Components of a smart device and smart device interactions

Alan Davy
Telecommunications Software and Systems Group
adavy@tssg.org

Abstract

A smart space environment is characterized by the presence of "intelligence" in everyday objects (such as fridges, cars, lightening systems etc). In many cases these objects are capable of communicating in order to jointly delivery a service to the user. As the presence of intelligent in smart devices increase, the smart environment can become increasing chaotic and difficult to manage. The overhead in supporting the operation of the smart environment becomes a major cost in the overall operation of the smart environment. As yet there is no standardized architectural model capable of modelling the smart environment. In order to model this environment one must understand what this environment is composed of. This paper will attempt to identify what a smart device is composed of. This will be done from two points of view. 1) Physical, being what are the minimal physical requirements a smart device must contain in order to take part with in a smart space. 2) Logical, being what does a smart devices bring to a smart space and how will these smart devices interact at a logical level with other smart devices and services. With such a model, one can begin to identify common ways in which smart devices may be used, controlled and maintained within a working and sustainable smart space environment.

1. Introduction

At present there is a lot of research into the development of smart spaces and smart device control and operation such an MIT's Oxygen[1] and NIST Laboratories Smart Space[2]. These projects use present available technology along with specialised equipment and software for smart device interoperability such as Oxygen's E21 devices and NIST Laboratories Smart Flow System (NSFS). The aim of this paper is to outline some of the problems that will be faced in the construction of a device manufacturer independent smart space. More precisely addressing the issue of developing a generic smart device model for smart device interaction. With the aid of such a model manufacturers would be able to identify the requirements to which their devices must adhere to if they wish these devices to be able to take part within a smart space.

At present there are various standards bodies that deal with physical characteristics of devices, and specify models for manufacturers to follow in order to guarantee interoperability of their equipment with other manufacturers. The Distributed Management Task Force (DMTF) has been developing a model for computer components for a while called the Common Information Model (CIM) [DMTF, 2003].

[1] oxygen.lcs.mit.edu/
[2] www.nist.gov/smartspace/

This model is very complex and detailed with regards to describing the physical components of the devices. In regards to a smart space device, such a detailed model is not required, but is a very good candidate to be based upon. In (Section 2) I will outline the minimum physical components a smart device will require in order to function within a smart space. A model will be outlined based of a high level DMTF CIM model.

Apart from a physical model of a smart device, there is also a need for a logical model for a smart device. Such a model must outline what a smart device offers to a smart space with regards to the services it can provide to the environment. The model must also outline interactions between smart devices, and smart services within a smart space. There are various models present today that have similar approaches to modelling device. Two such models would include Home Plug and Play (HPnP), which is slightly out of date but still applicable, and the newer Universal Plug and Play (UPnP), which is an open standards body. These two standards bodies have modelled devices by the services that they offer and have also developed interaction models for device communication. Another emerging standard for defining services in an abstract way is with the use of the Web Services Definition Language (WSDL). In (Section 3) I will outline a logical smart device model, with a view on present models involving device description and service description.

The reason for deriving such a model is for the benefit of manufacturers, service providers and end users. Manufacturers will have a model to which they can follow with regards to their devices taking part within a smart space environment. The services and interactions of these smart devices must adhere to present and future service composition architectures, making the job of the service providers easier to offer services to the smart space, which can interoperate with, smart device services.

Section 4 outlines an implementation independent model of a smart device and smart device interactions within a smart space. Section 5 will outline possible future work with regards to implementation.

2. Physical Components of a Smart Device

There are several organisations that publish standards to which manufacturers must follow if their products are to function with the desired capabilities. One such organisation is the DMFT[3], which created the specification called the Common Information Model (CIM). This is an object oriented information model, which defines a conceptual framework for describing management data [DMTF, 2003]. In this model the DMTF describe a schema to represent the physical and logical representation of various devices. The concept of this schema is a good candidate to follow with respect to describing a smart device with two models, one being physical requirements and the other logical requirements. The DMTF model is very detailed and as the model I wish to develop is aimed to be implementation independent I will steer clear of the physical details and keep them to a higher level of abstraction.

An intelligent device is any type of equipment, instrument, or machine that has its own computing capability [whatis.com, 2003]. On to of this definition is a smart device, capable of communicating with other devices within the environment. It can perform intelligent operations on its own behave with respect to its functionality and its relevant

[3] Distributed Management Task Force

surrounding environment. Embedded systems are being places in a wide range of products from microwaves to car keys. These devices must have a minimum set of physical components to be categorised as smart devices. These components are as follows,

2.1 Power component

A power component is any source of power being provided to a device. This may be provided in a variety of ways, such as a mains power supply, battery, solar etc. It may be a one-time battery charge or a replenishable supply of power scavenged from the environment. A power component has the responsibility of providing all electrical (or otherwise) components of a device with sufficient power to operate within reasonable parameters. In the DMFT – CIM model of a device, a power component may include sub components in addition to aid its function such as a controller for an uninterrupted power supply or power saving components. A power component must be aware of the energy demands of the device and be able to operate the device within normal working parameters.

2.2 Memory component

A smart device is able to made intelligent decisions on its own behave with respect to its environment. Almost all embedded systems contain internal memory to store operations. The reason this is such an important component with regards to the smart device is that the requirement of memory will increase with the complexity of the operations being performed by the smart device. With the ever-increasing miniaturisation of computer components, I have no doubt that this will not be a major factor in the evolution of smart device production. In the DMTF – CIM model, the memory component is split up into processor memory, cache, volatile and non-volatile memory. It takes a logical view of the physical components being independent from any implementation.

2.3 Processing component

The need for an adequate processing component is evident. As intelligence in devices increases so does the requirement for their operations to execute faster and more efficiently. It can be seen for example in the mobile phone industry the increasing demand on manufacturers to supply mobile phones with the capability of running operating systems such as [Symbian OS, 2003] or Java virtual machines for running [J2ME, 2003].

2.4 Communications interface

This component lets a device to (at the least) communicate with other devices and service within its smart space. This is an important component because if a device is to be able to interact with other devices within its smart space and let other devices and services interact with it; it must provide a means of communication to these other devices. There are a variety of physical communications that may be used ranging from wired digital and analog connections such as Ethernet, IEEE 1394, USB (Universal Serial Bus), ProfiBus, serial and parallel cables, etc. to wireless connections including IEEE 802.11*, Bluetooth, IrDA, GSM, GPRS, transceivers, etc. In the DMFT – CIM model, there is no standard communications interface component, but the bases of the smart device model is that any communications interface is possible. This model is independent of how devices communicate (physical links), but does reinforce the fact that there must be a means of device access by the environment if a device is to interact with the smart space environment.

3. Present Device Service and Interaction Methods

Apart from a physical model of a smart device, there is also a need for a logical model for a smart device. Such a model must outline what a smart device offers to a smart space with regards to the services it can provide to the environment. The model must also outline interactions between smart devices, changes in the state of smart device operation, and smart services within a smart space. There are various models present today that have similar approaches to modelling device. Two such models would include Home Plug and Play (HPnP), which is slightly out of date but still applicable, and the newer Universal Plug and Play (UPnP), which is an open standards body. These two standards bodies have modelled devices by the services that they offer and have also developed interaction models for device communication. Another emerging standard for defining services in an abstract way is with the use of the Web Services Definition Language (WSDL). Along with describing services a smart device can provide, there must also be a way of representing changes in states of smart devices and how devices react to these changes within a smart environment. On this subject I will review the concept of Reflection programming and how it may be applied to smart device interactions within a smart space.

3.1 HPnP device interactions

The Home Plug'n'Play architecture describes devices being composed of three hierarchical layers, namely objects, contexts and devices.

Object – A term used to define a single control function within a context. For example, an Audio context contains a Gain Object**.**

Context - A group of one or more objects representing a common device function. Several of these contexts may be present in a single device.

Device - A mechanism that exposes state and control variables through a home network using the Common Application Language (CAL) protocol. Devices might be stand-alone hardware devices or might be implemented in software on a PC. A device is a container for a set of contexts that collectively receive messages addressed to the same transport layer address. This address may be a unique system or unit address, one of many group addresses, or a broadcast address. **Contexts** receive messages from other **devices** to invoke actions on **objects**, which are contained within a **context**. This hierarchical model describes a good logical layout for how smart devices may offer services and interact with each other. Another method of devices description and interaction can be seen in a newer model developed by the Universal Plug and Play Group [UPnP, 2000].

3.2 UPnP device interaction

Another more generic model that is aimed at total device interoperability is the UPnP model. UPnP is a lot newer than HPnP, first being developed by Microsoft to be an extension to the concept of Plug'n'Play technology in desktop computer systems. UPnP describes three components within its model, **the device**, **the service** and **the control point** [UPnP, 2000].

A device is a container of services and embedded devices, similar to that of the HPnP model. Each device contains an XML document describing it and what services it has to offer. Information would include device name, model, serial number, etc, and a list of available services.

A service exposes actions and models its state with state variables. For instance, a clock service could be modelled as having a state variable, current_time, which defines the state of the clock, and two actions, set_time and get_time, which allow you to control the service. Similar to the device description, this information is part of an XML service description standardized by the UPnP forum. A pointer (URL) to these service descriptions is contained within the device description document. Devices may contain multiple services. A service in a UPnP device consists of a state table, a control server and an event server. The state table models the state of the service through state variables and updates them when the state changes. The control server receives action requests (such as set_time), executes them, and updates the state table and returns responses. The event server publishes events to interested subscribers anytime the state of the service changes. For instance, the fire alarm service would send an event to interested subscribers when its state changes to "ringing."

A control point in the UPnP network is a controller capable of discovering and controlling other devices. After discovery, a control point could:

- Retrieve the device description and get a list of associated services.
- Retrieve service descriptions for interesting services.
- Invoke actions to control the service.
- Subscribe to the service's event source. Anytime the state of the service changes, the event server will send an event to the control point.
  It is expected that devices will incorporate control point functionality (and vice-versa) to enable true peer-to-peer networking.

3.3 Web Services Description Language (WSDL)

WSDL is a rapidly emerging technology for describing services in an abstract way. It provides a way of offering a service, regardless of its implementation to other heterogeneous systems over the Internet [WSDL, 2003]. WSDL lets a client talk to a service through a common protocol known as Simple Object Access Protocol (SOAP). WSDL and SOAP are XML based, so as long as the client and server can parse and understand the XML documents, the implementation of the client and server can be completely independent.

A WSDL document is made up of 6 major elements

Type: - which provides data type definitions used to describe the messages exchanges between endpoints.

Message: - which represents an abstract definition of the data being transmitted. A message consists of logical parts, each of which is associated with a definition within some type system.

PortType: - which is a set of abstract operations. Each operation refers to an input message and output messages.

Binding: - which specifies concrete protocol and data format specifications for the operations and messages defined by a particular portType.

Port: - which specifies an address for a binding, thus defining a single communication endpoint.

Service: - which is used to aggregate a set of related ports.

Using these 6 major elements it is possible to describe homogeneous interactions between heterogeneous services. As manufacturers of devices develop the services of these devices (at most) in house, offering these heterogeneous services to an environment in a homogeneous way would not be possible without a level of service abstraction. The concept of using WSDL to hide heterogeneity of service implementations is very applicable to smart spaces and the interoperability of smart devices.

3.4 Reflection

A Smart Space is a physical space rich in devices and software services that is capable of interacting with people, the physical environment and external networked services [M-Zones, 2003]. In line with this paradigm smart devices are pieces of physical equipment that offer services and are capable of interacting with the user and each other. Smart devices must be able to react to changes within the environment with respect to the user and other devices and services within the smart space. As this is a major feature of the smart space, it is important that a smart device can reflect its present state of operation within the smart space. A smart device with reflective capabilities provides a representation of its own behaviour, which is amenable to inspection and adaptation, and is *causally connected* to the underlying behaviour it describes. "Causally-connected" means that changes made to the self-representation are immediately mirrored in the underlying system's actual state and behaviour, and vice-versa. [Geoff Coulson, 2003] This Meta representation of a device's state of operation must also be visible to all other devices and services within the smart space, if they are to be able to react to changes in this Meta representation. A simple example where a reflective state of a smart device may be required would be in the case of an alarm clock. When the alarm clock reaches a certain time it will start ringing. Other devices within the smart space may wish to perform some operation when the state of operation of the alarm clock changes, such as a kettle switching itself on.

There are other important considerations that must be reflected by devices, such as power levels, and bandwidth consumption. [Licia Capra et al, 2001] put forward the idea of having an application profile for each application on a mobile device, defining a specific reaction to chances in the reflective state of the resources of a device. This application profile is an XML document defining a condition and reaction to resource states of devices.

```
<RESOURCE name="battery">
        <STATUS operator="lessEqual" value=x/>          % context configuration
        <BEHAVIOUR policy="disconnect"/>                % policy
</RESOURCE>
```

The above describes the reaction of an application when battery power is low.

4 Smart Device Services and Interactions

The above section has outlined present methods of modelling the description and services a device can offer to a network, and how reflection can be used to allow inspection and adaptation of the state of operation of a device on a network. The following section will discuss how such methods can apply to a smart device and produce a logical smart device model for representing smart device services and interactions.

4.1 Services Component

The service component of a smart device is made up of various internal components. Together these components allow a smart device to **publish** its services to a smart space on entering, allow other smart devices and services to understand and **invoke** these services in an abstract way. The services component allows other smart devices and services within the smart space to access the **internal operations** of a device without the requirement of manufacture specific controls or drivers. This is done by using an abstract services definition to describe to other smart devices and services within the smart space, how to invoke the smart devices operations. Through this abstract interface a smart device's internal operations can be invoked.

4.2 Reflective State Component

A reflective state component of a smart device can be taught of a Meta representation of the devices own internal states. The reflective state component contains various internal components that allow the smart device to publish a reflective Meta representation (RMR) of itself so other smart devices and services may be able to see the present state of the smart device and react to changes. The main internal component of the reflective state component is the internal states component. This component will hold values on various states the smart device can take on, e.g. POWER ON/OFF/LOW, ALARM RINGING, PRESENT_TIME, TEMPERATURE, etc. The internal states may be manipulates by the invocation of an internal operation of the smart device, or may be reflecting the status of a physical component such as power supply or bandwidth. Smart devices and services can subscribe the RMR of a smart device and manipulate it, through what is called the **Control Component.** Manipulation of the RMR will change the internal operation of a smart device. For example an alarm clock may have a RMR showing that the alarm functionality is turned off. If a Control Component of a smart device or service were to set this attribute to on, the necessary internal operations of the smart device would be invoked to reflect this change.

4.3 Control Component

If a smart device wishes to participate in invoking a smart device service, or subscribe to state change notifications, it must contain a **Control component**. The control component effectively acts as a client of the services smart devices offer to the smart space. The control component contains various internal components, which allow it to interact with smart devices.

4.3.1 Service Publication Listener

This component listens out for any new services being introduced by a smart device entering a smart space. This component also would handle such things as leasing and deregistering of smart device services.

4.3.2 Query Services

This component can query its registry of available services for a service of a particular type. It can also query the registry of other control points or broadcast a search query to available smart devices within the smart space.

4.3.3 Invoke Services

The Control component can invoke services belonging to other smart devices within the smart space by binding with them through this component. An example of how this component may work can be seen in WSDL web service binding through the use of SOAP and URLs.

4.3.4 Subscribe RMR (reflective meta representation)

When a device enters the smart space, it can publish its reflective meta representation to any interested smart devices and services containing a Control Component. The Control Component uses this internal component to listen and subscribe to the smart devices RMR publications. This component can also request an RMR of a smart device. The Control Component will then store a list of the RMRs it's subscribed to in the **Available RMR Subscriptions** component.

4.3.5 Manipulate RMR

A Control Component can manipulate the RMRs it is subscribed to. Through this process the change in an RMR value will be sent to the corresponding smart device, which will then reflect this change through the use of internal operations. This operation also requires a binding between the smart device and the Control Component.

4.3.6 Available Services List

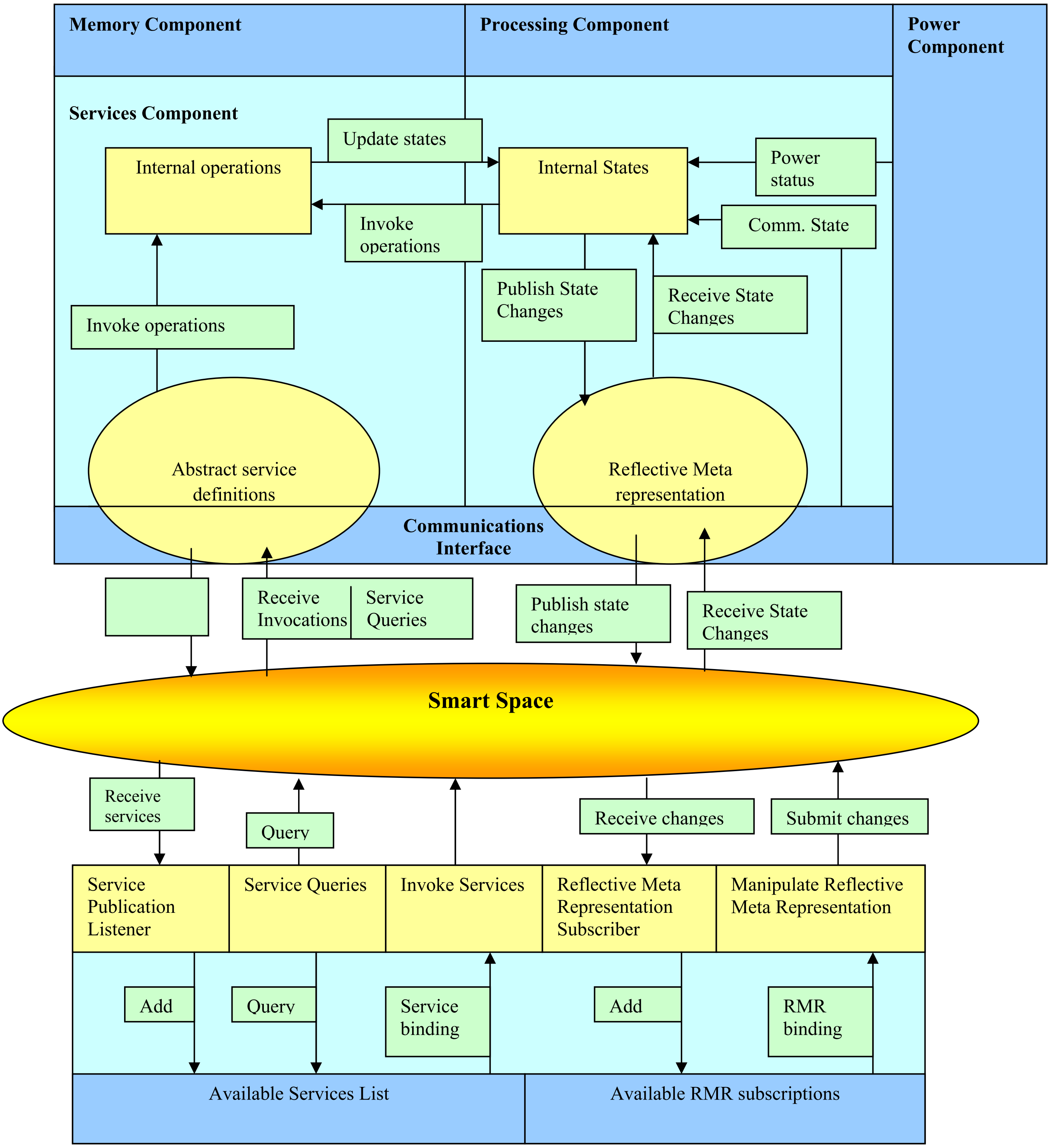
The Control Component must hold a list of all smart device services available to it within the smart space. This list can be queried for services of a certain category, returning information on how these services are to be invoked, i.e. parameters, location, device, etc.

4.3.7 Available RMR Subscriptions

The Control Point must hold a list of smart device RMRs it is currently subscribed to. This component holds information on how to update any given RMR of a smart device.

4.4 Smart Device component and interaction model

The following figure is a model of the components smart device and smart device interactions based on this paper.



5. Future Work

The intent of my future work is to develop software emulations of devices conforming to the above model programmed in Java. This implementation will be IP based, using Universal Plug and Play as a means of device and service discovery. Services description and reflective Meta representations will be implemented through the use of the Web Services Description Language (WSDL). The implementation will be heavily based on XML, using the Simple Object Access Protocol (SOAP) as the message-passing medium. Various use case scenarios will be developed to test the model for efficiency, scalability, and stability. Throughout the implementation and testing phase this model will be refined or expanded wherever necessary.

6. Conclusion

In this paper I have discussed the main components required by a smart device for it to interact with other devices and services within a smart space. This model enables devices of a heterogeneous nature to join any communications interface and be controlled by other devices without the need for driver installation. The fact that a device knows information about itself, its capabilities and how it is used enables heterogeneous devices to control and be controlled by each other and services from within and outside a smart space. It is apparent that with the use of self-aware smart devices in a smart space new devices and services may be added and removed at will without the worry of interoperability. With the use of such architecture, manufacturers can provide access to control of their products in a standard way. A universal way of controlling and using embedded systems, sensor networks, house hold appliances, network services etc. Independent manufacturers can develop products that can seamlessly interoperate with other products available on the market.

7. References

A. Friday, N. Davies and E. Catterall, *Supporting service discovery, querying and interaction in ubiquitous computing environments*, 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access, May 20, 2001, Santa Barbara, CA USA.

DAML-S http://www.daml.org/services/ 2003

Distributed Management Task Force Common Information Model specification 2.7 http://www.dmtf.org/standards/documents/CIM/CIM_Schema27/CIM_Device27-Final.pdf Visited 2003-05-30

Geoff Coulson, What is Reflective Middleware? http://dsonline.computer.org/middleware/RMarticle1.htm visited on 2003-06-17

Gary Hoffman et al. "*IEEE 1394: A ubiquitous bus*", 40th IEEE Computer Society

International Conference (COMPCON'95) 1995

HomePlug www.homeplug.org , visited 17-04-2003

HomePNA www.homepna.org, visited 18-04-2003

Home Audio/Video Interoperability
Specification of the Home Audio/Video Interoperability (HAVi) Architecture, http://www.havi.org/techinfo/docs/release-May15-HAVi1.1(clean).pdf
15-May-2001

J2ME Java 2 Micro Edition http://java.sun.com/j2me/
Visited 2003-05-19

LonWorks www.echelon.com visited 19-04-2003

Riku Mettala, Bluetooth Protocol Architecture,
https://www.bluetooth.org/foundry/sitecontent/document/whitepapers_presentations 1999

Salutation www.salutation.org visited 19-04-2003

Symbian OS http://www.symbian.com/
Visited 2003-06-09

UPnP Universal Plug and Play www.upnp.org
Visited 22-04-2003

WSDL Web Services Description Language http://www.w3.org/TR/wsdl
Visited 2003-05-23