n Proceedings of IET Intelligent Environments, Athens, Greece, July 2006.

ZUMA: A Platform for Smart-Home Environments
The Case for Infrastructure*

J. van Greunen, Y. Markovsky, C. R. Baker, J. Rabaey, J. Wawrzynek, A. Wolisz#

University of California, Berkeley, USA
{janavg, yurym, crbaker, jan, johnw, wolisz}@eecs.berkeley.edu

Abstract: A wide range of disconnected communication networks is emerging in the home. Each has different functions and implementation architectures. The user experience as well as the overall functionality would be vastly enhanced if these networks could collaborate in a seamless fashion. Collaboration is challenging because it requires disparate devices and components to interoperate in a user friendly, scalable and upgradeable fashion. We conjecture that such cooperation might be most efficiently assured using a supporting infrastructure. The *Universal Contents Router (UCR)* is introduced as an infrastructural component. The UCR moves the burden of connectivity away from the end devices to the system core. The paper spells out ZUMA, or functions that a UCR should support, discusses UCR architecture and presents a first-order prototype.

1 INTRODUCTION AND MOTIVATION

A number of independent communication networks are emerging in the residential space. Personal electronics and multimedia devices are the most noticeable. However, other devices, such as appliances, are now also communication enabled. The evolution of these networks is outlined below.

Voice, one of the earliest home communication systems, developed from a single line to a cluster of cordless devices connected to a central "base station". Cellular phones, with different manufacturers, user interfaces, and a distinct set of functions, operate in parallel.

Further, data communication has evolved from a single computer with low speed Internet access to a local area network that interconnects numerous desktops and laptops. Recently, high-speed connections, such as DSL, have become commonplace. In addition, wireless technologies are used to add extra mobility in the home. They are used to extend the LAN (WiFi) or to connect wireless peripheral devices to PC's (via Bluetooth).

Communication infrastructures are also being developed for the home automation and monitoring spaces. Automated air conditioning and light control systems have the potential to control peak power consumption, and increase energy efficiency [13]. These systems use proprietary commercial communication networks (e.g. LonWorks and BagNET). Manufacturers are also outfitting white products (laundry machines, refrigerators, etc.) with communication capabilities. Moreover, the house may have a fire alarm, security and/or a surveillance system. These are connected to a dispatch center and may have several camera feeds that need display devices (or can use the in-house TV circuit).

Last, entertainment systems are becoming more complex to set-up. The number of media devices within the home is increasing, for example, portable cameras, MP3 players, and complex HiFi sound systems are being added. Most of these devices can only be connected using specific communication technologies e.g. point-to-point wiring. Further, devices are controlled though dedicated interfaces (keyboards, infrared remote controls).

In summary, a house contains stand-alone components or isolated clusters of networked elements. The possibility of interconnecting these components could open the door for enhanced user experiences. Let us consider a small example as motivational scenario:

> *After dinner John walks into the living room and turns on the television. This starts a multimedia-viewing application. John likes to watch violent sports, and according to his preferences, the television automatically switches to WWE wrestling on the set-top box. John watches for a while. At 8pm a reminder from John's PDA is displayed in the corner of the television. John dismisses the reminder.*
>
> *Later John's son Adam walks in with his laptop. Adam is a minor and is not allowed to watch WWE. The television immediately stops showing WWE and replaces it with a content selection menu. In the background, the system keeps recording the WWE channel for John and also stores the time at which John stopped watching. This will allow John to resume watching WWE later.*
>
> *John now selects "Finding Nemo" (from Adam's laptop). "Finding Nemo" starts playing on the television set. The lights in Adam's room and his radio set are turned off. Sometime later there is a noise at the front door— the view from the camera at the garden gate is blended into the corner of the TV set. However, no threat is detected. When the movie is 10 minutes from the end, Adam's bedroom heater turns on.*

---

*This work is part of the ZUMA project, funded by the Gigascale Systems Research Center. www.gigascale.org

A. Wolisz is affiliated both with TU Berlin and UC Berkeley.

This scenario demonstrates a number of concepts. First, it establishes the requirement of universal connectivity—for example, between the television, a laptop, the house heating system, and a PDA. Second, new devices (such as Adam's laptop) must be seamlessly integrated into the system. Third, the system consists of heterogeneous devices (sensor nodes, laptop, set-top box etc). These devices range from powerful and flexible to very cheap, closed components. Fourth, content is decoupled from the rendering devices. To render content on multiple devices, the content may need to be reformatted and trans-coded. Fifth, the system must be able to detect environmental changes such as the movement of people (called ambient events). Sensor networks can provide this information.

At a higher level, users benefit from the integrated environment (further called an Ambient Intelligence System or AIS) by being able to set preferences. Further, applications can utilize user activity information to optimize their performance. For example, the home automation system can adjust heating, lighting, and turning off unused media devices according to user activities. (The California Energy Commission projects media devices' standby power will soon equal that of a refrigerator.)

In order to integrate all devices in the home environment, formidable challenges must be overcome. While new devices and appliances will have wireless interfaces, these devices will not necessarily interoperate. Standards can go a long way in helping devices to collaborate, but are not sufficient. Standards emerge and become obsolete at a rapid rate, while the networked home should remain operational for decades or more. Finally, the user should not be aware of the system's complexity [15].

In this paper (and in its companion paper [6]), we present a platform for the smart-home, called **ZUMA**. The main properties of ZUMA are enumerated below:

- Zero-Configuration: minimal need for device configuration.
- Universality: ability to connect any device to any other device.
- Multi-User Optimality: optimized user experience, in the presence of multiple users and many simultaneous tasks.
- Adaptability: the platform has the ability to change according to users' desires, their presence, and the integration of new devices.

To accomplish these objectives, we propose a two-pronged approach. From the top-down, we define a set of abstractions that make it easy to deploy and execute an application on the home platform. From the bottom-up, we provide a strategy to seamlessly connect devices with widely diverging interface formats, effectively accomplishing the *Universality* goal of ZUMA.

The latter is the topic of this paper, in which it is argued that the best way to meet the objectives is to build the home system around an infrastructural core, called the "Universal Content Router(s) (UCR)". The UCR approach provides a common backplane with the following properties:

- Supports any interface or format, at any level of the protocol stack.
- Future proofs the system with an extensible core, while maintaining backward compatibility.
- Optimizes the quality of service (QoS) for simultaneous tasks via a flexible resource management strategy.

The rest of the paper is organized as follows: In section 2 we make the argument for an infrastructure-based approach, while in Section 3 the functionality requirements for such an infrastructure are presented. A basic architecture for the UCR, contrasted with existing systems, is illustrated in Section 4. Section 5 discusses the UCR's computational requirements. Section 6 presents a prototype implementation. Section 7 concludes the paper.

2 A CASE FOR THE SMART-HOME INFRASTRUCTURE

We envision that most, if not all, components in the smart-home will have a wireless interface. Currently, a multitude of standards co-exist: cellular phones operate in the licensed 800-900 MHz and 1.8 and 1.9 GHz band; cordless phones, computers, sensor networks, and peripherals all compete for the unlicensed 2.4 and 5.2-5.8 GHz bands using a variety of physical layers and protocol stacks such as Bluetooth, 802.11 a/b/g and 802.15.4; RF remote controls and security systems operate in the unlicensed 400 and 800 MHz bands; and most multimedia remote controls use infrared signalling. Most of these standards were introduced in the last decade. It is very unrealistic to hope that all these standards will be replaced by a new universal solution. The diverse needs of different devices in terms of bandwidth, latency, power consumption, and cost preclude such a solution.

Further, even with physical connectivity (via a wireless protocol), universal connectivity is not ensured. Different data representations at the application layer may prevent communication. For example, media information can be represented as M-JPEG or H.264. Several solutions can to address the universal connectivity challenge:

The stovepipe solution: a single company or group of companies select a set of vertically integrated standards. Compliant devices are ensured interoperability on all layers. Examples of this approach are the HAVI standards (for wired multimedia based on the IEEE1394 physical layer), or the Lon and BAGNet networks for home automation. The problem with these approaches is that they are exclusive in the covered functionality, and do not allow for dynamic expansion of this functionality.

The flexible peripheral approach: Peripheral devices can be made flexible and adaptable so that they can support a variety of protocols and formats. Examples of devices with multiple standards are the cell phone and the PDA, which now carry a variety of wireless interfaces (GPRS, Bluetooth, 802,11), and provide software/hardware support for a variety of multimedia formats. The disadvantage of this approach is that it is expensive, power consuming, and not upgradeable.

The network overlay solution: This approach uses gateways to connect different networks; for example, coupling an 802.11 network to an 802.15.4 sensor network. This solution may have complexity and management problems. Yet, it is the only one that takes the responsibility of connectivity away from the end-devices and moves it to an infrastructural element.

The approach advocated in this paper presents a powerful generalization of the third option. In a sense, it mirrors some of the evolutions that took place in the LAN area. The original Ethernet was, in principle, a completely passive broadcast medium (the infamous yellow cable), with network adapters and networking software residing at individual computers. Access control and processing was distributed over the population of hosts. As the number of end-points increased, this approach became impractical: the system was hard to manage, any break in connectivity blocked the whole network; the bandwidth was shared by all end-systems; and all connected end-systems had to operate using the same signalling rate. To solve these problems, a switch was placed in the middle. Initial reliability concerns (the switch as a single point of failure) have proven to be misplaced. Switches can be built in a robust technology in contrast to the end-systems with vulnerable operating systems.

Our approach to the organization of the home infrastructure, the *Universal Content Router* (UCR) [15] is based on the same foundations. We argue that concentrating processing power into a few central units is the ideal infrastructure for the smart-home environment. These central units should be extensible so that they can interoperate with new peripherals. From a system management point of view, it is simpler to deploy, update, and maintain a few key pieces of infrastructure than a heterogeneous collection of devices. The economics of centralization are also favorable; after the initial cost of infrastructure deployment, peripheral units can be added at low cost.

Finally, the UCR provides connectivity. This connectivity is not reduced to the physical or even transport-level, i.e., “provision of byte streams”. Rather, it encompasses the concept of exchanging semantically interpretable units of content, resulting in interoperability at the presentation or (partially) the application layers. Any device in the system must be logically accessible by all other devices. If devices cannot communicate with each other, then they are not part of the same system.

3 UNIVERSAL CONTENT-ROUTER FUNCTIONALITY REQUIREMENTS

For two devices to communicate, they must be compatible at the various network layers defined by the OSI (open system interconnect) standard. However, it is unreasonable to expect that all devices will adhere to common standards for all layers, because devices and applications have different constraints that are best served by different wireless technologies (for example, WLAN for high data-rate and cellular networks for longer-range communication). In the absence of a single standard, a gateway device, or “router”, is required to enable communication. The infrastructure component in the middle performs seamless provisioning and manipulation of content— hence the term *content router*.

This section discusses the multiple requirements which we have identified for such device..

3.1 Connecting devices with different physical layers

Building on the assumption that most end-points will support wireless interconnection, the UCR should provide flexible and adjustable wireless interfaces. The emerging technologies of software-defined radios (SDR) and mostly-digital radios (MDR) are making such functionality increasingly plausible. In a fully flexible implementation, new interfaces can be uploaded and installed as they become available. The UCR should also connect to standard wired interfaces (such as Ethernet).

3.2 Protocol conversion

The UCR must be capable of supporting the different networking stacks. Some networks use TCP/IP, UDP or RTP with SIP signalling, others, such as sensor networks will use dramatically different packet structures, media access mechanisms, addressing, and routing strategies. Protocol conversion functions can be computationally intensive and rely on the implementation of complex state machines, timers, and bit-stream operations.

3.3 Format conversion

Connecting components with different data and frame rates, encoding and data representation formats, requires trans-coding and reformatting. While the signal processing requirements associated with real-time conversion of multiple simultaneous streams (in a flexible fashion) may be quite formidable today (in fact, most current media processors use hardwired accelerators), technology scaling is promising unprecedented computational power.

3.4 Routing

The dynamic environment offered by the home with its multitude of devices offers an interesting routing challenge. In the AIS environment routing will be influenced by the physical location of the different users, applications, and their preferences. These also have a profound impact on the way addressing is performed. For instance, in a sensor network data is accessed in a region-based fashion. In multimedia delivery, the display may be addressed by user position (i.e. nearest). Therefore, the router should be location and capability-aware.

3.5 Experience enhancement

The UCR can modify its routing to enhance the user experience. It needs to be aware of the location and the capabilities of the peripheral devices, the preferences and the location of the end-devices and the ambient properties of the environment (visibility, position of light sources, reverberation and echo characteristics). For example, rather than relying on a small number of high quality speakers, superior audio playback can be obtained by careful beam-forming of the audio by an array of cheaper speakers.

3.6 Discovery of new devices and capacities

To enable ease of deployment and dynamic self-configuration, the UCR must discover the presence of active components, detect new components, and automatically upload the necessary interfaces to interact with them. Discovery can be performed in many different ways. A UCR may scan the spectrum in a continuous way to detect the presence of candidate devices (a typical function of the cognitive radio, described in Section 3.8), or broadcast beacons on dedicated control channels. Conversely, a peripheral device may periodically broadcast its presence on a control channel. Upon connection, specific information about the device requirements and capabilities are exchanged.

3.7 Security and media rights protection

Protecting licensed media data is an essential component of any modern digital media system. The current approach, Digital Rights Management (DRM), only works for point-to-point connections, and is an impediment to the universal connectivity paradigm promoted in this paper. The UCR opens the door for a radically different way of rights management. In addition to rights management and encryption of sensitive data, the UCR should also manage the security of the home networks and ensure its privacy.

3.8 Optimization of resource utilization

Home infrastructure will have constraints on the cost, size and power of devices. In addition, efficient utilization of the wireless spectrum is essential. The support of simultaneous high-data rate streams is beyond today's commercial technology. This is especially true in a dense urban environment, where there is interference from the neighbours' networks. The current unlicensed bands (900MHz, 2.4 GHz, and several bands in the 5.2-5.8 GHz range) are not sufficient to support these high-bandwidth applications.

The available spectrum, both unlicensed and licensed, can be used more creatively. *Cognitive Radio* is one way of doing just that [13]. As per the FCC, a Cognitive Radio (CR) is a radio "capable of changing its operating characteristics based on interaction with the environment in which it operates". The key properties of cognitive radios are:

- Sensing: RF technology that "listens" to huge swaths of spectrum.
- Cognition: Ability to identify primary users (PU's) of the spectrum[1] (based on power profiles and/or other footprints).
- Adaptability: Ability to best use unused spectrum (includes changes to operational frequency bands, power, and modulation).

CR's present many challenges: First, the analog front-end must be able to transmit and receive signals over a wide range of frequencies. Second, detecting PU's is challenging due to the nature of the wireless channel (fading, shadowing and multipath effects). In addition, there are hidden terminal effects in wireless channels. To effectively address these challenges, a single CR must have radio sensitivity above what can be provided in a peripheral device [17]. Multiple CR's can also be used to detect PU's [11]. Hence, the UCR's in multiple households can cooperate to manage spectrum.

3.9 Scalability

The home network architecture should scale to cope with an increasing number of end-points, users, and applications. Nothing prevents the backbone from being a network of UCRs, connected using wired or high-bandwidth wireless channels, while from a top-down perspective the network looks like a single UCR

---

[1] Primary users – users which have been allocated spectrum based by the FCC and are legally entitled the use of the spectrum.

4 ARCHITECTURES FOR UNIVERSAL CONTENT-ROUTERS

The connected home has drawn much attention over the last few years. A wide range of industry sectors, ranging from consumer, communications, computing, to software, are promoting their particular solutions to the problem. To our knowledge, none of these are as comprehensive as the UCR platform concept promoted in this paper. In this section, we will first outline some of the solutions promoted in industry and then discuss the proposed architectural solution.

4.1 Commercial Solutions to Universal Connectivity

Universal Plug and Play (UPnP) [3] is an open (written by a consortium of PC and consumer electronics companies), peer-to-peer, standards-based, networking architecture for connecting PCs, intelligent appliances, and other mobile, wireless devices. UPnP is designed to be content and operating system independent and rely on existing networking technologies (*e.g.* TCP, IP, HTTP, XML, *etc.*) for interconnection and communication between devices. UPnP provides specifications for various devices (*e.g.* Internet gateway, HVAC, media serving/rendering, wireless LAN, printer/scanner, *etc.*) that define the device's control protocols (description, control, events, presentation). A discovery protocol is also specified. Discovery occurs when control points hear a particular device's discovery protocol.

Another home solution is Microsoft's popular operating system, Windows XP, preinstalled with a Media Center application and packaged as Windows XP Media Center Edition (MCE) [5]. MCE is a media entertainment-centric solution, designed to run on a commodity PC. The latest edition can support up to four TV tuners (two analog, two HDTV); however, it can only record two programs simultaneously. In addition to video, it can manage audio and photo content. MCE can provide TV-guide, scheduled recording (TIVO-like), and DVD-burning capabilities. With additional wireless or wired extenders, other devices in the home (e.g. a bedroom television) can utilize the content on the MCE PC. Add-on kits also exist for the XBOX game system.

UPnP, MCE, and other solutions ([4][1][2]) provide limited functionality with respect to an ambient intelligent system (AIS). MCE has limited scalability and is inflexible at the physical layer. UPnP is more scalable than MCE, but is still limited in many ways. For example, UPnP only specifies peer-to-peer connectivity, and while this is certainly necessary, it is not sufficient for an AIS. The failure of UPnP to provide an over-arching system architecture means that UPnP cannot possibly manage a diverse set of devices and technologies efficiently, even if those devices can communicate. For example, UPnP cannot decide how and where to perform trans-coding, how to adapt the system to the presence of users QoS, and scheduling remains unresolved. UPnP stands to be relegated to the AIS in the same way the network is to an operating system.

4.2 Proposed Architecture

We propose an architecture that brings together the requirements of flexibility, reconfigurability, high content throughput, and scalability. The UCR has a *"router-oriented" architecture*, with multiple streams being processed concurrently. In that sense it resembles a network processor, yet with a far more demanding and flexible functionality.

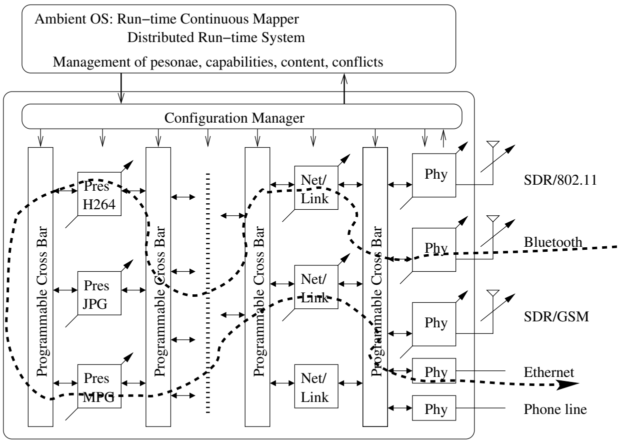

Figure 1: Logical structure of the UCR.

The architecture shown in Figure 1 contains three key components: a set of radio front-ends, a set of compute blocks, and a set of cross bars. All three types of components must be configurable and programmable at run time. A "Configuration Manager" module performs the configuration of each of these blocks. This module in turn is controlled by the Ambient OS, the subject of the companion paper [6]. The configurable radio front-ends are an essential requirement to the router's ability to interconnect distinct devices, as discussed in section 5.1.

Component programmability serves dual purposes in the proposed architecture. First, programmability ensures that the system is future-proof. Second, it enables the Ambient OS to adapt the system to changes in the environment, requirements in connectivity, and types of devices. The compute blocks can be programmed at run time to implement a range of required functions in the proposed platform: base-band processing and modulating for the physical layer; network and link protocol layers; and, content format decoders and encoders, filters and other transformations. All these operations can be chained together in an arbitrary flow-graph that converts communication protocols and content formats on the fly (shown with a broken line in Figure 1). Flow-graphs are not restricted to processing a single stream of content, but can include multiple streams and fan out.

Configurable compute blocks can be implemented as conventional microprocessor cores, DSP processors, Application Specific Integrated Circuits (ASICs) and Field Programmable Gate Array (FPGAs). Each implementation choice has a distinct price, performance, and power point in the implementation space. To be universal, the compute blocks must permit run time reprogrammability and meet real-time content processing requirements.

Programmable crossbars serve a key function in connecting all compute blocks. While the physical implementation of this component is unspecified, it must be logically able to interconnect compute blocks to form a content processing flow graph. The only requirement for the physical implementation of the cross bar is to provide sufficient bandwidth to transmit content between the compute blocks. Furthermore, it is important to enforce constraints on the latency through the cross bars to enable synchronization of multiple streams.

The proposed architecture scales in a straightforward manner. By increasing the number of compute blocks available to the "Configuration Manager" and increasing the number and the bandwidth of the cross bars, the system will offer proportionally more content throughput. Alternately, a home could contain several Universal Content Routers interconnected together and managed by the Ambient OS. This enables the system to process multiple content streams concurrently. The Ambient OS will be directly affected by the platform scaling, since the OS manages *all* reprogrammable resources. In practice this is not a critical problem. Events in the home occur relatively infrequently (at most several events per second), which means that the Ambient OS does not need to perform complex reallocation and resource reprogramming quickly. Latency or "system reaction time" of less than 100ms is actually acceptable in most cases; it is comparable to the human perception time and easily achievable even in a modest implementation.

On the other hand, the compute blocks require high performance and high content processing throughput particularly in light of modern content coding standards (*e.g.* H.264) and complex base-band processing (*e.g.* Software Defined Radios). The computational requirements of these and other key tasks are evaluated in the next section.

5 FUNCTIONAL REQUIREMENTS MAPPED TO THE PROPOSED ARCHITECTURE

The physical layer requirements are discussed first, followed by protocol conversion and format conversion requirements. System requirements raised in the previous section are discussed in the companion paper [6].

5.1 Connection of heterogeneous devices

In the smart home environment, devices with distinct wireless and wired communication standards must be seamlessly connected. Furthermore, distinct link-layer, networking and transport protocols must all be bridged. In the following discussion, we highlight techniques for flexible, programmable radios and look at computational requirements to support this technology.

Flexible Radio Spectrum Use. A truly flexible AIS must adapt to new wireless protocols on-the-fly. First, to adapt to a new wireless protocol, the AIS must be able to detect the new signal when it is transmitted. This can be achieved by modifying the front-end of the receiver to scan a wide band of frequencies [10]. Feature detection can be used to identify a frequency profile. However, the AIS needs additional protocol specifics to decode the signal. Unfortunately, simply listening to the new signal will not yield specifics in a timely manner.

We envision the following sophisticated protocol discovery system: After the AIS has detected a signal and its frequency profile, it can connect to an online database containing frequency profiles for all known wireless standards. The AIS then picks the closest match, *implements* it and decodes the signal. Different known standards can be tried until the decoded signal is intelligible.

From a hardware viewpoint, it is not trivial to implement a flexible wireless protocol. This is because traditionally a large portion of the baseband, intermediate frequency and front-end processing is done in analog hardware. These steps must be reprogrammable in flexible radios. In other words, the radio protocol must be defined in software. Such a radio is called Software Defined Radio (SDR) and it can support multiple wireless standards with the same hardware. With this flexibility, new wireless standards can be programmed on the same hardware.

To transform a conventional transceiver into an SDR, the digital processing needs to be placed as close to the antenna as possible, and analog processing must be minimized. More specifically, an SDR must have its channel modulation waveforms defined in software [14]. This means that the modulation waveforms must be generated as sampled digital signals, converted from digital to analog with a wideband Digital to Analog Converter (DAC). Similarly, received signals must be digitized using a wideband Analog to Digital Converter (ADC) that captures all of the channels of the software radio node. This digital signal is then extracted, down-converted to its baseband frequency and demodulated, often using software on a general-purpose processor.

An SDR can be implemented using a general-purpose processor, but it may be more advantageous to use reconfigurable hardware to do so. The most prevalent reconfigurable devices are FPGAs.

Wireless computation requirements. To quantify the computational requirements for wireless receivers, a receiver that combines Bluetooth (which uses Gaussian Phase Shift Keying (GFSK)) and HiperLAN/2 (which uses OFDM) is considered. However, the requirements for the HiperLAN/2 portion of the receiver will be quoted as it has a higher data rate and is more computationally intensive. Figure 2 shows the components of a SDR transceiver architecture. Each component of this architecture will be considered separately in the computation requirements analysis given below.

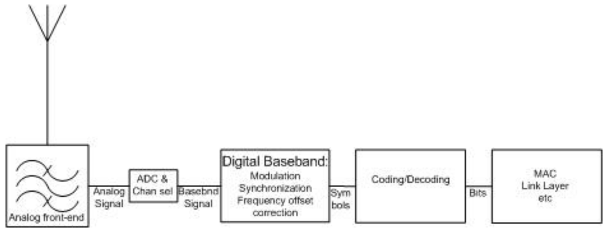

Figure 2: SDR transceiver architecture.

Analog to Digital Conversion & Channel Selection. Shannon's sampling theorem establishes a minimum sampling rate and resolution to prevent aliasing when an analog signal, at a given frequency, is converted into the digital domain. These samples are then digitally filtered to select the frequency band of interest. [16] found that a sampling rate of 80 million samples per second (MSPS), with 12 bit resolution and a 25-tap FIR filter was sufficient. Assuming symmetric FIR filters and a decimation of 4, the computational load is: 500 million 15-bit multiplications and 960 million-bit additions per second.

Demodulation. OFDM demodulation consists of four parts: frequency offset correction, a 64-point FFT, channel equalization and QAM demodulation. The computation requirements (the sum of multiplications, additions and comparisons) for each step are listed in [16], and summarized in Table 1.

TABLE 1: Computation Requirements for OFDM demodulation

Stage	MIPS	#bits
Freq. offset corr.	38.4	16
FFT	230.4	24
Channel eq.	28.2	16
QAM-demod	38.4	16
Total	335.4	

In total, a HiperLAN/2 software radio requires computational power of roughly 1.5 Billion operations per second (where those operations are additions or multiplications). Note, this approximation does not include the operations required to synchronize the receiver, which depends on the transmission pattern.

In the intelligent home environment, several streams of both video and audio can be active at the same time. Moreover, the computation estimations above do not include packetizing, framing, or transmitting the data.

5.2 Format Conversion

Multimedia processing, protocol processing and wireless communications are computationally intensive tasks, and come at a substantial implementation cost, especially when implemented in a flexible, adaptive format. This section gives a summary of the computational costs of the dominant format conversion tasks of the UCR.

Multimedia encoding and decoding. The amount of processing varies greatly according to the type of multimedia (audio vs. video vs. image), the quality (size, resolution etc.), and the type of encoding used (e.g. MPEG4 vs. MPEG2, etc.). Figure 3 gives a first-order estimate of the MIPS (millions of instructions per second), memory, and data-rate requirements for both audio and video processing (as obtained from [8] and [9]).

The video estimates in this chart are based on MPEG-4 CIF, which at a resolution of 352x288 and 30fps is a fairly small size video format used for online video streaming. Standard television is approximately 4-CIF (4 times the resolution), while High-Definition Television is 16-CIF. In this implementation, the encoding and decoding code was hand-optimized, and the MIPS are given in DSP instructions. A complex instruction on a DSP may translate into several instructions on a general processor. MPEG-4 decoding requires roughly 31 MIPS, while encoding varies from 200-400 MIPS, depending on the desired quality.

In the case of MPEG-4 processing, the encoder is substantially more computationally intensive than the decoder. In AIS, we may need real-time encoding as part of a trans-coding step to display a movie on a different display. In general there is a trade-off between encoder and decoder complexity and new standards are emerging with more balance between encoding and decoding. These will therefore permit "light-weight" encoding.

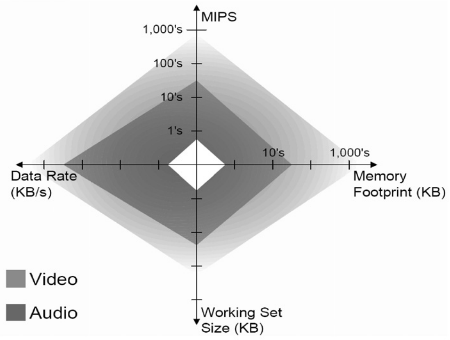

Figure 3: Computation requirements for video and audio processing [8].

Typically, DSPs, with complex instructions (like multiply-accumulate), are used to process video in real-time. DSP's. Currently, a dedicated DSP chip is used per video strean. Similarly, video decoding requires a large portion of the resources on modern FPGAs. MPEG-2 decoding takes about 65% of the Altera Stratix II EP2S15 [8].

6 A PROTOTYPE UCR IMPLEMENTATION

We have developed a prototype platform that attempts to match the requirements for the smart home infrastructure. The prototype serves as a test-bed for research about features needed in the UCR. In other words, the prototype is a platform to experiment with implementation architectures and the run-time system environment that controls its operation. Although a full implementation is beyond the scope of an academic project, the prototype implements several tenets of ZUMA. The prototype set-up is described first, followed by a description of the prototype demonstration and an evaluation of the prototype according to the tenets of ZUMA.

6.1 Prototype Architecture

At the heart of the prototype is the BEE2 FPGA module [12] that implements the Universal Content Router (Figure 4). In general, FPGAs contain microprocessor cores and run-time configurable logic blocks. The microprocessor cores run software and are well suited for complex control tasks (*e.g.* resource management, handling special case, and user interaction) that do not demand high throughput and performance. The configurable logic enables a designer to construct on-chip content processing data-paths with throughput comparable to ASICs.

The BEE2 board contains five FPGAs: one master that runs the Ambient OS, and four slaves that are programmed with required trans-coding and other content transformation functions. The Ambient OS, discussed in the companion paper [6], performs continuous mapping and allocation of resources to accommodate user requests and meet environment and user specified constraints. The Ambient OS manages all the components in the architecture: (1) compute blocks (slave FPGAs) are programmed at run-time with trans-coders (*e.g.* MPEG4 to Motion JPEG video trans-coder), and (2) the programmable cross bars (interconnections between the outside world and the slave FPGAs) are configured to create content processing pipelines.

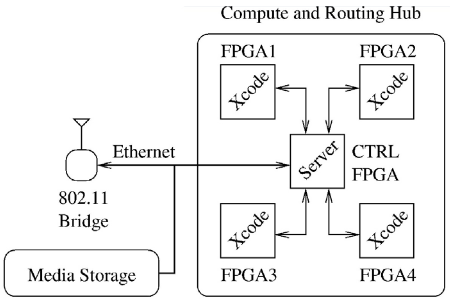

Figure 4: Simplified diagram of the prototype UCR.

The prototype does not currently contain configurable radio front-ends. The communication to the "outside world" is performed through a wired Ethernet connection and an 802.11n bridge. This shortcoming is only temporary as we are currently looking toward integration of Software Defined Radio research with the prototype.

6.2 Prototype Demonstration

We chose a home multimedia-watching scenario to demonstrate the prototype's capabilities. In the demonstration, a user selects a video from a remote computer using a PDA as a remote control device. The Ambient OS sets up a session to stream the selected video from its source to the display located in the same room as the user. The demonstration contains two rooms with a display in each one. A small sensor network with the help of pressure-sensitive floor mats detects the location of the user. As the user moves from room to room, the Ambient OS reroutes the video stream from a display in one room to the display in the other. Thus, the video "follows" the user around the house.

To demonstrate the system's adaptability, the two displays do not accept the same video encoding standard. One display works with MPEG4 and the other with Motion JPEG video. As the stream is rerouted between the two displays, the Ambient OS automatically recognizes the format mismatch, and programs a slave FPGA and the communication cross bars to perform on-the-fly trans-coding before the content is shipped to the rendering device. Most devices in the demonstration share a common IP network via the wireless 802.11n router.

The hardware components of the prototype are listed below:

Rendering Devices (Audio/Video). The prototype includes three video displays and one set of speakers. The displays are encapsulated in decoding clients with an 802.11g front-end and an FPGA processing module to convert the network format to the display's native format. This rendering client is illustrated in Figure 5.

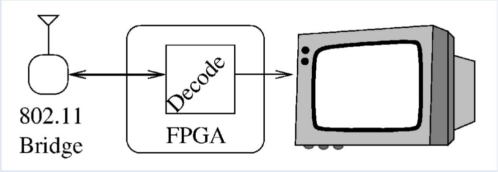

Figure 5: Logical client architecture: the client handles control and data packets.

Multimedia Sources. The demonstration includes two small personal computers (MacMini's), which run the Linux operating system. The computers are used for their hardware storage capacity and contain lightweight clients that stream content upon request.

Control/UI. The remote control device provides command entry, content selection and feedback to the user. A simple remote control is implemented on a small Nokia 770 Internet Tablet device.

Sensor Devices. Pressure sensing floor mats are used to detect the location of people. The demonstration contains two floor mats, where each mat corresponds to a room with one or more content rendering devices.

6.3 Demonstrating ZUMA

To achieve Zero-configuration, Universality, Multi-user optimality and Adaptability of the prototype requires a combination of the right choices in the architecture components, the software stack and the Ambient OS. The architecture components and the way they enable ZUMA is discussed below.

Zero-configuration. Zero-configuration is the ability of the system to integrate new devices, content, or users without manual set-up. The burden of configuration is placed on the Ambient OS and is described in [6].

Demonstrating Universality. In the demonstration, different devices are bridged together around the BEE2 board. The dynamic configurability and processing power of the BEE2 chips enables the prototype to translate content representation formats in real-time. For example, the demonstration contains a MPEG4 video source that is trans-coded into Motion JPEG on one of the slave FPGAs. The prototype enables two incompatible applications (devices) to communicate.

All the devices share a logically common IP network. IP was chosen because it is a very flexible and well-supported choice. The IP network is implemented on top of wired Ethernet, wireless 802.11g and Mica2 900Mhz connections. The BEE2 hub offers different radio front-ends with the help of 802.11g bridge and Mica2 900Mhz access point node. In the demonstration, a sensor node (900Mhz) communicates with the BEE2 node (over 802.11g).

The prototype demonstrates a first step toward universality. A more general solution would include Software Defined Radio and would require a few simple extensions. First, the prototype must be extended with a wide-band radio front-end. Second, the server FPGA must be able to find and download digital base-band processing cores for FPGAs for different wireless standards. The Ambient OS on the server FPGA can connect to the Internet to download these implementations, similar to the way trans-coders for different data formats are downloaded.

Demonstrating Multi-User/Task Optimization. In general, multi-user optimality involves arbitration to resolve user conflicts, as well as ensuring that the hardware is capable of supporting the user-desired functionality. The Ambient OS handles conflict resolution and resource management [6]. Resource utilization is specifically addressed after the following section.

Demonstrating Adaptability. There are two parts to adaptability. First, the prototype must be able to handle changes in the environment (devices, content, or users). Second, the prototype must be future proof. Our demonstration highlights automatic adaptation to a person moving around the house. The system relocates a running video stream from one video display to another, *i.e.* the video ``follows" the person around the house. This includes dynamically selecting an appropriate trans-coder as the video switches between incompatible devices. The real time trans-coders are dynamically mapped and configured into slave FPGAs. Thus, the prototype can adapt to changes in the environment.

Second, the current prototype runs the Linux Operating System on an FPGA and has a connection to the Internet. This puts a range of existing (and future) software, libraries and communication protocols within immediate reach. The prototype can download software and FPGA “gate-ware” as it is standardized or released and load it on one of the slave FPGAs when needed based on a user request or change in the environment.

Prototype Resource Utilization. Although the BEE2 FPGA board contains expensive by modern metrics components, the capabilities it provides will be available at a small fraction of a cost in the future. Its scalable structure applied in the stream-like content processing domain, enables the architecture to take the full advantage of improvements brought by Moore’s law.

The prototype’s has considerable computational capacity required to support multiple users and content streams concurrently. The audio and video clients (Figure 5) require a content decoder and employ Xilinx 2VP30 FPGA that contain approximately 31K logic cells. The BEE2 module has four slave 2VP70 FPGAs for content trans-coding. The 2VP70 offers more than twice of computational resources of the 2VP30. Table 2 provides essential information on the resource utilization for four key codecs used in the prototype. These numbers do not include the cost of overhead such as on-chip buses, memory controllers and others, which typically accounts for 10-20% of the complete design. As the table shows, the employed FPGAs provide more than sufficient resources to enable the proposed infrastructure. Specifically, depending on implementation strategy we can choose smaller and cheaper FPGAs while retaining the same performance, or manipulate several media streams concurrently.

Currently, we do not perform any base-band processing. Assuming that base-band processing would double the computational requirements, we can see that a single slave 2VP70 FPGA would easily contain one or even two complete stream transformation pipelines from an arbitrary wireless standard and content presentation format to any other.

TABLE 2: Computation requirement of media coders on the Xilinx Virtex II Pro FPGAs. The video is 4CIF at 30 frames per second.

Operation	Resources (cells)
MPEG4 Decoder (4 CIF)	11k
MPEG4 Encoder (4 CIF)	14k
Motion-JPEG Decoder	2.5k
Motion-JPEG Encoder	3.6k

Finally, considering wireless spectrum usage. The prototype uses an 802.11g wireless network with a theoretical 54Mbits/sec of bandwidth. Our experiments show that MPEG4 achieves an average compression ratio of 1:30 relative to the raw RGB video. The prototype uses 4CIF frames (704x576 pixels), and at that rate requires about 9Mbits/sec of bandwidth. Assuming only 50% utilization of the wireless link (attributing the rest to the framing and other overhead), the prototype is able to transmit three concurrent video streams through a single channel of 802.11g on the clients. Further, once a wideband radio front-end is added to the prototype to implement SDR for universality, this front-end and reconfigurable baseband processing can also be used to implement a cognitive radio system, which can utilize unused licensed frequencies to obtain more bandwidth.

7 CONCLUSION

The *Universal Contents Router* provides a seamless way to connect the myriad of emerging peripheral components in the Smart Home of the Future. It can hence be considered the cornerstone of any platform for Ambient Intelligent Systems. While a full realization of all the ideas presented in this paper is currently still out of reach (at least at the power, size and cost levels necessary for wide acceptance), technology advances projected for the next few years will definitely make these lofty goals attainable. At the same time, the UCR platform provides an ideal testbed for the exploration of a range of challenging research topics, such as efficient trans-coding, cognitive radio and spectrum optimization, and power and cost effective architectures.

8 ACKNOWLEDGEMENTS

The generous support of the BWRC sponsors as well as the GSRC FCRP centre is gratefully acknowledged. The authors also appreciate all who contributed to the many discussions that helped to formulate the ideas presented in this paper, especially P. Subrahmanian, Jan Newmarch, Dan Burke, and Ken Lutz.

REFERENCES

[1] www.sun.com/software/jini

[2] http://www.havi.org/

[3] http://www.upnp.org

[4] http://drrc.lbl.gov/

[5] http://www..microsoft.com/windowsxp/mediacenter/default.mspx

[6] C. R. Baker, Y. Markovsky, J. van Greunen, J. Rabaey, J. Wawrzynek, A. Wolisz, “ZUMA: A Platform for Smart-Home Environments,” In submission to IEE IE06.

[7] BDTI, “Smart Processor Picks for Consumer Audio/Video Applications” presented at the Embedded Systems Conference, March 30, 2004.

[8] BDTI, “Processors for Consumer Video Applications” presented at the GSPx conference, October 24, 2005.

[9] BDTI, “Introduction to Video Compression” presented at the GSPx conference, October 24, 2005.

[10] D. Čabrić, S. M. Mishra, D. Willkomm, R. Brodersen, A. Wolisz, "A Cognitive Radio Approach for Usage of Virtual Unlicensed Spectrum", 14th IST Mobile and Wireless Communications Summit, June 2005

[11] D. Čabrić, Shridhar Mubaraq Mishra, Robert W. Brodersen, "Implementation Issues in Spectrum Sensing for Cognitive Radios," Asilomar Conference on Signals, Systems, and Computers, 2004.

[12] P. Droz. Physical design and implementation of BEE2: A high end reconfigurable computer. Master’s thesis, UC Berkeley, 2004.

[13] FCC. Et docket no. 02-155. Order, May 2002.

[14] J. Mitola. “The Software Radio Architecture.” IEEE Communications Magazine, Volume 33, Issue 5, May 1995, 26-27.

[15] J. Rabaey, “Wireless Sensor and Consumer Multimedia Networks - A Story of Converging Trajectories?”, Keynote Presentation, CCNC 2005, Las Vegas, January 2005, www.eecs.berkeley.edu/~jan/presentations/CCNC-Jan05.pdf

[16] R. Schiphorst, F. W. Hoeksema and C. H. Slump “A Bluetooth-enabled HiperLAN/2 receiver”

[17] A. Sahai, R. Tandra, and N. Hoven, "Opportunistic spectrum use for sensor networks: the need for local cooperation," submitted to the 2006 IPSN.