

Accept Mobile – A Mobile Tool for the SINMETRO Accept Information System

João Caldeira ¹, Edgar Dias ¹, Bruno Paulo ¹, Paulo Neves ¹

1) Departamento de Engenharia Informática, Escola Superior de Tecnologia de Castelo Branco,
Av. Do Empresário, 6000-767, Castelo Branco, Portugal
jaldeira@est.ipcb.pt, edgardias2005@gmail.com, sephipt@gmail.com, pneves@est.ipcb.pt

Abstract

Increasing demand of mobile applications for on-site data acquisition pushes the development of flexible and easy to use mobile tools, with great advantages over the traditional computer-based approaches. The Accept System from SINMETRO allows data gathering for quality control, in the form of Inspection sheets. Such tool allows quality assurance by monitoring some samples of a given material, for instance milk, wine, and even maintenance management.

This paper presents a mobile application in the Accept System that allows a Personal Digital Assistant (PDA) device to perform data gathering based on XML Inspection templates. Using .NET Compact Framework through C#, and database the technologies SQL Server and SQL Server CE, we developed Accept Mobile.

Accept Mobile uses the Remote Data Access (RDA) mechanism to send data over to the server through a synchronization service, while also providing the needed support for disconnected operation. We prove that the mobile application is very convenient and provides enough functionality for the user to dismiss the portable computer, although the main application was never developed with mobility concerns in mind.

Palavras chave: Mobile Computing, Remote Data Access, Information Systems, Information Retrieval

1. Introduction

In recent years, there has been a growing use of mobile systems, not only for personal and entertainment use, but also, for professional use. Such demand pushes the development of new and better mobile applications (Lim, 2005).

For the companies, the use of mobile systems became very helpful to access their Information Systems (IS) anywhere, and thus capitalize on the job done by its remote/mobile employees. Advantages arise like dismiss of the portable computer, which in many scenarios is not portable enough (data collection in small areas for instance), introducing usability issues. With these systems and their applications they can work directly in the main IS virtually anywhere, without being directly connected to the corporate Local Area Network (LAN) or Wireless Local Area Network (WLAN).

However, that benefit contributes with one more charge to the companies, who have to pay all the remote connections established by all the field-employees that are work in areas not acceded by the company's LAN/WLAN. At this point, is where the programmers may contribute to suppress this additional charge, evaluating if the mobile system's connection is essential to their work, and take into account the need for disconnected operation (Satyanarayanan, 2001).

This paper presents a PDA mobile application, Accept Mobile, created to collect field-captured data, enabling disconnected operation. With disconnected operation the device is able to provide

application services without server connection, thus presenting no additional charge for the company on data rates (GSM/UMTS).

The Accept Mobile application is based on the existing Accept System's application ACCEPT (SINMETRO, 2008), being able to provide the existing system with the ability to collect field data with the convenience of a very portable device. The advantage provided by the use of mobile devices is not only in ease of handling and transport, but also in its prices compared to the portable computers. Moreover some devices feature GSM/UMTS connectivity and Global Positioning System (GPS), effectively replacing in single device the portable computer, the mobile phones and GPS device, also referred as an *all-in-one* device.

Accept Mobile is not only an application to collect digital data on-site, it also features synchronization between the PDA's database (SQL Server CE®) and the ACCEPT main application's database (SQL Server 2005®) using the RDA technology.

Since this mobile application was developed only for this system, we didn't made any state-of-the-art analysis. Many reunions were made to understand the "client" needs, using UML tools, while also taking into consideration the recommendations of (Lim, 2005).

The remainder of the paper is organized as follows. In section 2, we introduce the Accept System and its working principle. In section 3 we dig into the technology that empowers our solution and in section 4 we present the mobile application. Finally, in section 5 we present some conclusions and lessons learned.

2. The Accept System

The Accept System provides a great range of services regarding metrology and quality, from consulting and planning, to the development of informatics applications and Information Systems (SINMETRO, 2008).

With the constant evolution of the technology, the Accept System aims to take their family of applications to be mobile. We use ACCEPT-Q from the Accept System to communicate with the mobile application.

The ACCEPT-Q is a system that enables an organization to make a statistical control of variables and attributes throughout its structure, for example, from performing inspections to, for instance, raw materials and subsidiary materials, variables of the process and inspection of finished products. (SINMETRO 2007) The Pocket PC application contributes to the ACCEPT-Q system, in field-captured data of variables and attributes, named as characteristics of one product, in order to enable statistical control by the ACCEPT-Q application. The data collection is performed in the form of Inspection "sheets". An inspection is a collection of records related to a number of features subject to a specification. This specification allows validate the records released. In the end must be assigned a result (OK or Not OK) to each inspection in accordance with a rule previously set. This rule, for example, can be defined by the number of records that are not in accordance to their defined specifications.

There are three main factors to define an Inspection:

- **Inspection Plan.** The Inspection Plan, groups one or more Analysis and defines the product's field which will be analyzed.
- **Analysis.** The Analysis, is a group of characteristics, corresponding to one product, which can be numeric or alphanumeric and have its own validation requirement.
- **Reference.** The Reference sets the product that will be analyzed according to an Inspection Plan.

In order to proceed to an Inspection, one must first select an Inspection Plan. Then an Analysis shall be selected, from the selected Inspection Plan's Analysis group. Finally the selection of the product Reference takes place. With all this information the application can now generate the user entered-form that provides the data collection mechanism.

3. Remote Data Access

SINMETRO clearly wanted to target the Microsoft® Windows Mobile® operating systems, so we had basically two choices here: with or without touch screen solutions. We opted for the touch screen solutions, also referred as Pocket PC. Although at the time SINMETRO did not use Microsoft® SQL Server, with their help we managed to migrate the important information to this database engine, allowing the use of Remote Data Access (RDA) technology. This technology, developed by Microsoft®, features SQL Server/SQL Server CE synchronization and data access, thus allowing an application running on the PDA, using an SQL Server CE® database, to access data from a remote SQL Server® database (Dove, 2005, TechNet, 2007).

RDA can be used for:

- **Downloading data.** The application can pull data from the remote SQL Server® database to a SQL Server CE® database.
- **Capturing data.** The application can collect data from user entered-forms and store that data in a SQL Server CE® database.
- **Uploading data.** The application can push the collected data, stored in a SQL Server CE® database, to a SQL Server® database, updating the SQL Server® database.

These techniques can be combined, creating a versatile application. For example on an application oriented for teachers, one can pull the data of their students from a SQL Server® database. During class the teacher can store various data from the student's performance on a SQL Server CE® database. Finally at the end of the day, upload the changed data back to the SQL Server® database. (Dove 2005; TechNet 2007)

On the client's side (mobile device), the SQL Server CE® Engine takes care of the data stored in the local SQL Server CE® database. The SQL Server® Client Agent serves as the layer for programmatically manipulating SQL Server CE®. It implements the SQL Server CE® object Remote Data Access (RDA).

On the server side, the SQL Server CE® Server Agent acts as the mediator, mediating between SQL Server® and SQL Server CE® for connectivity issues. All of these interactions are done via Hyper Text Transfer Protocol (HTTP), through the Internet Information Service (IIS) Web Server

The SQL Server CE® Client Agent replies to the RDA object methods calls that can perform the following actions:

1. **Pull.** Requests the tables recordset to the SQL Server CE® Server Agent. After receiving the recordset it stores the recordset in the SQL Server CE® database.
2. **Push.** Extracts all the inserted, updated, and deleted records, from the SQL Server CE® database and sends them to the SQL Server CE® Server Agent through HTTP.
3. **SubmitSQL.** Forwards the specified SQL request to the SQL Server CE® Server Agent through HTTP.

In response to the RDA object method calls, the SQL Server CE® Sever Agent can perform the following actions:

1. Pull. Receives the request from the SQL Server CE® Client Agent and returns the resulting recordset to the SQL Server CE® Client Agent through HTTP.
2. Push. Receives all the inserted, updated, and deleted records from the SQL Server CE® Client Agent, connects to SQL Server® through OLE DB, and inserts, updates, or deletes the records in the SQL Server® database.
3. SubmitSQL. Receives the specified SQL request from the SQL Server CE® Client Agent through HTTP, and invokes the client's SQL statement.

This mechanism presents several advantages like easy database access on both sides, together with management features, track changes, and data and schema definition occurs at the client when data is pulled from the server to the client. In terms of track changes, RDA uses system objects in SQL Server CE to track changes made to data. With such information, RDA pushes only the changes, specifically the changes rows on the tables, from client to the server, thus optimizing the connection.

However some disadvantages must also be considered, like the addition/removal of tables at the server is not automatically replicated to the client, it does not support automatic conflict management (it has to be a programmer's task at the client side), and finally it does not handle identity columns when pulling tables.

4. Accept mobile system

As the main goal of this application, running on the client's side (PDA), is to provide the user with real-time dynamically generated Inspection-Forms, which will be completed with field-captured data and stored in the PDA's database. Later on, the stored data, in the PDA's database (SQL Server CE®), may be "pushed" to the server's database (SQL Server 2005®), providing the synchronization of both databases.

Any solution starts with the requirements identification phase. This phase makes both parts involved (SINMETRO and the developers) agree on features and objectives of the application. Among the first identified requirements, we point out the synchronization between client and server databases, error/failure detection and recuperation, dynamic inspection forms generation, storage of complete inspection forms on the client, inspection forms push to the server, and provide mechanisms for easy configuration of connection parameters and support for configuration parameters reuse.

The XML creation system grabs information from the server and creates XML files that describe the possible inspection forms. This leads to the requirement that in the client the dynamic inspection forms must be generated through the knowledge of the Inspection Plan, the required analysis, and the product Reference. Moreover, stored inspections on the client must not be static, but allow view and edition.

The architecture of Accept Mobile system is depicted on Figure 1. The XML creator creates a shared folder accessible to the client, allowing different "inspection templates" to be loaded into the client. Afterwards, the Accept mobile is running, using wireless connectivity when needed, but providing disconnected operation through the SQL Server CE database. The process of dynamic Inspection-Forms generation is accomplished by the interaction between the mobile application, the PDA's database (SQL Server CE®) and eXtensible Markup Language (XML)-format files.

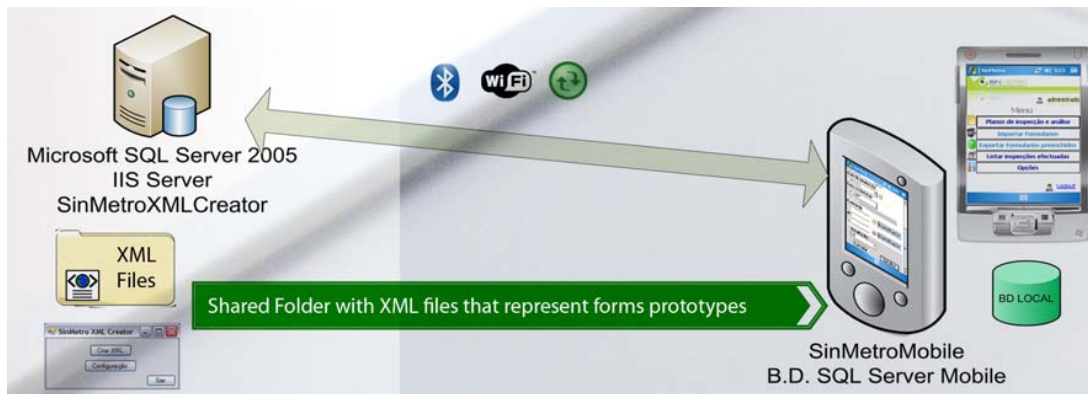


Figure 1 - Accept Mobile system architecture (top view).

Figure 2 presents the interconnection between the SQL Server CE® and the SQL Server®. The server is running Internet information Services, allowing connection to the OLE database provider through the SQL Server CE server agent. As expected, the client must authenticate itself to achieve database connectivity.

On the client side, the SQL Server CE stores the database, while the SQL Server Client Agent establishes communication with the SQL Server remote database engine, while SQL Server CE agent is responsible for local database connection.

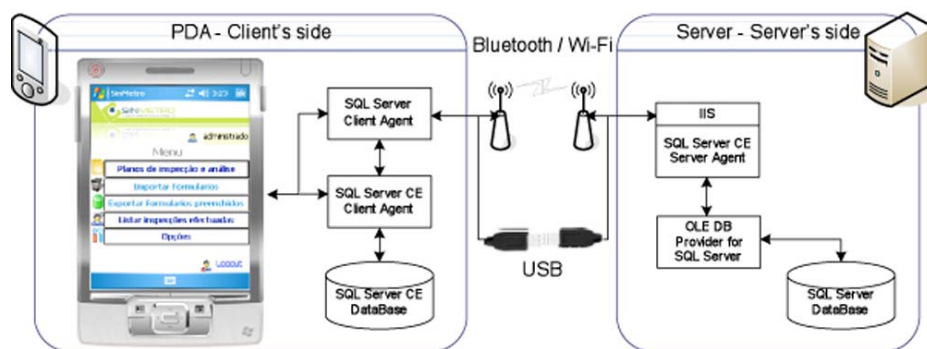


Figure 2 - Architecture of Accept Mobile system - interconnection between database engines.

4.1. XML Creator application

The use of XML files is required due to the high memory consumption that is noticed from the interaction between the application and the PDA's database on the process of Inspection-Forms generation. In order to maintain the application's performance we decided to use XML files, which would contain the Inspection-Forms templates. This way, the application would only access the PDA's database in order to complement the data analyzed on the XML files.

These XML files are generated and shared on the server's side, by an independent application that we created. The front-end of this application is show in Figure 2.

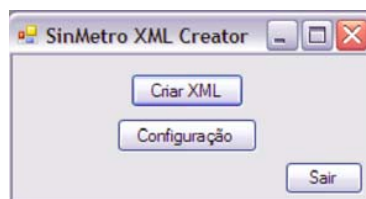


Figure 2 – XML Creator application frontend.

This application creates XML files by reading the server's database looking for Inspection Plans and locates them in a directory named like the Inspection Plans found. Several directories can be created depend on the number of Inspection Plans found. The XML files, whose name refers to an Analysis, are located in the corresponding Inspection Plan's directory. All this processing is issued by clicking "Criar XML" – Create XML.

Once the XML directories creation is performed on the server's side, we have to deploy them to the PDA. This occurs by sharing the directory (Apotheosis May 2005) that contains the XML files and copies it to the PDA. This operation is performed simultaneously with the synchronization of both, PDA and server's, databases.

As previously mentioned the Inspection-Forms are dynamically generated and there are no predefined Inspection-Forms. This process goes through a very delicate analysis of the XML files and the database, where the characteristic of the various fields to be created, are analyzed and stored on a complex system of object vectors (ArrayList (Wigley, 2002)). This way, the various fields on an Inspection-Form can be readily accessed.

4.2 Accept Mobile application

The Accept Mobile application gives the user total control over its features, based on a main menu with four main options, after some initial configuration. This startup configuration may only be performed once, since the application has the ability to store the last used good configuration. At this stage Accept Mobile has performed initial database synchronization and XML files retrieval from the shared folder. The application flow is depicted in Figure 3.



Figure 3 – Application flow of Accept Mobile – initial setup.

The first option, Planos de Inspeção – Inspection plans, creates the dynamic Inspection-Forms by selecting the Inspection Plan, the Analysis and the Reference (the product subject to analysis). The Reference can be created, or use one that has already been created previously. To create the final Inspection-Form the correspondent Analysis's XML file is analyzed. The application flow is shown on Figure 4.



Figure 4 - Option “Planos de Inspeção” operational workflow.

In the second option, Importar Formulários – Forms Import, depicted on Figure 5, Accept Mobile application features the ability to get new XML files from the server. We considered the fact that the user can force the search for new XML files on the server, without which this operation is only performed during the synchronization of databases. This operation is performed, as described in section 4, using a server’s shared directory.



Figure 5 - Option "Importar Formulários" operational workflow.

Option three “Exportar Formulários” – Forms Export, presented in Figure 6, presents the exportation of stored Inspections to the server’s database. As a result the execution of synchronization methods (Dove, 2005, TechNet, 2007, Wigley, 2002) is performed with this option. The development of this mechanism, in our case, became more difficult because of some reported troubleshooting RDA issues (Boske, Dec. 2001, Microsoft, 2008). This troubleshooting is related to identity columns and the fact that in the PDA’s side we work with empty tables to reduce the space occupied by the database. For that reason, the removal of the entire contents of the tables on the PDA’s side, all the IDs gets the value one. To solve this problem we have to control the ID manually in the PDA’s side. When the initial synchronization is performed the mobile application gets the last ID present in the server’s database and stores it in a single value table. That way, an insertion operation increases the ID value. So, when synchronization is performed all the line’s IDs are in a sequential order starting the last ID of the server’s database, and all works fine.



Figure 6 - Option "Exportar Formulários" screen.

Finally, the last option “Listar Formulários” – List Forms, presented in Figure 7, allows the user to edit Inspections Forms previously completed by him, that is, to edit the SQL Server CE database contents that himself created while using the device.

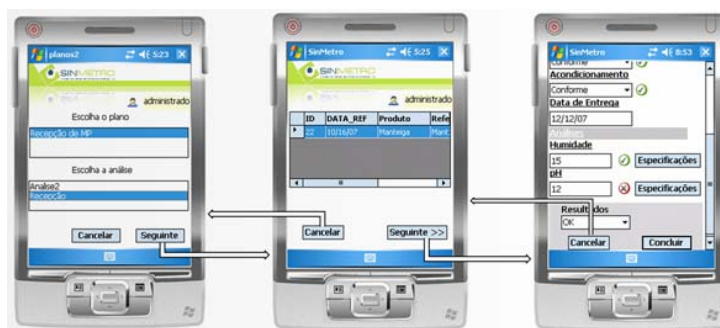


Figure 7 - Option "Listar Inspeções" operational workflow.

5. Conclusions

It is clear from our work that a huge gap is still present between programming for a personal computer and programming for a PDA. We can consider that the functionalities that are available for mobile computing are a “light” version of all the functionalities available for desktop programming (Wigley, 2002).

There is a constant work in progress regarding technologies that allow synchronization between databases on a desktop and a PDA on a disconnected environment. Although functional, there still are some limitations. The lack of strong error detection methods and weak solutions when facing a competition environment on some methods, are the most serious limitations.

With the evolution of the technologies we hope to see this and all the remaining limitations regarding this subject to be fixed in order to provide the user more powerful applications and simple to develop.

In terms of our application, SINMETRO requirements were fully met, as stated by one of the company’s shareholders. We prove that in some applications a PDA is more adequate than a portable computer, especially when the mobile user constantly roams around a building or a campus to collect data.

Acknowledgment

The authors of this work are pleased to acknowledge to SINMETRO - Sistemas de Inovação em Qualidade e Metrologia, Lda for the opportunity to develop this project. Special thanks to Eng. Gonçalo Martins from SINMETRO, for the availability and the support to work together with

this project. We would also like to thank the Computational Systems Laboratory of the Superior School of Technology for providing physical devices and space for this project.

6. Referências

- BOSKE, K. J. (Dec. 2001) Troubleshooting Microsoft SQL Server 2000 Windows CE Edition Connectivity Issues.
- DOVE, D. (2005) A Technical Comparison of Replication and Remote Data Access Features in SQL Server 2005 Mobile Edition 3.0. <http://technet.microsoft.com/en-us/library/ms345126.aspx>.
- LIM, W. M. (2005) Towards More Usable Mobile Application Development. *Mobile Technology, Applications and Systems, 2005 2nd International Conference on*.
- MICROSOFT, T. P. (2008) Selecting an Appropriate Primary Key for a Distributed Environment (Synchronization Services).
- SATYANARAYANAN, M. (2001) Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, 8, 10-17.
- SINMETRO (2008) Sinmetro Accept System. <http://www.sinmetro.pt>.
- TECHNET, M. (2007) Using Remote Data Access (RDA). Microsoft.
- WIGLEY, A., WHEELWRIGHT, STEPHEN (2002) *Microsoft .NET Compact Framework*, Microsoft Press.