Device Management specifications

The Open Mobile Alliance (OMA) specified a platform-independent device management protocol called OMA Device Management. The specification meets the common definitions of an open standard, meaning the specification is freely available and implementable. It is supported by several mobile devices, such as PDAs and mobile phones.

Over the air

Over-the-Air (OTA) capabilities are considered a main component of mobile network operator and enterprise-grade Mobile Device Management software. These capabilities include the ability to remotely configure a single mobile device, an entire fleet of mobile devices or any IT-defined set of mobile devices; send software and OS updates; remotely lock and wipe a device, which protects the data stored on the device when it is lost or stolen; and remote troubleshooting. OTA commands are sent as a binary SMS message. Binary SMS is a message including binary data.[9]

Mobile Device Management software enables corporate IT departments to proactively and systematically manage the growing universe of mobile devices used across the enterprise; subsequently, over-the-air capabilities are in high demand. Enterprises using OTA SMS as part of their MDM infrastructure demand high quality in the sending of OTA messages.

Adaptation

Content adaptation is the action of transforming content to adapt to device capabilities. Content adaptation is usually related to mobile devices that require special handling because of their limited computational power, small screen size and constrained keyboard functionality.

Content adaptation could roughly be divided to two fields:

1. Media content adaptation that adapts media files
2. Browsing content adaptation that adapts Web site to mobile devices.

Advances in the capabilities of small, mobile devices such as mobile phones (cell phones) and Personal Digital Assistants has led to an explosion in the number of types of device that can now access the Web. Some commentators refer to the Web that can be accessed from mobile devices as the Mobile Web.

The sheer number and variety of Web-enabled devices poses significant challenges for authors of Web sites who want to support access from mobile devices. The W3C Device Independence Working Group described many of the issues in its report Authoring Challenges for Device Independence.

Content adaptation is one approach to a solution. Rather than requiring authors to create pages explicitly for each type of device that might request them, content adaptation transforms an author's materials automatically.

For example, content might be converted from a device-independent markup language, such as XDIME, an implementation of the W3C's DIAL specification, into a form suitable for the

device, such as XHTML Basic, C-HTML or WML. Similarly a suitable device-specific CSS style sheet or a set of in-line styles might be generated from abstract style definitions. Likewise a device specific layout might be generated from abstract layout definitions.

Once created, the device-specific materials form the response returned to the device from which the request was made. Content adaptation requires a processor that performs the selection, modification and generation of materials to form the device-specific result. IBM's Web sphere Everyplace Mobile Portal (WEMP), BEA Systems' WebLogic Mobility Server, Morfeo's MyMobileWeb and Apache Cocoon are examples of such processors.

Wurfl and WALL are popular Open Source tools for content adaptation. WURFL is an XML-based Device Description Repository with APIs to access the data in Java and PHP (and other popular programming languages). WALL (Wireless Abstraction Library) lets a developer author mobile pages that look like plain HTML, but converts them to WML, C-HTML and XHTML Mobile Profile depending on the capabilities of the device from which the HTTP request originates.

Mobile Computing platforms such as mobile phones, PDAs or wearable computers operate in a much more volatile and limited environment than their stationary counterparts. Such platforms are inherently resource poor and subject to highly changeable resource availability. Applications for Mobile Computing require adaptation for best performance under such variable conditions, to make best use of available resources without assuming the minimum set Current systems are able to notify an application to adapt but fail to say how.

The application author must provide the actual adaptation mechanism. The adaptation mechanism is pervasive through application and system layers providing tight integration of adaptation both vertically (through an application) and horizontally (between applications). Mobile computing situates applications in an environment rich in resources and services but poor in resource availability or predictability. Applications that wish to make best use of available resources under such variable conditions without assuming the minimum set must adapt.

Typically, adaptation for mobile applications involves trade-offs: varying an application’s quality of operation or even locality of operation to fit the current resource profile. The most common resource triggering adaptation in current applications is communications bandwidth as it is typically the most limiting and unpredictable resource in the system, particularly in the case of wireless communications systems. CPU capacity, memory availability and battery power are other adaptation criteria.

Diversity: Mobile devices will be called upon to execute a large range of applications concurrently over a large range of conditions. Current devices assume that the user only requires to perform one operation at any one time and so mobile phones and PDAs often run simple operating systems that only run and present a single application at a time. Future applications may be more passive, pervasive and hence will require that multiple such applications be run concurrently on a single device or single resource set.

Adaptation: To perform such operation efficiently, adaptation is required. Connected mobile devices have access to resources beyond the device itself in the form of remote data and remote

services. An application that uses such remote services is a *distributed* mobile application. Using remote resources can greatly enhance the capabilities of a mobile application but at the same time, the limitations of the mobile infrastructure means that the application must be able to cope with poor connectivity (up to and including disconnected operation), flagging resources (battery or CPU), and insecurity (mobile devices are much more susceptible to theft or security violations than their stationary counterparts). In particular, locality of data, locality of processing and locality of control must be resolved consistently. Intermittent connectivity means that if data is to be accessible at all times, it must reside on the device itself. On the other hand, the mobile device is small, limited and vulnerable only small data sets can reside on the device. The device may only manipulate restricted data sets and employ external facilities for more data. Additionally, maintaining data access and consistency (in the case of shared data) through disconnection become major issues.

Data Locality: Essential data must be held locally. Extended data sets may be held remotely. As for data, provision must be made for processing within the isolated case without restricting operation to exclude the advantages of connectivity to external facilities. However, some applications rely on a remote data source (eg: media streaming) in which case operation in a disconnected environment is meaningless.

Process Locality: A mobile device must be self-sufficient for the minimum of processing. Additionally, for adaptive distributed systems, care must be taken as to where the control of such systems originates. For mobile systems, for the reason given above for both data and processing, control must reside with the mobile device for operation through disconnection:

Control Locality: The mobile device must be in control of its resource usage and subsequent configuration.

Adaptation Models

An adaptation model describes the structure and policies of an adaptive system. It defines the system level at which adaptation occurs and also at which level adaptation decisions are made. This section describes three broad divisions of possible models and argues that only a fully integrated application aware model can effectively provide adaptation services to a mobile system. Transparent adaptation models perform adaptation at operating system or session levels in a manner transparent to, and concealed from, the applications. Operation of system services is modified to adapt to resource availability in a manner that is transparent to applications using that service. An application-transparent model is beneficial in that existing applications may be run unmodified and new applications need not deal with adaptation issues.

Central control of resources and adaptation avoids competition between applications and aids efficiency in resource use. However, such an adaptation model must operate without the application specific information that can help it make decisions concerning the best adaptation strategy. Working with unknown data types or the exact requirements of the applications, the adaptation mechanism must second-guess the nature of the application from its observable characteristics. This is the fundamental limitation of application-transparent adaptation: it is unable to efficiently handle situations outside its very limited initial remit. In contrast to the centralized approach of the application transparent model, application-specific adaptation places the task of adaptation solely with the application with no explicit system support. While this allows each application to perform adaptation exactly to suit its needs, there is no cooperation or coordination between applications.