



Provided by the author(s) and NUI Galway in accordance with publisher policies. Please cite the published version when available.

Title	Portable user interfaces for remote access to embedded home systems and home networks
Author(s)	Corcoran, Peter
Publication Date	1998-06-04
Publication Information	Corcoran, P.M. and Papal, F. and Zoldi, A. (1998) Portable User Interfaces For Remote Access To Embedded Home Systems And Home Networks Consumer Electronics, 1998. ICCE. 1998 Digest of Technical Papers. International Conference on
Publisher	IEEE
Link to publisher's version	http://dx.doi.org/10.1109/ICCE.1998.678302
Item record	0-7803-4357-3; http://hdl.handle.net/10379/4034

Downloaded 2019-04-24T22:26:47Z

Some rights reserved. For more information, please see the item record link above.



TUPM 12.1

Portable User Interfaces for Remote Access to Embedded Home Systems and Home Networks

Peter M. Corcoran, Ferenc Papai and Arpad Zoldi.

Dept. of Electronic Engineering, University College, Galway

Abstract - *The design and implementation of portable, lightweight user-interfaces to provide access to remote embedded home-systems and home-networks is described. This paper is focussed on emerging standards in the consumer electronics field, including embedded-Java, the handheld device markup language (HDML) and remote frame buffer (RFB) technology as implemented in the public-domain virtual network computer (VNC.) software.*

A key goal of the present paper is to provide an objective technical assessment of the applicability of each of these technologies to the development of practical consumer appliances and handheld terminals providing enhanced user-access to the home environment.

1. Introduction

As we approach the 21st century it is clear that a new generation of consumer electronics products is emerging. These new products are increasingly network aware and offer new dimensions in user access. Recent industry initiatives such as the adoption of the *hand-held device markup language* (HDML) by several key industry sectors, the announcement of *embedded-Java* and the *remote frame buffer* (RFB) suggest that many new consumer electronics products will support remote access over either a home network, or even over a wide-area-network (WAN) such as the Internet.

As users adopt these new technologies they will come to expect that new appliances may be controlled remotely from multiple access points within the home environment. For example, an end-user might expect to be able to access the home stereo from (i) a Web-browser on his home-PC, or from (ii) a Java-phone terminal, or using (iii) his TV set and its remote control or from (iv) a personal organizer using a wireless link to the home automation network. Furthermore, a consumer will expect to achieve this access using an almost identical interface from each of these varied access point.

In this paper we examine how suitable, lightweight and portable user interfaces can be designed using some of today's key emerging technology standards. The implementation of some example applications is also

discussed and the practical problems involved in handling interfaces across a range of display and access point technologies is discussed.

2. User-Interface Technologies

In creating the user interfaces to access a Home Network two main requirements are defined. First, the portability of the user interfaces, allowing access to the home network from a wide range of hardware platforms and operating systems. The other principle requirement for the interfaces is that they should be as thin as possible to facilitate dynamic uploading and operation on hand-held computers and resource constrained devices. In this paper we focus on three key approaches to the problems outlined above: embedded Java, HDML and remote access to user interfaces using the RFB protocol.

Embedded Java is an application development environment for creating software for embedded devices and resource constrained environments. Similar to Java, embedded Java consists of core and standard extension APIs, more precisely Embedded Java includes a feature level subset of Java, therefore Embedded Java applications are upward compatible with Java. If Embedded Java is used, the interface is uploaded to the remote access client as executable Java byte-code and executed in a *Java virtual machine* (JVM) on the client. However this can be resource-heavy, particularly at the client end as a full implementation of the JVM is required and this, typically, requires 16M or more of memory.

The hand-held device markup language (HDML) provides an efficient markup language for wireless and other hand-held and resource-constrained devices. Its focus goes beyond presentation and layout, providing an explicit navigation model which does not rely upon the visual context required of HTML. HDML is an efficient mean to provide content via the WWW infrastructure to hand-held devices such as cellular phones, pagers and wireless PDA's. In this second approach a HDML interface is loaded by a HDML interpreter - somewhat analogous to a Web browser - and events are generated in the remote server

application using the well-known common gateway interface (CGI).

The remote frame buffer (RFB) protocol is a simple protocol for remote access to graphical user interfaces. This protocol is truly a "thin-client" protocol: it has been designed to make very few requirements of the client. In this way, clients can run on the widest range of hardware, and the implementation of a client is kept as simple as possible. This protocol also makes the client stateless, and because of this stateless nature of the client the connection can be closed at any time by either of the sides without any system-level consequences. Because of the design of the RFB protocol writing a viewer for a client is a very simple task, and RFB clients may function optimally on small hand-held appliances.

3. Prototype Hardware Infrastructure

To simplify some of the design and implementation issues we assume that access to most appliances will be through a central home-gateway which acts as an interface-server to each remote access point. Thus the actual control application runs on a home-gateway server and is common across all access points. This is illustrated in Fig 1 below.

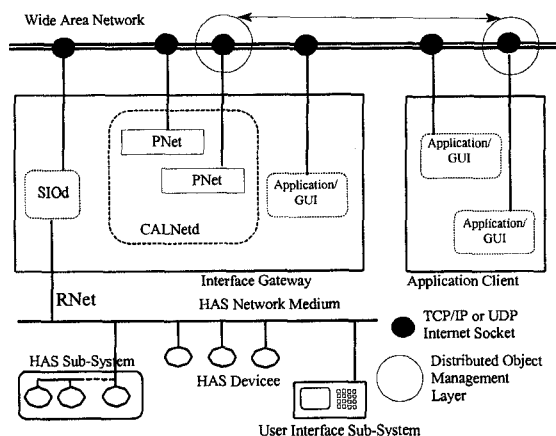


Fig 1: Diagrammatic representation of the relationships between system software components, the Interface Gateway and a single Application Client.

In this scenario only the interface must be uploaded to the remote access point, or application client. These interface components should thus be lightweight and permits separate, customized interfaces to be uploaded to access points with different hardware platforms and/or operating systems.

The Home Gateway is the central component of the system hardware infrastructure. In our prototype system a standard PC represents the home gateway; in a practical application we expect that a set-top box or

embedded gateway would replace this. The PC is connected to a typical power-line home network and in this way can provide access to networked consumer appliances. Connections to other home networks, including USB and IEEE 1394 devices can readily be achieved by adding adapter cards to the standard PC. A TCP/IP internet connection is also implemented and allows the prototype system to download interface and device-driver modules from the TCP/IP network. This Home Gateway can also communicate with hand-held computers and PDAs using serial-line connections and, ideally, wireless network connections.

4. System Software Components

The system software components run mainly on the Home Gateway computer, except the client interfaces which is uploaded to an application client. The most important component of the system is a main server module. This module can communicate with both a home network and an application client.

Communication with devices on a typical home network is realized using a server sub-module, or translation layer. In our implementation this module is able to send, receive and interpret packets from devices connected to a power-line network and sends back the results to the main server. Users communicate with the server through the specific interfaces and using a TCP/IP connection.

When a new device is connected to the Home Network the server detects it and downloads the corresponding user interface package for that device from the Internet. On a request, only the user interface will be uploaded to the client access point.

5. Conclusions

In this paper we present different ways for remote access to Embedded Home Systems and Networks. Using lightweight user interfaces on the client's side which require just a thin remote client, allows us to use hand-held devices as well for remote access. The provision of such remote access will greatly enhance the easy of access Embedded Home Systems and Networks for users.

To emphasize the practicality of implementing portable interfaces with today's technology we have utilized a varied range of access point client technologies. These have included conventional Web browsers, OS platforms such as Linux, Inferno, and Windows-CE, several hand-held organizers and even a GSM telephone client.