

## The Universal Remote Console: A Universal Access Bus for Pervasive Computing

Bill LaPlant, Shari Trewin, Gottfried Zimmermann, and Gregg Vanderheiden

IEEE *Pervasive Computing*'s editor in chief, M. Satyanarayanan, has been quoted as defining pervasive computing as "the creation of environments saturated with computing and wireless communication, yet gracefully integrated with human users."<sup>1</sup> Given the ever-increasing complexity of these ubiquitous environments, and the number of devices and services involved, we need a new standard to enable this graceful integration. The emerging V2 standards are arriving just in time to facilitate the fulfillment of the pervasive computing promise.

### THE STANDARDS

V2, a technical committee of the International Committee for Information Technology Standards ([www.incits.org](http://www.incits.org)), is developing standards for Infor-

mation Technology Access Interfaces. As part of this endeavor, V2 ([www.v2access.org](http://www.v2access.org)) is about to issue the *Universal Remote Console* architecture, a set of draft standards for public review and comment.

The five standards specify communications between a *Target*—a device or service the user wishes to access—and a Universal Remote Console—software that is typically hosted on a user's personal device. The standards specify mechanisms for the discovery, selection, configuration, and operation of user interfaces and options:

First, the Universal Remote Console specification provides a framework of components that combine to enable remote user interfaces and remote control of network-accessible Targets through a URC. This document describes the URC architecture.

Second, the *Target Properties Sheet* specification defines an XML document describing a URC-conformant Target so that a URC can discover it. A TPS provides the information a URC needs to connect to one of the Target's Portals to start a control session. A Target has exactly one TPS.

Third, the *User Interface Socket Description* specification defines XML language for User Interface Socket Descriptions. A User Interface Socket is an abstract concept that describes a Target's functionality and state in a machine-interpretable manner. The Socket exposes a Target's relevant information so that

a user can perceive its state and operate it.

Fourth, the *Presentation Template* specification defines XML language for Presentation Templates, which provide hints for building a usable, consistent user interface for a Target. The hints are abstract and are intended to apply to any delivery context.

Finally, the *Resource Descriptions* specification defines the syntax for describing *Resources* relevant to a Target's user interface. A Resource is an identifiable object that's used as an atomic entity in the construction of a concrete user interface. Resources can include text elements of a user interface such as labels, help text, keyboard shortcuts (access keys), and associated words (keywords). They might also include nontext elements such as icons, sounds, or videos. A Resource refers to a specific element in a Socket, a Presentation Template, or a TPS.

### TARGETS AND URCs

The URC standard will facilitate the development and deployment of a wide variety of devices (from different manufacturers) that can act as URCs for an equally varied range of Targets. In other words, the standards will let users control any number of electronic and information technology devices in their environment.

Targets can range from things as simple as light switches and thermostats, to more complex items such as audio-

### ABBREVIATIONS

RUN	Resource-URC Network
TPS	Target Properties Sheet
TUN	Target-URC Network
UIID	User Interface Implementation Description
UIML	User Interface Markup Language
URC	Universal Remote Console
WAI	Web Accessibility Initiative
XML	Extensible Markup Language

visual equipment, home appliances, electronics in cars or other constrained or specialized environments, Web-based services, and any other devices or services that can be controlled electronically (or via information and communication technology—ICT).

A Target can be in the same location as the individual who desires to control it through the URC. Or, it can be at any distance from the URC or user, as long as some type of network connection exists between the URC and the Target. This is possible because a URC provides the user with all the necessary controls as well as the prompts and other information that the Target displays.

Common devices such as personal-computing and information technology devices (for example, laptops and PDAs) or telecommunications/WAP (wireless application protocol) devices (for example, cell phones) can act as URCs. URCs can also be functions implemented in assistive-technology devices, or they can be devices that are specially built to function as URCs. They can also be devices that function primarily as remote consoles for a particular product family (for example, a remote console designed to control components of an integrated home audiovisual system) but that also can control any other device that's URC compatible. They behave similarly to universal remote controls today, except that

- They have much greater function and scope
- They synchronize with the Target in both directions (that is, they can display the Target's current status)
- They don't need to be programmed by the user (they automatically discover devices that are controllable in a user's vicinity, discover the Targets' abstracted user interface, and present it in the way the user prefers)
- Depending on the networking technology used, they can function out of sight of the product they're controlling

URCs can provide output interfaces that are all visual, all tactile, or all verbal (or any combination thereof), because the URC standard specifies a Target user interface's content independently from the form in which it's presented. Similarly and for the same reason, the control interfaces can be by voice, keyboard, mouse, or any other available technology.

So, you could have a URC that gives you speech access to any URC-compatible Target, even if that Target doesn't have voice recognition or voice control functionality. You might, therefore, be able to say to your URC, "Record Channel 12 and show me *Law and Order*." Or you could be lying in bed and say, "Set the alarm to 6:30 a.m., turn the coffee on at 6:00 a.m., and turn on the home security system." Or, if your spouse is already asleep, you could pick up your PDA or any other compatible URC and accomplish these same tasks silently either by calling up control panels or by issuing the instructions in writing. (The URC standard doesn't provide the natural-language control but provides the information and control necessary for control by a natural-language-processing URC.)

## HOW THE STANDARDS WORK TOGETHER

The URC standard specifies communications between a Target that a user wishes to access and operate and that Target's URC. The URC is software that's typically hosted on the user's physical device, but a distributed approach is also possible.

Communication between the Target and URC take place over the *Target-URC Network* (TUN). The URC standard doesn't specify a network protocol, so a Target or URC can support any number of appropriate connection protocols. These protocols provide discovery of Targets and establish and maintain sessions with Targets. Targets and URCs access the TUN through TUN Links. As we mentioned before, targets support discovery by providing

essential information in a TPS.

Interaction between a Target and a URC consists of two phases. The *discovery phase* initializes the URC to locate and identify all available Targets. During the optional *control phase*, a Target and a URC initiate, maintain, and terminate a *control session*. A control session is a connection between the URC and a Target that lets the URC control a functional unit of the Target via the Portal's Socket.

Each Target provides documents that describe its interface. The URC uses these documents to find or generate an appropriate user interface, given the Target's functionality, the nature of the URC device, and the user's interaction preferences.

One type of document is a *User Interface Implementation Description*. A UIID describes a particular Target's interface, which may take an abstract or concrete form and be very general or specific to a given URC device class, user profile, and Target task. The UIIDs provide a mechanism by which a manufacturer can provide tuned interfaces for its Targets. These interfaces are predefined and optimized to work on particular URCs—for example, Pocket PCs or Palm-based PDAs.

The URC standard defines one, very general, form of UIID—the Presentation Template, which doesn't assume any particular class of URC.

Another type of document is a *Resource Sheet*, which contains descriptions of Resources pertaining to a Target. A URC can take advantage of Resources in the Target (*Target Resources*) or outside it (*Supplemental Resources*). The URC can use Supplemental Resources to replace, supplement, or help interpret or translate any Resources in a Target.

In general, Resources for building user interfaces can be obtained from the Target, stored on the URC, or gathered from the Internet. A URC can use any form of networking to access Resources via a *Resource-URC Network* (RUN). The URC standard doesn't specify the mechanism by which a URC accesses Supplemental Resources.

## STANDARDS, TOOLS &amp; BEST PRACTICES

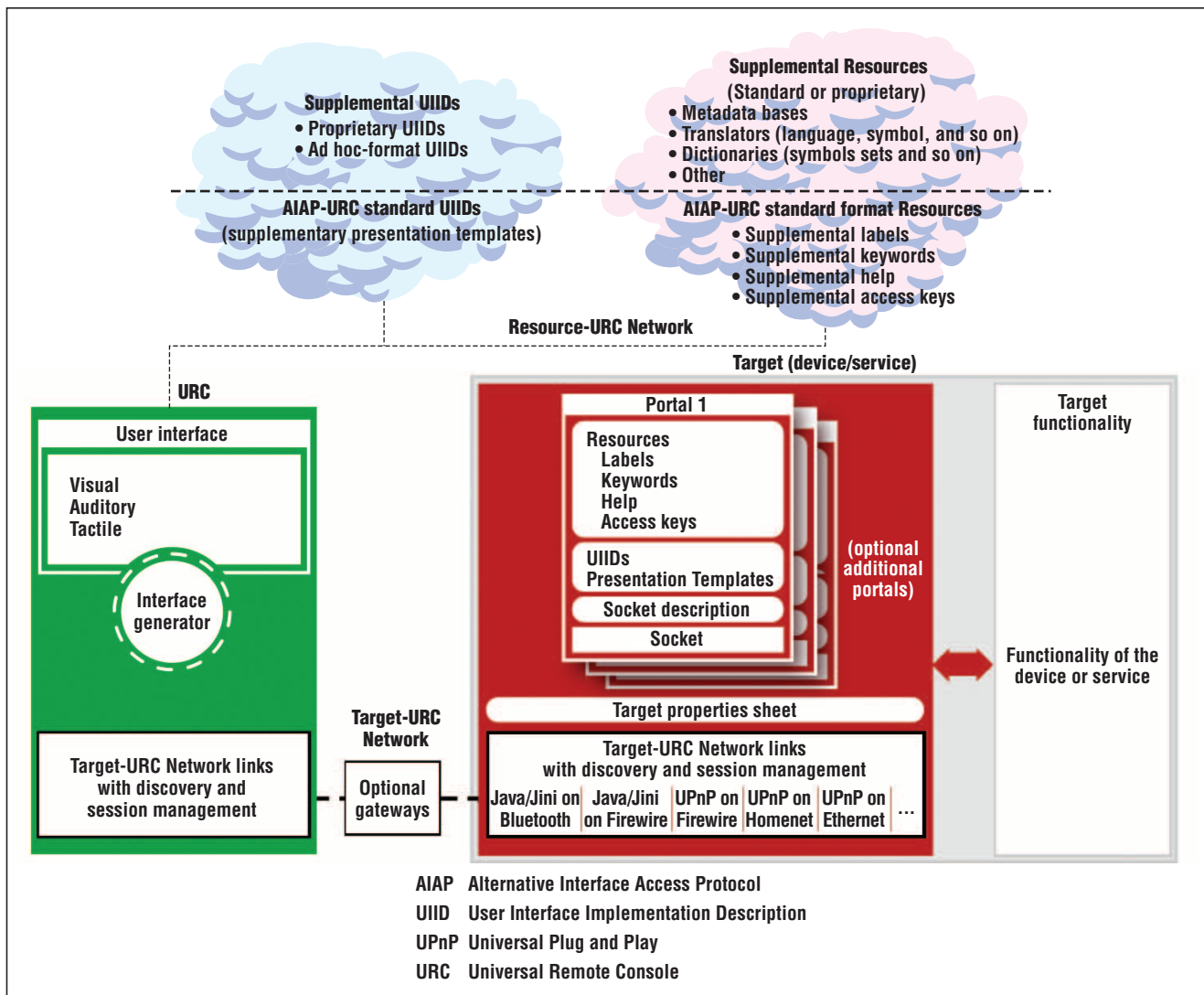


Figure 1. The Universal Remote Console specification's structure and components.

## ARCHITECTURE

Figure 1 illustrates the URC standard's main components within the broader context of an application. It illustrates both elements for which specified standards exist and elements that can use these or other standards. This figure is a conceptual map; it requires implementing the illustrated functionality rather than prescribing a particular architecture. Various documents specify a Target's functionality and user interface. In the figure, these documents appear as rectangles with rounded corners.

As the figure shows, you can think of the URC architecture as comprising

four major components and two networks. The components are

- The URC
- The Target
- Supplemental UIIDs
- Supplemental Resources

The networks are the TUN (which is required) and the RUN.

## CONFORMITY

For these standards to work, the devices and services that different organizations develop must be able to interoperate. So, the document for each

standard contains the requirements for conformity to each of that standard's components. These requirements can be used to test conformance with the standards.

Conformance will be determined by industry interoperability testing methods. The component (URC or Target) developer or manufacturer will bear the costs for this testing.

## MEETING A GROWING MARKET'S NEEDS

A burgeoning market already exists for personalization of content and appearance on the Web, with regard to

small handheld devices and for commercial sales purposes. Companies such as IBM, ATG, and Vignette have products and services for this explicit purpose. This market should grow considerably in the next three years.

Another rapidly emerging segment of technology is pervasive computing. Of particular note are the numerous offerings in

- Unified messaging or mobile computing, providing a central Internet access point for users' messaging needs (eFax and Hotmail)
- Net-based information stores for people's core documents and pictures (Freespace and Apple iTools)
- Transcoding and reformatting services for various devices (Everypath, YahooMobile, and IBM)
- Home automation & multi-access-

point management (Sony, Echelon, Microsoft, and IBM)

- Personalization of content and related marketing data (Vignette and NetPerceptions)

This follows the burgeoning of wireless technology from companies such as Qualcomm, Nokia, Ericsson, and Motorola, as well as the efforts of groups that promote wireless-device applications. Such groups include the Wireless Application Protocol Forum (which merged with the Open Mobile Architecture initiative to form the Open Mobile Alliance, [www.openmobilealliance.org](http://www.openmobilealliance.org)) and the Salutation Consortium ([www.salutation.org](http://www.salutation.org)). Handheld devices such as the Palm and the Pocket PC are targeted at the mobile computing environment. Another important factor in the wireless environment is the Bluetooth RF (radio frequency) infra-

structure, for which many manufacturers are building compatible devices.

### Challenges

Although these services aim to increase access to a user's critical information, they rely on users to create, change, or maintain configurations for this access. In addition, the burden of incorporating these many devices and services into a daily regimen is taxing on the user and highly error prone. The key shortfalls, however, rest in the personalization of the user's interaction with each of these separate systems and in the changes and dynamics that individual users will need to coordinate and use these services effectively. This can be especially difficult for a user who has a disability, who is aging, or who might need or prefer a different interaction mode.

To make these systems truly usable,

**Intelligent  
Systems**

## Call for Papers

[www.computer.org/intelligent/author.htm](http://www.computer.org/intelligent/author.htm)

**Submission Guidelines:** Submissions should be 3,000 to 7,500 words (counting a standard figure or table as 250 words) and should follow the magazine's style and presentation guidelines. References should be limited to 10 citations.

**To Submit a Manuscript:** To submit a manuscript for peer-reviewed consideration, please access the IEEE Computer Society Web-based system, Manuscript Central, at <http://cs-ieee.manuscriptcentral.com/index.html>.

## Special Issue on Intelligent Manufacturing Control

**M**anufacturing organizations face conditions of unprecedented disruption and change, and systems that control the many operations along the manufacturing supply chain must be able to adapt to these conditions. Recently, a major thrust in addressing these requirements has been the application of tools from distributed AI. The deployment of tools such as neural networks, fuzzy logic, and evolutionary programming has provided new routes for tackling complex issues in scheduling and control of manufacturing processes. In addition, manufacturing control and management systems based on the *multiagent system* paradigm have received significant attention, because they promise to provide a high flexibility and easy reconfigurability in the face of changes. A closely related development is the *holonic manufacturing systems* methodology, which couples intelligent software elements such as agents with physical entities such as equipment, orders, and products to effectively provide a "plug and play" factory. Many of these developments are now at the point where industrial deployment is a serious possibility and major systems vendors are considering integrating intelligent control capabilities into their product offerings.

This special issue will feature articles that address the issue of develop-

ing intelligent control systems for the manufacturing supply chain. In particular, the issue will aim to position this work in terms of its potential longer-term impact on industry and on the issues required to see more widespread deployment. In addition, the issue will explain some fundamental research concepts in this area.

For this special issue, we invite original, high-quality submissions that address all aspects of intelligent control as it is applied to the manufacturing supply chain. Submissions must address the issues of how the developments described will impact the manufacturing supply chain and what barriers to their adoption exist. Papers addressing performance evaluation of intelligent control systems versus more conventional systems will be extremely welcome.

### Guest Editors

Duncan McFarlane, University of Cambridge  
Vladimir Marik, Czech Technical University  
Paul Valckenaers, Katholieke Universiteit Leuven

**Submissions due 1 May 2004**



## STANDARDS, TOOLS &amp; BEST PRACTICES

service providers must bridge the user's networks (office, mobile, and home), devices (phone, Web pad, and computer), and information spaces (email, documents, and news and information).

Key to interaction across devices and information spaces is the ability to provide the solution in a ubiquitous form factor. Users should be able to interact with the devices and services around them in a consistent way, appropriate for the personal device they're using and their own interaction preferences.

### Toward solutions

The URC standards address these challenges by providing a means for targets to advertise their functionality and make it available. User devices can then interpret compliant targets and use or generate appropriate user interfaces.

Several organizations are building

systems based on the V2 draft specifications. They've publicly demonstrated their initial implementations in several venues over the last two years.

The market is ripe for third-party developers to build products that carry out interface transformations for content, appearance, and user controls and to build applications that lend themselves to interface selection or transformation. Many of these already exist. The *User Interface Markup Language* uses Sun Microsystems' Java Swing to modify rendering of interface elements. UIML is compatible with and expressible by XML. Sun is also pushing its Jini connection technology as a means of mediating access to alternative interfaces (the URC standard would run over Jini). Other venues include the Universal Plug and Play (UPnP) forum, HomeAPI, and HomeRF.

The URC standard, if widely adopted, could enable consistent, personalized access to devices and services available in a pervasive computing environment. The V2 committee is looking forward to receiving comments on the draft standards during the public review period. ■

## ACKNOWLEDGMENTS

This article is based on the work of all the current and former members of V2 and its predecessor organization, the National Committee for IT Standards IT Accommodation Study Group. We particularly thank Rex Lint, who edited the preliminary business case, and Wayne McCoy, who edited the original requirements document.

## REFERENCE


1. V. Stanford, "Using Pervasive Computing to Deliver Elder Care," *IEEE Pervasive Computing*, vol. 1, no. 1, Jan.-Mar. 2002, pp. 10-13.

The  
IEEE  
Computer  
Society

publishes  
over  
**150**  
conference  
proceedings a year.

for a  
preview of  
the latest  
papers  
in your field,  
visit

Not a member?  
Join online today!



[www.computer.org/proceedings](http://www.computer.org/proceedings)

**Bill LaPlant** is a consultant on IT policy and standards implementation and is the chair of V2. Contact him at [blaplant@computer.org](mailto:blaplant@computer.org).

**Shari Trewin** is a researcher in the Pervasive Computing Solutions department at the IBM T.J. Watson Research Center and a coeditor of the V2 standards. Contact her at [trewin@us.ibm.com](mailto:trewin@us.ibm.com).



**Gottfried Zimmermann** is an independent IT consultant working for the University of Wisconsin and a coeditor of the V2 standards. Contact him at [zimmer@trace.wisc.edu](mailto:zimmer@trace.wisc.edu).



**Gregg Vanderheiden** is a professor of industrial and biomedical engineering at the University of Wisconsin-Madison and the director of the University's Trace Research and Development Center. Contact him at [gv@trace.wisc.edu](mailto:gv@trace.wisc.edu).

