Ubiquitous applications comparative: applications executed in server vs. applications in client

Javier Martínez, Javier Gómez, Alfonso López, José J. Martínez, Carmen Pages
Computer Sciences Department
University of Alcalá.
Ctra.Barcelona km 33.6 – 28871 Alcalá de Henares (Madrid), Spain
{jmartinez, alfonso.lopez, josej.martinez, carmina.pages}@uah.es
y jgomezr@soluziona.com

Abstract. Since the ubiquitous computation is born and different mobile devices begin to be able to execute applications, it is necessary deciding on if it is optimal the load and execution of the application on the mobile device or on the contrary, if it would be better the implementation of an application server that the different mobile devices realize to the requests and transactions by web. Certainly, there are increasingly companies that use mobile devices integrated in their applications. Because of it, it is necessary the development of this type of applications. The above mentioned decision is of vital importance for the functioning, maintenance and cost of the application. This document tries to present the advantages and disadvantages of each one of both options and an example about how both would be implemented for the management of records of a quality system.

1 Introduction

Nowadays, every time it is more necessary the use of applications that do not depend on the device in which they are implemented [1] [2]. This article tries to realize the comparative between applications that are thought for environments where a management system by quality is implanted; nevertheless, this type of applications and their analysis are valid in any system where they are implanted.

Nowadays, on the market we can find, basically, two types of applications, Those who use an url to accede, load forms in the devices in HTML, or any of its variations, fill completely these forms and sent these forms complete; and of another side, those applications that load in the devices, execute in the above mentioned devices and need some type of synchronization to send the information to the information process center [3].

In the model of processes orientated quality management based on the norm ISO 9001:2000 [4], among the points mentioned in this norm, it is possible to observe a paragraph that is “Quality records Control”; these records must be established and supported to provide evidence of the conformity with the requirements. The records must remain legible, easily identifiable and recoverable, especially for their posterior analysis. As good this norm contemplates, the information management is an intangible good that it is necessary to manage, for what the incorporation of mobile applications, which provide to the system of ubiquity, will help in the management of any organization.

2 Description of applications to comparing.

Both applications to comparing are based on the idea of managing information generated in the own application. The managed information treats on the processes and the associated records in a management system by quality.

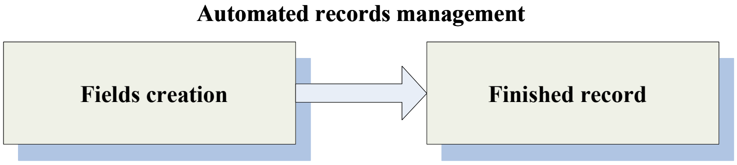

Fig. 1. Automated records management.

The initial point is the same for both types of applications, of a side we would have records/forms to refill and for another we would have users who must complete this forms in order that the information generated in the above mentioned quality records joins to the system and is evaluated for the posterior capture of decisions, the follow-up of the actions determined to do, as well as the refill of the same ones by the responsible users.

Therefore, to lead to end everything raised previously, basically two approaches exist: *development of applications that are executed in the server and implementation of stand-alone applications in devices*.

2.1 Application executed in server

This application type is formed, normally, by a set of pages, for example developed with JSP, which implement the business logic and interact by a JDBC connection with the information layer, where the qualities records are stored finally. This application type is composed basically of three modules:

Records administration module, in this module the above mentioned quality records will be created. With a certain profile, everything actions related to the forms management will be able to be realized, from its creation up to its elimination, with all the intermediate contemplated steps. In a quality system the documentation must be "alive" therefore in this module, every time any modification related to the system exists, it must join in order that the final user always has the last version of any form. This module has been implemented using JSP (Java Server Pages), for later, and using DOM (Document Object Model), to be able to generate the XML document corresponding to the above mentioned generated control sheet. This module is acceded by http that, after user's identification, will allow the creation or modification of the form, including the necessary validation controls.

Formatting records module. This module formats the above mentioned records to make the same ones compatible with any mobile device. Also, this module formats the above mentioned documents and generates the code corresponding to the device from where they are going to complete, for example I-HTML, C-HTML, WML. For it XSL will be use to transform the XML into all three formats.

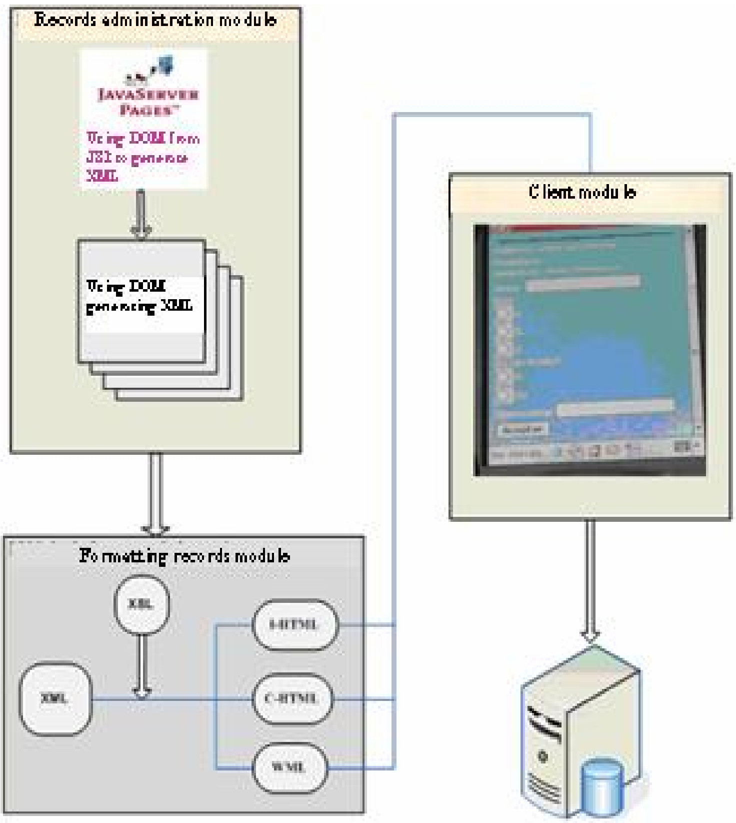
Records administration module
JAVASERVER PAGES
Using DOM from JSP to generate XML
Using DOM generating XML
Client module
Formatting records module
XSL
XML
I-HTML
C-HTML
WML

Fig. 2. General architecture of the application.

Client module, where the records are refilled and are sent completed to the central process unit for its posterior analysis. In this part of the application, the users insert information in the data base using the browsers of theirs devices. It is a responsibility of the application to detect what type of device is requesting the page to offer the control sheets in the suitable format, so much if it is HTML (for PC), as I-HTML (telephones I-MODE), as WML (telephones WAP), etc. For do it, it is necessary capture the user-agent head record [5], which returns to us information about the client's browser. So we will associate with every device the corresponding page and we will use the adapted format for every device. Another mission of this module is that the responsible user of realizing any work applying the corresponding procedure and its associated control sheet , after his identification in the system, will be able to use only those control sheets for which, in the creation module, he is authorized. For what every user, depending on his profile, will be able to use different control sheets. At the end of the process, all the generated and completed information will be load in a data base.

Advantages of these applications

It is not necessary application unload: when the mobile device wants to use the application, it does not need to unload it, only it is necessary that it knows the URL server where the application resides. This characteristic causes that the costs of unload and development the application in the device there save.

It is not necessary to notify possible modifications to the user: is not necessary that the process central unit notifies this event to the users, because they do not return to unload the application to the mobile devices. For this motive the server applications maintenance is simpler, is transparent for the user and has a little cost.

More experience in these developments: to develop an application in a server frame who can be attacked by the http protocol, it is not necessary to ask for help of specialists in languages of programming for mobile devices like J2ME, DoJa, Symbian, Visual Basic Embedded, etc. Only we must use an active server language as JSP, ASP, PHP, etc. A major number of professionals with experience exit in these languages. But nowadays this is not very importantly due to the fact that tools exist - frameworks - that facilitate very much the work in any environment.

Information refill without cost: in the applications executed in server, when the user refills with information the form that was unload in the client device, it does not generate any cost since the user completed the form in local way and the information is not sent until the user decide.

Independence of modules: the most important thing is the independence of the different modules management. This type of mobile applications is independent from the final device [6].

Independence of the devices in which it is in use: when the XML documents are generated and formatted for a concrete device, in case of incorporation of new formats, the XML documents will be the same. For what it will be necessary only to develop the sheets of styles adapted for the above mentioned devices.

Disadvantages of these applications

Interchange of information: if the information transmission between the device and the server is raised, the functioning cost goes off. Therefore this option is the suitable one when the frequency of the information updates is low, that is, when the device does not need to send information to the application constantly.

Indispensable coverage: when we implement this option we must consider that we need a connection like GPRS, UMTS, etc. Beside, we must implement mechanisms in the server not to destroy the data base coherence when a user remains without coverage during the information transference.

A lot of shared information: this option will implement protections mechanisms of information, as for example critical regions, in order that the actions of the different users do not hit some with others.

2.2 Stand-alone applications in devices.

The architecture that is analyzed in this point will have the following layers:

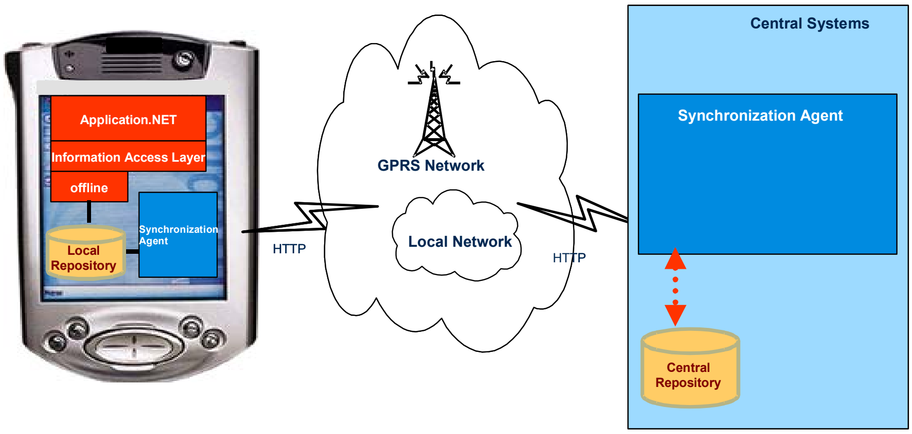

Fig. 3. Architecture of the application executed in a mobile device.

Stand-alone application, that will execute and run autonomous in the device in which it was installed (pda, mobile telephone, table pc, personal computer, etc.). The stand-alone application will have a local data base where there will reside and stay the information with which the user works. This application will be able to be from a monolithic development to an application in several layers (presentation, business, resources, etc.)

Synchronization agent. There are two applications types: developed to measurement applications or commercial applications. They manage the interchange of information between the stand-alone devices and the central systems. The information that is transmitted will be information or versions of the application.

Central system of information (or server). This module will be the central data base where the final information will be stored.

The indicated structure is the basic one and it can be extended as necessary, depending on the new requirements, the incorporation of a load-balancer in the server, the need of safeties copies of the data base, etc. Equally the information deposited in the central repository might be acceded and exploited by resident applications in the own server or by other external systems.

Advantages of these architectures

The principal advantage is the independence between the application and the communications across *the network*, this will allow:

The final user has a better vision of the application. The information interchange with the central systems does not exist. It improves the times of response of the applications for the final users.

Falls of network. The application and the information always are available. There is no dependence of the availability of network not of the saturation of the same one.

Cost reduction of transmission. If the network by which the information is transmitted causes costs of consumption then these would be reduced in these architectures.

Disadvantage of these architectures

Duplicity of information. As the information is duplicated in the final clients, every device user can have its own version. Even, these versions can be different of the actual one (if the initial information has been modified in the server). Because this, it is necessary to reinforce the update processes from the final repository.

It is necessary processes of synchronization management. These processes might go from a simple development with a trivial politics of weight (between client and server) to a complex application with decisions-maker on conflicts in every possible case. The commercial existing products have a very direct relation between their power and the complexity in the configuration and administration of the tool.

Update of versions in clients. The problematic is not this in the load of new software versions in the clients, because exist tools that allow and facilitate this, but in two topics:

- The update of the versions not always can be realized of automatic way in all the clients. In many cases, the clients must initiate the process of information update using the synchronization processes.
- Data bases modifications. Any change in the structure of the data base has to be processed of a particular way. It is advisable that all the changes are "incremental" (new fields and tables) being avoided as possible the field modification or the field elimination in the existing tables.

3 Conclusions

It is clear that every type of application has its advantage and disadvantages. Depending on the use, an application type or other one will be more advisable.

Choosing an stand-alone architecture, leaving out economic questions relating to support the communication costs, is advisable when the information of the application is very static (do not changes) and when any modification or insertion of information from the clients does not have any type of effect on other systems.

Nevertheless, the advantage of applications, in which the forms was unload by http, is the low need of resources to incorporate the application in the receiver device, besides other advantage is the fact that at the moment of unloading any new form, it is updated according to the module of creation of forms. Nevertheless, in the stand-alone applications, when a modification is realized in the application, it is necessary to synchronize and update the application in the different devices. Against that, this type of applications can control neither the type of inserted information nor the information loss when the refill of the form in use is stopped before ending. Until the information is not sent and the server does not receive it correctly, we do not have guarantee of which the above mentioned information has been inserted.

The principal recommendation is to evaluate first the requirements of information. It is necessary to understand how the processes of the application are going to work, in order to verify if it is better the information " just in time" or, on the other hand, if it is possible to obtain the information after a time.

References

1. Honkala, M., Cesar, P., Vuorimaa, P. A device independent XML user agent for multimedia terminal, IEEE 6th International Symposium on Multimedia Software Engineering, Miami, FL, December 2004.
2. Grimson, R. Device Independence Principles, W3C Working Group Note. http://www.w3.org/TR/2003/NOTE-di-princ-20030901/. 2003.
3. ORACLE Inc., Oracle Database Lite Business White Paper. In: http://www.oracle.com/technology/products/lite/BWP_Lite_10gR2.pdf.

4. ISO 9001:2000, Quality management systems-Requirements. International Standards Organization, Geneva (Switzerland), 2000.
5. HTTP- Hypertext Transfer Protocol, World Wide Web Consortium. In: http://www.w3.org/Protocols.
6. Lie H.W., Saarela J. Multipurpose Web publishing using HTML, XML, and CSS, Communications of the ACM, vol.42, no.10, pp.95-101, October 1999.