Information Appliances
&
Infrastructure for smart devices

Nils Månsson, 14.06.2000, ETH Zürich

1 Introduction

The meaning of the term information appliance is not obvious. Are all electronic devices information appliances? Do they have to be small? Do they have to be portable? Do they even have to be visible? Is a laptop an information appliance? The answer to all these questions is NO, at least according to Donald A. Norman in the book "The invisible computer" [1]. His definition of and his discussion about information appliances will be covered in the first part. To make all these heterogeneous devices to interact there is need for some Middleware, a common software layer accommodated above existing operation systems. Today there is middleware such as COM/DCOM and CORBA for statically configured networks and computers without severe resource limitations. The information appliances and the ubicomp environment have totally new properties and requirements and therefore new solutions are required. The vision is an environment where all devices can communicate freely and easily with each other. Unfortunately the way to reach the vision, is probably longer than Route 66. Today there are among others three major competing commercial middlewares supported by big American companies and one interesting research project at MIT Media Lab, all providing different solutions. Requirements of middleware in an ubicomp environment and the four mentioned approaches will be covered in the second part.

2 Information Appliances

To get a first clue what an information appliance is, it could be helpful to have a look at a dictionary:

Appliance

A device or instrument designed to perform a specific function, especially an electronically device, such as a toaster, for household use. See synonym at tool.

Information appliance

An appliance specializing in information: knowledge, facts, graphics, images, video and sound. An information appliance is designed to perform a specific activity, such as music, photography or writing. A distinguishing feature of information appliances is the ability to share information among themselves.

From the dictionary the conclusion can be made that an information appliance should be specialized in information, has a specific function and be able to communicate with other information appliances.
It should also have a low maintenance overhead, and perhaps most important, it should be easy to use. The devices should fit the task so well that they become a part of the task.
But what's about the size, the portability and the visibility? The devices will in most cases be small and also portable and visible, but it is not required.

2.1 Three axioms

Norman has defined three axioms for information appliances. They are:

- Simplicity
- Versatility
- Pleasurably

The first axiom, simplicity, is the major driving force for appliances: to break the complexity barrier.
The second axiom, versatility, comes from experiences with the PC and the power of its ability to perform previously unknown, novel interactions and combinations of many individual activities. The information appliances can only provide this power if they are able to communicate freely and effortlessly with each other.
The third axiom, pleasurability, says that an information appliance should give pleasure to our lives. The tasks should be fun to do.

2.2 The Vision

The most important goal for information appliances is to overcome complexity and introduce simplicity. The user interfaces will often be invisible to all but the most advanced user and one should not be aware of that it's a computer one are using. Above all, you should not be concerned about file formats, configuration and connectivity. The vision are ubiquitous devices that are easy to use and are a natural part of our lives. To implement this vision the design must be human-centred rather than technology-centred, the devices must be at low cost to the consumers and there must be a common infrastructure for free sharing of information. One of the benefits with free sharing of information is the power of serendipitous flexibility, the flexibility the PC has to combine different outputs of a diverse set of applications, allowing creativity to reign. To give the information appliances this power, the information sharing must be free, easy and unimpaired by technical limitations and trade restrictions. Another benefit is the avoidance of monopolistic control. Today's situation with the PCs where its possible to share information freely among all programs as long as they are made by the dominant company, and where they can change their dominant standards at any moment, which is not desirable for information appliances. Avoiding these problems requires free, open standards, controlled by an international standards committee, available to all.

2.3 Examples of information appliances

- Some appliances have been with us for some time and often we probably don't think of them as information appliances. Keyboards, drum machines, synthesizers, MIDI, digital voltmeters, school calculators, fax machines and all kinds of telephones are examples of these devices. They all perform a specific activity, are specialized in information and are able to communicate.

- The Internet appliance already exists, but is it really an information appliance? In one way it is, but in another the Internet will be so natural and commonplace that it will be unnecessary to call something "an Internet appliance". Besides this it only names the technology.

- The entire concept of a "camera" will change; a digital camera will be like an image-gathering device. As input it can, except for taking an ordinary picture, take a scanned picture, extract a picture from the Internet, purchase a picture like a postcard in a kiosk or copy a picture from another camera. For output there are even more possibilities. Send a picture to a storage device, to a printer, or to another camera. Show it on a TV or on an electronic photo viewer. Publish photos on a website or send them to a photo kiosk and have them developed to traditional pictures.

- The digital pen captures simultaneously information electronically as well as physically. The first commercial product comes from a company called Anoto [2] in Lund in Sweden and uses Bluetooth technology for communication. The pen has to be used on a special paper with a pattern on which the pen can orientate. One of the first things printed on this special paper will be a calendar, which will easily synchronize a calendar on a mobile phone or on a PC. There will also be paper with special fields for emails and sms-messages, which can be automatically sent. Ordinary notes will be saved as pictures and will thereby unfortunately not be searchable.

- Intelligent reference guides already exist in cars, displaying a map of the locale to the driver, along with directions for driving to any desired location. Enhanced maps can show facilities as restaurants, hotels and scenic attractions along the route. There will also be portable devices for example for hikes in the mountain, where it’s possible to see location and elevation, and for stores, where the floor plan and daily sales are displayed.

- When medical sensors are cheap enough, rugged and reliable for homes, many possibilities occur [3]. Then there will be devices for blood tests, blood pressure, pulse, body temperature, urine analyses and weight, and perhaps even for pulmonary assessment and EKG. The results from all these devices could be used as input to an automated electronic version of today’s medical books for home use. It should be able to ask additional questions and provide diagnosis and then recommend actions, for example to contact a doctor.

- As mentioned above, information appliances neither have to be portable or visible, in fact they can be embedded in things.

  - Devices embedded in airplane seats make it possible to see the flight route and position, to read news, play games and listen to music.

  - Chips embedded in cars make it possible to pass the prepaid tollgate without stopping and to fuel without using cash or credit card.

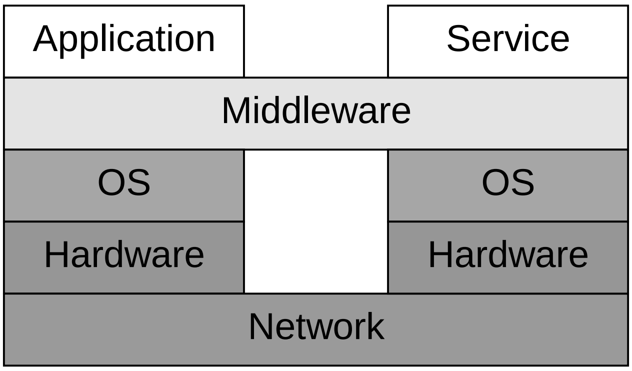
  - Whiteboards will be able to send notes directly to different devices.

  - There will be embedded devices for power generation and radio transmission in shoes, for body sensors in clothes and for display and camera in eyeglasses.

  - It’s even possible that next, after the pacemaker, we’ll se embedded hearing aids, telephones and credit cards in our bodies.

3 Middleware

All existing information appliances are limited in their communication within the platform they are using and communicate only in a restricted manner. To make it possible for all to communicate with each other and to bridge heterogeneities there will be need for so called middleware. Middleware is a common software layer accommodated above the respectively operation systems. Middleware additionally allows for faster development and provides solutions for common tasks.



The properties and the requirements on middleware in an ubicomp environment are different from those in statically configured networks, where for example COM/DCOM and CORBA are used. Therefore there is need for new middleware suited for the new conditions. Middleware in an ubicomp environment has to consider the following issues[4]:

- **Limited resources.**
  Information appliances often have limited computational speed, memory, bandwidth and power.

- **Scalability.**
  An information appliance should work directly with another device as well as in a network and perhaps even in the Internet.

- **Automatic configuration.**
  The user shouldn't be concerned about the configuration of the devices. Sometimes it will not even be possible, since there will be no interfaces.

- **Resource discovery.**
  The devices shall fastly and automatically find other devices or networks they want to communicate with.

- **Security**
  In an ubicomp environment where all devices *can* communicate with each other, we want to restrict which devices we *want* to communicate with. When a connection exists we want to set permissions for services within the devices.

- **Intermittent connectivity**
  For reasons like cost, power and bandwidth we want to disconnect, and for reasons like latency, congestion and limited range its possible that the connection unintentionally is cut off.

3.1 Examples of middleware for ubicomp

There are today three major alternatives for middleware for ubicomp environment.

- **Jini**, which was first launched, is totally Java-based and supported by Sun,
- **Hive**, which is a research project at the MIT Media Lab,
- **Universal-Plug-n-Play**, UPnP, which is Microsoft's answer to Jini and an extension of Plug-n-Play, and
- **Chai**, which is HPs answer to Jini and is a closely related to UPnP.

These are certainly not the only suggestions for ubicomp middleware, but they are the most popular ones. Below follows an overview of these four middlewares according to the considerations mentioned above. Since all products are very new and some not even completed, most information was taken from the products web sites. It is likely that information found on sites from research projects is considerably more objective than sites from commercial products.

3.1.1 Jini

Jini is a product from Sun totally based on Java. It is operating system independent and uses code mobility [5,6].

- **Limited resources.**
  According to Sun all devices that can pass digital information in and out can participate in a Jini-network. For example PDAs, cameras, VCRs, TVs, CDs, and phones can participate. If a device is not using Java it requires a Java-wrapper to the code that announces the service to the lookup service (LUS). The only demand on the network is that there is at least one Java Virtual Machine (JVM), which can execute code on behalf of devices without their own JWM.

- **Scalability.**
  According to Sun Jini is powerful enough to build a fully distributed system on a network of workstations and small enough to build a community of the simplest devices such as home entertainment devices and cellular phones. There should also be no limitations on the number of devices and services participating.

- **Automatic configuration.**
  No prior knowledge of devices, no configuration and no driver installing should be necessary.

- **Resource discovery.**
  When a device wants to connect to a network it first uses a discovery protocol to locate the LUS. Then it registers itself with the LUS and uploads a proxy. When a device then wants to use a service on the network it downloads a proxy from the LUS and then interacts directly with the selected device.

- **Security**
  Jini uses an extended version of the Java 2 security model. An access control list associated with every proxy decides whether access is allowed or not.

- **Intermittent connectivity**
  When a device registers itself with the LUS it receives a so-called “lease” which has to be periodically renewed. If it’s not renewed the device is removed from the list of services offered.

3.1.2 Hive

Hive is a research project from MIT Media Lab, which like Jini is totally Java-based and uses code mobility. In opposite to Jini, Hive is using agents. After existing some years internally to connect TTT (Things that think, smart devices developed at MIT) [8], Hive is now in a phase moving outside the Media Lab [7].

- **Limited resources.**
  Hive requires all devices to have their own JVM. The smallest device where Hive is used is an Intel 486 processor running Linux.

- **Scalability.**
  Hive is a totally decentralized system. When a device looks for a service, all participating agents must be asked sequentially, which is very inefficient in a large network.

- **Automatic configuration.**
  All devices must be well known.

- **Resource discovery.**
  As mentioned under scalability all agents must be asked sequentially if they provide a specific service. Questions can be given both “syntactic”, using the Java-classes, when you ask for a service or a device and “semantic”, using XML, when you ask for an objects place in the world.

- **Security**.
  Three cases can be distinguished; protect host from agent, protect agent from agent and protect agent from host. None of these cases are fully solved. In the case “protect host from agents”, agents permissions to get into a cell are temporarily solved using passwords, but there is not yet any permission policy to control permissions once an agent is inside the cell.
  Hive is today implemented in Java 1 but will move to Java 2 and its extended security model.

- **Intermittent connectivity**
  When an agent fails it looses its connection.

3.1.3 UPnP

UPnP is Microsoft’s answer to Jini and is an extension of PnP to support networks, peer-to-peer discovery and configuration. It’s mend to be independent of operating system, programming language and physical medium and in difference to Jive and Hive it is using standard web communication like HTTP, URLs and TCP/IP instead of code mobility. An engineer, who asked not to be named, says in ZD Net News [9] that if UPnP wants to provide the same functionality as Jini it has to either use code mobility or has to provide a common set of services (Windows) across all machines. UPnP is planned to be launched sometime in the autumn 2000 [10].

- **Limited resources.**
  The first device that will use UPnP is a PC running Windows Millennium. Microsoft claims however, that also small devices that for various reasons do not run UPnP, can be connected by proxies exposing them to their UPnP-peer.

- **Scalability.**
  UPnP is said to be small enough for the smallest application and powerful enough to scale to the Internet.

- **Automatic configuration.**
  Neither configuration nor installing device drivers should be necessary.

- **Resource discovery.**
  A device will dynamically be able to join a network, obtain an IP address, announce its name, convey its capabilities upon request and learn about the presence and capabilities of other devices. This was until recently handled by the Simple Service Discovery Protocol (SSDP) which suffered from scalability problems. In SSDP the client sends an UDP multicast with an identifier of the desired service, and the services respond if they provide

the requested service. To scale this system to the Internet Microsoft recently implemented a service directory where devices can register. If a service directory exists in the network it is used, otherwise the discovery is provided by the SSDP.
When a device is found, it uses XML to expose its semantics.

- **Security**.
  Microsoft just mentions some vague words about web authoring and server technology.

- **Intermittent connectivity**
  A device shall be able to leave automatically without leaving any unwanted state behind.

3.1.4 Chai

Chai is HPs answer to Jini and is since recently using UPnP to communicate. Chai has its own JVM implemented in C++ and assembly language [11].

- **Limited resources.**
  The memory footprint of Chai starts at 228 KB, which according to HP is a fragment of the size of competing products. Only the libraries required by the device are loaded.

- **Scalability.**
  Chai is built from off-the-shelf components, which all work together and make code easy to reuse. If you want to add a new functionality, you only have to add a new component to the existing code.

- **Automatic configuration.**
  See UPnP

- **Resource discovery.**
  See UPnP

- **Security**.
  See UPnP. Chai is also using the Java 2 security model.

- **Intermittent connectivity**
  See UPnP

4 Conclusions

An information appliance is an information-handling device with a specific function that can communicate with its environment. They don't *have* to be small, portable and visible, but they will in most cases be.
We will for sure see more and more information appliances around us, but how powerful and successful they will be depends on the middleware. Today they are communicating with device-specific, limited ad-hoc solutions.
The development of ubicomp middleware seems to be harder than the developers had thought. No product is developed with wireless communication in mind and none satisfies the demands of personalization of the devices.
There are today two major approaches; The Jini and Hive approach using Java and code mobility and the UPnP and Chai approach using web communication. To reach the vision of a worldwide agreement on an appropriate infrastructure were we can get the full power and flexibility of the information appliances will not be easy. Instead we will in a foreseeable future most probably see several products compete against each other.

5 References

[1] Donald A. Norman: The invisible computer. Cambridge, MA: MIT Press, 1998
[2]http://www.anoto.com
[3] http://www.businessweek.com/reprints/00-10/b3671021.htm
[4] http://www.cs.washington.edu/research/portolano/
[5] http://www.sun.com/jini/
[6] http://www.jini.org/
[7] http://hive.www.media.mit.edu/projects/hive/
[8] http://www.media.mit.edu/ttt/
[9] http://205.181.112.101/sp/stories/infopack/0,,2239858,00.html
[10] http://www.upnp.org
[11] http://www.internetsolutions.enterprise.hp.com/chai/whatschai.html