

Designing Mobile Applications

Abstract

When moving from client/server to Web based applications, developers had to rethink they way the designed and built applications. The same paradigm shift has to occur now as we move from web-based applications to mobile web applications. Building mobile web applications demands that we overcome challenges related to device, network and technology limitations.

Device issues include screen size, memory, lack of keyboard for data entry, general usability (scrolling, navigation, etc.), numerous device types and no standards means of determining individual device characteristics. Some device issues will resolve themselves as manufacturers build new devices with different form factors and larger screens. Voice-based computing, long considered the “killer application” for mobile computing is beginning to come into its own. But this brings with it its own challenges. So, what can developers do today to overcome some of the device challenges that exist?

First, they must realize that building device based applications is different than building Web applications. The applications will be used in mobile environments and so must deliver summary information in a concise manner. The applications must be smaller. They must be designed to minimize data entry (i.e. small number of data fields). The business logic must perform discrete functions. Small applications will increase performance and allow for the scalability required for consumer based applications and accommodate page size limitations. The ability to isolate discrete “chunks” of functionality will accommodate reuse across device types thereby eliminating the need to rebuild the same functionality for different devices.

Network issues such as inconsistent connectivity and bandwidth limitations lead to session management, message queuing, and transaction management issues that have no generic solution.

Session management issues relate to the inability for devices to store cookie information. There is no generic mechanism for server-side applications to retain information about the user or the device that a user is currently using. Many devices don't even have unique identification parameter that is accessible by a server-side application. This demands that a server-side application dynamically create device id's and then sync-up those device id's to cookie information for each user session. In other words a generic session management capability is required. The time length of a session must be variable to accommodate the likelihood that mobile devices will not maintain connectivity (i.e. if user goes under a bridge and loses service provider connectivity.) This type of session management capability can be accommodated with message extension time coding or connection polling capability.

Technology limitations include multiple browser standards including HTML, WML, cHTML or other “ML” derivatives. Also, while there is a way to determine device type via the HTTP request header, the variable name that manufacturers use in the header is different for different devices. This makes it difficult for developers to customize their applications to take advantage of device specific display characteristics.

Part of the answer to this challenge is generating application output (i.e. content) as XML and using XSL style sheets for formatting. In other words, when an HTTP request is generated, the application must interrogate the HTTP header, determine the device type and then forward the request to the business logic. The business logic must then render the response content as XML. An XSL style sheet must then be applied to the XML content to format it appropriately for display on the requesting device.

To enable the re-use of applications functionality for the widest array of possible devices, developers should use the Model View Controller (MVC). Architecture. The MVC architecture was proposed to develop GUI programs in Smalltalk. It is now widely accepted in OO technology. In the MVC architecture, an application is divided into three parts, the model, the view and the controller. The model is the logic that describes the possessing that occurs in the application. The view is how information is presented to a user – the interface. The controller provides a way to change the parameters of the model and the view. The key advantage in building applications where the presentation layer is distinct from the business logic layer such as the MVC architecture is reusability. The advantage is the time saved in not having to reproduce the same functionality over and over for each application.

As the IT development world shifts from building web applications to building mobile web applications, new challenges related to devices, networks and technology must be faced and overcome. The solutions to these new challenges are far from complete, but as our knowledge and experience grows the productivity gains achievable via mobile computing will be realized and the computing landscape will change forever.