Designing an Architecture for the Digital Home from the Top Down

Shwetak N. Patel

College of Computing & GVU Center
Georgia Institute of Technology
801 Atlantic Drive, Atlanta, GA 30332-0280, USA
shwetak@cc.gatech.edu

Abstract: We present an initial design of an architectural stack for the digital home. Our design was informed through the process of designing, building, and deploying an application for the home called InLook. We also reflect on the short comings on existing infrastructures in literature for building applications for the digital home and offer some suggestions for improvement.

1 Introduction and Motivation

An ambition of ubiquitous or pervasive computing is to create services that perform some of the important yet mundane tasks that we would like to do, but do not have the patience or time to perform. One particular domain is the home. In the home we are constantly inundated with information that both enters the home and information that is generated in the home. In addition, the home is full of many chores that homeowners forget or rarely have the time to attend to. There is an obvious opportunity for technology to handle much of this information on behalf of the user and offload some of these household chores.

We have the ability *today* to build a lot of these technologies. For instance, building a HVAC filter change sensor is trivial or automating the lights in a home is a popular off the shelf thing now. Likewise we can also build software and user interfaces to run these systems. However, a problem arises when we begin to add new devices and services. Each time new services are brought into the home, a custom interface has to be developed for it to interoperate with the local infrastructure. In addition, developers have an added burden of making lots of assumptions about the available resources present in an environment.

In this paper, we present an application for the home called InLook. The building of this application had two purposes. The first was to come up with some application to whet our appetite for work in this area and to help uncover some architecture requirements.

2 Related Work

2.1 Smart Homes

There has already been work started in smart homes. Georgia Tech's The Aware Home, built in 2000, provides a place to conduct research an authentic yet experimental setting. The Aware Home serves as the living laboratory for ubiquitous computing in home life. Work in The Aware Home has included assistive technologies for the elderly, location detection and smart floors, and toolkits for context aware system. Although there have been attempts there has yet to be a clearly defined architecture for developing application in the home.

Rochester's Smart Medical Home has the overall goal to develop an integrated personal health system. Their focus is to allow consumers, in the privacy of their own homes, maintain health, detect the onset of disease, and manage disease.

Philips HomeLab (also called Ambient Intelligence) is very similar to the Intel's Digital Home effort. They envision a domestic environment where devices automatically respond to changes in context and how consumer level electronics can be integrated in the home. The canonical example is the light dimming when soft music is played or when a movie is started.

MIT's House n focuses on activity recognition and the use of ethnographic studies to help inform the design if specific domestic applications.

MSR's Easy Living project focuses on vision-based activity recognition and person tracking systems.

UT Arlington's MavHome uses an agents and prediction based architecture. Rather than employing an impressive set of sensors MavHome tries to reply more on inference and prediction for activity recognition.

The Neural Network House at the University of Colorado Boulder employs a neural network to control heating, lighting, ventilation, and water temperature. This project uses machine learning techniques to optimize the control of these devices to provide services like minimizing operating cost.

Products like X10 provide a way to fairly easily instrument a smart home, however no clear standard has emerged to integrate these technologies. They simply provide a way to automate some specific part of a home, such as turning on lights when someone enters the room or turn on the sprinklers when it rains. Each actuator typically has a specific or proprietary sensor, and no clear interconnections between various devices. However, products like X10 are the only technologies that make deploying a smart home today possible.

2.2 Infrastructures

Although we see very little work on infrastructures of the digital home, there has been a lot of work in developing infrastructure for ubiquitous and pervasive environments. Most smart homes are created evolutionarily by adding more and more new devices to an existing home, instead of being developed in single instance like building a new home from scratch. This incremental nature of technology in the home requires a flexible infrastructure to accommodate both future and legacy systems without requiring extensive reconfiguration. We identify Metaglue, Speakeasy, and iSuff as three relevant projects for the digital home that attempt to address this problem.

Metaglue is a system that allows programmer to create software agents for intelligent environments. Originally part of MIT's HAL project and now AIRE, metaglue provides architecture for distributed devices to communicate and share resources. Metaglue offers the following features:

- Configuration management
- Communication channels between agents
- Maintain agent state
- Modify and insert agents without shutting down or restarted the whole system
- Manage shared resources
- Event broadcasting

Metaglue maintains both user configured and learned state of an agent. As agents adapt their state evolves in the metaglue database. Logical connection can be formed between components, for example connecting a microphone to a display. An interesting aspect of metaglue is that when agents fail, the system automatically tries to find a similar agent nearby and create a new connection in a sort of a self healing way. Many devices like cameras and display tend to be very scarce because of their high costs and typically are shared between components. Metaglue arbitrates resources automatically to prevent conflicts and unnecessary deadlock. Finally, agents in metaglue can also broadcast event information to other agents.

Many of the basic concepts introduced by the metaglue system such as communication channels, agent state, event broadcasting, and hot swapping are very relevant to a digital home infrastructure. However, metaglue lacks how exactly new devices are discovered. It currently assumes a manual initial configuration. For a digital home we would expect devices to enter the space and automatically configure itself.

Obje (formally Speakeasy) is a system that addresses the inoperability problem between various devices. Meta-interface serves as a way to abstract out communication means and protocol negotiation. Rather than relying on a standard device manufacturers conform to a general meta-interface. During runtime these meta-interfaces handle the intricacies of the discovery and data transfer between two dissimilar devices. Interestingly meta-interfaces can dynamically negotiate new services or capabilities unforeseen by the devices.

Meta-interface example: "An example of an Obje meta-interface is the Data Transfer meta-interface. When an Obje-enabled music jukebox connects to an Obje-enabled music player, the Data Transfer meta-interface establishes a network transport mechanism based on the network?s capabilities, and also establishes a media format based on the device?s capabilities and media requirements. The jukebox then

?teaches? the network protocol and media format to the player, by sending a small software bundle to the player that enables it to receive and decode the music. Once the player has been ?taught?, the media begins to play."

iStuff is a project at Stanford's iRoom that includes a toolkit for interactive applications for physical devices and a software infrastructure to facilitate infrastructure. Although the focus is more on physical device interface many of the systems requirements they have derived are applicable to the digital home.

- flexible lightweight devices
- platform independence and cross-platform capabilities
- wireless protocol independence
- ease of integration with existing applications
- multi user support

3 Architecture Requirements

As an initial starting point, we brainstormed the high-level services and system requirements that would be required in the digital home. This exercise mainly served to give us an idea of the salient pieces of a digital home architecture.

3.1 The Services

-Discovery: How devices discover each other and new devices that may enter the home.
-Location of people and objects: Locating people and object to provide contextual information for applications
-Store and access: A central store for the applications in the house.
-Activities and availability: The status of individual in the home
-Media annotation and capture: The capture precious memories (videos, pictures, etc) and how they can be retrieved easily later.
-Communication: The different means that devices and/or people will communicate with each other.

3.2 Architecture Requirements

-House layout: At some level there needs to be some representation of the physical layout of the house. This could be manually entered or automatically generated using sensing techniques. What ever the means, there needs to be a notions of where people and devices actually reside.
-Network connection: and A fast network connection within the home and to the outside world is needed for communication of the various devices.
-User preferences and configuration (inferred and preprogrammed): At some level application must have a sense the user’s preferences.

-Annotation: The annotation of media and even physical artifacts require annotation of relevant meta-data for both indexing and future retrieval.
-Storage (index, access, and store): An easy, but yet power storage mechanism is needed for the massive amount of information that could potentially be resident in a digital home.
-Replication and backup: The architecture must provide seamless backup without little to no user intervention.
-Security and privacy of personal data: Archives information may contain personal information, so the architecture must provide a reasonable level of security.
-Discovery of semantics of devices: For devices to interoperate, the must be able to discover the services that another device offers.
-House/People model: Some application may be need higher level information of a household and the people in the house, which we call the *ecology* and the *ontology* on house.
-Service protocol (API to the devices): The architecture must facilitate easy access to the devices and services available in the home for quick and easy application development.

4 InLook

We used the design and developed of an application call InLook to help inform the design of the architectural stack for the digital home. InLook was not meant to be the killer application for the home, but rather serve as a starting point for use help discover the architecture requirements. InLook is an application that runs on a large screen display that is designed to quickly present pertinent information about the status of the home, family members, or any other domestic related information. Figure 1 show a screenshots of the InLook application. The idea behind InLook is that it is one application that a homeowner could easily install in the home and relevant devices and services are detected in the environment. Based on what is available in the home, various services are generated in the InLook application. For instance, if a network connection is found in the home a simple new ticker service is offered by InLook. Another example is if a camera is found in the entry way of home, a simple family member state service is offered.

InLook is meant to run on a large display placed in an area that family members are likely to pass often, such as an entryway. However, one could imagine running InLook lots of smaller displays placed throughout the house.

Figure 1: Screenshots of Inlook

4.1 Times of Interaction

We identify four different times of interaction with InLook in the home: the "nesting" phase, passing by, dwelling, and leaving. We used this information to help guide the type of services we wanted to build in the InLook prototype.

Nesting Phase: This constitutes the routine a person goes through when first coming home at the end of the day. The nesting period is pretty short, 15 minutes, and is typically fairly consistent. For instance, a person may way in and put their bags down near the entry way, check mail, see who else is in the house, check the answering machine, etc.

Passing By: This is when a user passes by a device or display with little or no attention. The display could possible get the user's attention by flashing some sort of image, but the user's main focus is not on the display.

Dwelling: This is when a user purposely stops at the display to view its contents. The user in this case may be curious of what the general status of the home or may want to solicit specific information.

Leaving: This is when a user passes by a device or display with little or no attention. The display could possible get the user's attention by flashing some sort of image, but the user's main focus is not on the display.

4.2 The Services

For the InLook prototype we decided on four services: a RSS ticker for news, pictures of the day, house status monitor, and a family member monitor. Figure 1 and 2 shows the four services running on InLook.

The RSS ticker provided up to the date news feed. Although we can imagine the new content customized for the individuals in the house, our prototype simply presented the top stories. The new content was updated each time a new story was received.

The pictures of the day application served as the "entertainment" service. Our vision is to provide personalized content such as movies or pictures. For instance, InLook may display pictures of a past holiday or birthday based on the current date. Since we were limited in time and the available content from our users, we decided to show interesting pictures of the day from around the world. The content was actually served from SF Gate's website.

The house state monitor is a service that alerts users about upcoming house maintenance, potential problems, and warnings. The premise behind this service is that a variety of sensors in the home feed into InLook and appropriate alerts are presented to the user based on these readings. As new sensors are brought into the home, the user should go through minimal configuration for the house monitor. For our prototype and deployment, we simulated commons household alerts such as "change your HVAC filter" or "there is a leak in the sink." Warnings were giving high priority and appear in red. In addition, to grab the user's attention, a warning message flashes on the screen when InLook detects a person is walking by or in front of the display.

4.3 The Implementation

InLook runs on a 42 inch Sony Plasma display that is driven by a regular desktop PC. The InLook application is written entirely in Java. The Java Media Framework (JMF) was used to handle media requirements, such as the status monitor camera and the dwell camera. The dwell detector and the status monitor use frame differencing to detect simple motion.

The Java RSS ticker simply made RSS requests to yahoo.com and presented it on the news display. Although we could have allowed for customized controls for the new feed, we thought that the breaking news from yahoo.com would be sufficient for your initial prototype.

The personalized media content was obtained by downloading the “pictures of the day” from sfgate.com. We used HTTrack to download the daily pictures from sfgate.com. The InLook application would just cycle through the cached pictures. The reason using HTTrack was that “pictures of the day” could not be directly linked, therefore we had to download local copies daily. To ensure no daily pictures are missed, the “picture of the day” service thread would download the latest pictures every 30 minutes.

Since we were not able to instrument our user’s house with sensors, we “faked” the house maintenance monitor. InLook had a list of preprogrammed situations and priorities. The application would randomly select alerts and display it on the screen. Each alert is selected at random times to provide some level of an authentic use.

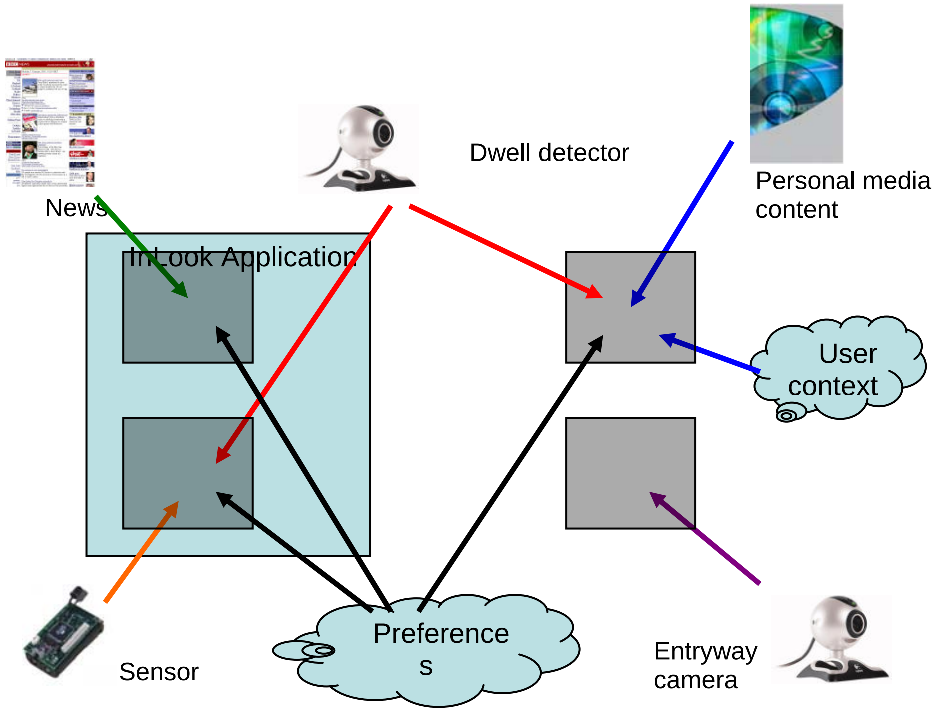

Figure 2: High level architecture and services of InLook

4.4 Reflections

Our initial implementation plan was to use existing architectures like meta-glue and speakeasy to build InLook. However, we quickly found the limitations of these systems made it difficult to some of the services we wanted. As a result, we decided to implement the application from scratch and note where the failures in the existing infrastructure were.

5 Deployment

We deployed InLook in an averaged sized (1000 sq ft) Berkeley home of a family of two. There were two goals of the deployment. The first was to get reactions on the services we built in InLook and get a sense of when and how often they were used. The other goal was to get an idea of the kinds of services that one would want beyond the four we presented.

5.1 The Setup

Figure 3 shows the placement of InLook. It was placed at the end of the of the entry way and near the living room. Since the living room was between the bedrooms and kitchen, it ensured that family member would frequently pass by the display. In addition, because the living room was the first place the family members tended to go during the nesting period, it was the idea an ideal place for InLook.

The status camera was pointed towards the entry door (figure x) and the dwell camera was mounted on top of the display facing the living room (figure x). A standard PC was use to drive InLook and a standard household cable modem provided the necessary network connectivity.

We were able to periodically check the status of InLook by using VNC. This allowed us to make any changes and ensure the application was running without any problems.

Figure 3: InLook setup

5.2 The Study

We deployed InLook for one week and then interviewed the two family members. The goal of the interview was to get an idea of the usefulness of InLook and potential improvements. Two researchers interview the two family members. The family members were encouraged to reflect on thoughts they had about the system, both positive and negative, in addition to the questions we asked.

5.3 Results

The one week deployment went surprisingly well with InLook running the full week with no major problems. Both family members had interacted with the display in some way for the entire week.

From our interview we found that InLook was an interesting and useful application. Our users were initially intrigued by the novelty of the application, but quickly learned developed their own opinions about it. The most profound result was that they characterized the large plasma as an “eye sore,” which was attributed to the large size of the plasma. They felt that a variety of smaller displays distributed around the house would be more aesthetically appealing and useful.

Based on our dwell logs we found that our users spent about 30 minutes a day inform of the display indicating that they spent quick a bit of time viewing its contents. The most popular of the four services was the pictures of the day, mainly because of it interesting content. They indicated that personalized media would be just as interesting. The news was also another popular service. The status monitor did one very important use during the deployment. The user's actually found that their landlord had entered their apartment without consent. They found this out by looking at the status monitor and saw the last two key frames showed in entering and exiting the apartment (figure 2 bottom). Although we did not envision this use, the users found this as on of the most helpful services during the week. The house maintenance monitor was the least useful of the four services. We attributed to the fact that all the data for that service was "faked" and there wan no compelling reason to use that service on the part of the user. The users agreed with this hypothesis.

The users indicated more customization, especially for the news ticker would be helpful. Although the users did not pay attention to the maintenance monitor, we asked them the kinds of things they would like with that service. They mainly viewed it as a todo or appointments list, thus they wanted a way to manually add content.

When asked if they would continue using InLook, both agreed they would. In general we found that InLook can be an useful application however it would many customizations options for users for it to be extremely useful. Also, many smaller displays seems more appropriate than a one large display.