI (Java Remote Method Invocation). Como estamos perante um ambiente de programação homogéneo (excluindo o processo Gestor da Placa Voice-Modem que é isolado do resto do sistema pelo processo Gestor de Comunicação) e a informação a trocar é relativamente estruturada (Objectos), Sockets e RPCs foram logo excluídos; ficando a decisão final limitada ao CORBA e ao JRMI. Em termos gerais, CORBA e JRMI são muito idênticos; no entanto, enquanto que o primeiro suporta ambientes de programação heterogéneos (pode ser utilizado na comunicação entre processos que utilizem linguagens de programação diferentes), o segundo está restrito à plataforma Java. Sendo assim, JRMI é muito mais simples de utilizar. Além disso, enquanto que a implementação de JRMI é completamente gratuito, as de CORBA são caras, poucas e incompletas. Dessa forma, a nossa decisão recaiu sobre o JRMI. [2, 3, 4]

O Java RMI é um ambiente de programação distribuído, orientado a objectos e homogéneo. O seu principal objectivo é permitir que os programadores de Java possam desenvolvam aplicações distribuídas, com a mesma sintaxe e semântica das aplicações não-distribuídas.

C. Comunicação/Controlo de uma Placa Voice-Modem

A palavra “consenso” não é a mais apropriada para caracterizar a indústria das telecomunicações; de facto, a competição é tão grande neste campo, que praticamente cada fabricante possui os seus próprios standards para cada uma das áreas existentes. A área da telefonia assistida por computador não foge à regra; os standards que suportam alguma variedade de dispositivos são poucos, mal construídos, e as suas implementações são caras e na maior parte das vezes inacabadas. Mesmo

assim, existem duas grandes soluções: TAPI (Telephony API) e JTAPI (Java Telephony API).

Em termos gerais, a arquitectura da segunda é muito superior à da primeira; isto deve-se não só ao facto de ser mais actual, como também de resultar do esforço de um maior número de fabricantes. No entanto, ainda não existe uma implementação completa da JTAPI, pelo que fomos obrigados a optar pela TAPI [5]. Como esta é uma API de C/C++, a componente de comunicação/controlo da Placa Voice-Modem teve que ser desenvolvida em duas linguagens de programação diferentes, Java e C/C++, tal como já foi atrás referido.

A TAPI é uma API que permite a utilização de serviços de telefonia em computadores pessoais. O termo telefonia refere-se às tecnologias utilizadas para transmitir a voz, de uma pessoa para outra pessoa, ao longo de uma determinada distância. Sendo assim, o objectivo da TAPI é possibilitar que os computadores substituam uma dessas pessoas, ou mesmo as duas. A única implementação que existe foi desenvolvida sob a forma de uma API de C/C++ pela Microsoft, e acompanha todos os sistemas operativos da plataforma MS Windows. A arquitectura da TAPI está representada na Figura 2. [6]

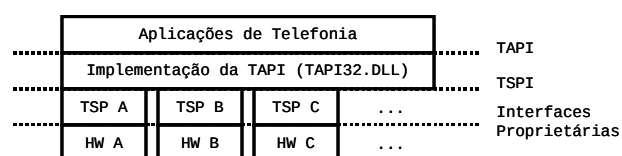


Figura 2: Arquitectura da TAPI.

Os TSPs (Telephony Service Providers), a camada de software imediatamente inferior à TAPI, são, normalmente, fornecidos pelos fabricantes, no entanto, existe um TSP genérico, o UNIMODEM/V, que acompanha os sistema MS Windows e que é capaz de funcionar com uma grande variedade de dispositivos. [6]

Em termos de funcionalidades, a TAPI apresenta as seguintes características:

- Controlo centralizado nas chamadas;
- Acesso ao media através das APIs standard;
- Independência da rede;
- Independência do modelo de ligação;
- Independência da plataforma (sempre que possível);
- Possibilidade de partilha das linhas telefónicas entre diversas aplicações.

D. Acesso a Bases de Dados Relacionais

Em Java existe uma solução de acesso a Base de Dados extremamente fiável e simples de utilizar, a JDBC (Java Database Connectivity). Sendo assim, não foi necessário procurar muito para chegar a uma solução.

A JDBC é uma API orientada a objectos e independente da plataforma, que fornece aos programadores de Java uma solução para o acesso a BDs relacionais. Tal como a maioria das APIs de Java, o objectivo do seu modelo é a simplicidade de utilização. [8]

IV. CONCLUSÕES E TRABALHO FUTURO

O SATA-SDUA, cumpre satisfatoriamente os objectivos propostos, com a mais-valia de apresentar uma Arquitectura Distribuída Cliente-Servidor, o que permite que as suas diferentes componentes sejam dispersas por várias máquinas. O facto de ter sido desenvolvido usando, essencialmente, a linguagem Java torna-o portátil, económico e de fácil manutenção. Apesar de actualmente ainda não estar a ser utilizado, acredita-se que tal venha a acontecer num futuro muito próximo, por se tratar dum produto completo, robusto e fiável.

Em termos de trabalho futuro, existem algumas melhorias que poderiam ser implementadas:

- Substituição da TAPI pela JTAPI, assim que houver uma implementação desta.
- Suporte para múltiplas chamadas em simultâneo.
- Parametrização dos diálogos de navegação através do uso de uma linguagem Scripting ou de técnicas de manipulação gráfica.

AGRADECIMENTOS

Aos SDUA, em especial à Dra. Emília Araújo, ao Eng. Filipe Bento e ao Miguel Pinto, pelos meios técnicos disponibilizados para a realização deste trabalho. A eles um muito obrigado.

REFERÊNCIAS

- [1] M. and Walrath Campione, The Java Tutorial, 2nd Edition, Sun Microsystems.
- [2] Govind Seshadri, Fundamentals of RMI, Java Developer Connection, 2000.
- [3] Java Remote Method Invocation, Revision 1.4, Sun Microsystems, 1997.
- [4] Java RMI Tutorial, Revision 1.3, Sun Microsystems, 1997.
- [5] Java Telephony API, Revision 1.1, Sun Microsystems, 1997.
- [6] C. Shells, Windows Telephony Programming, Addison Wesley, 1998.
- [7] K. Gregory, Special Edition using Visual C++ 6, Sams.net Publishing, 1998.
- [8] JDBC Guide: Getting Started, Sun Microsystems, 1997.