ε

Project Proposal

Epsilon

Randy Smith, Owen Densmore
Glenn Scott, Laura Hill

Project Description

The everyday things around us are becoming digital. Devices can achieve flexibility by being small, independent, and reconfigurable components in a network fabric. We see each device as a first-class network citizen, providing a service through existing standard protocols.

In mathematics, epsilon is as thin as an interval can get without vanishing. Project Epsilon devices are as thin as they can be: Functionality and relationships between devices reside in the virtual world of networked computing for flexible and easy device configuration.

In our vision, a device requires no local administration or programming: It is nearly state free and easily field replaceable. Because state resides out on the net, an upgrade does not really improve the device, it improves the service.

Note that this vision stands in distinction to a common Sun worldview in which many or most devices host significant local computation, and thus must communicate at higher levels, downloading code, passing objects, deciphering each other's interfaces to invoke methods through, for example RMI, JINI, or JXTA protocols. Relative to that future, there is another potentially disruptive technological future in which devices remain pretty dumb with all computation off the device, in back room servers somewhere.

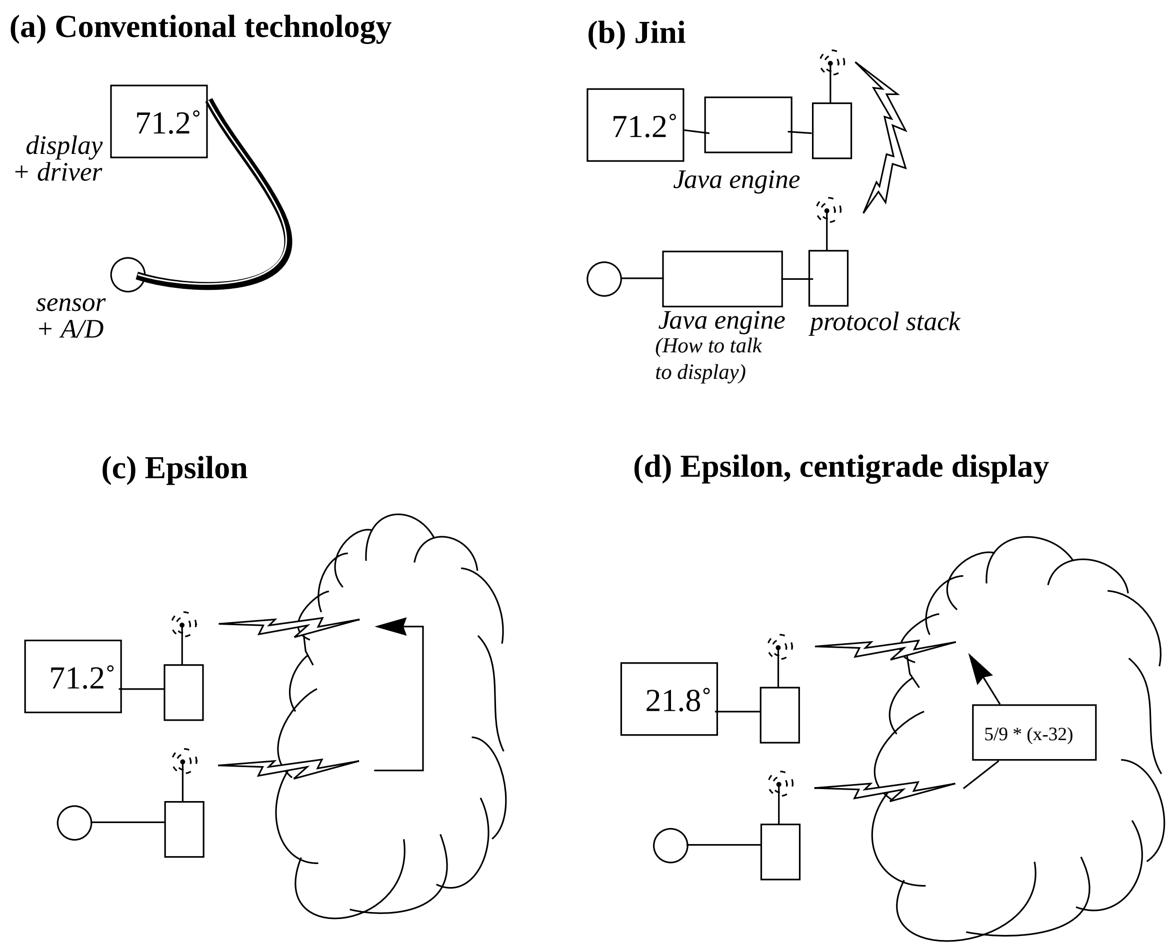
Note that in our story, the thorny issue of discovery becomes moot. Devices do not come with an agenda requiring them to find and then use other local devices. Rather, the user pieces things together defining device relationships in software to create an integrated, whole technology.

In this project all the fun will be outside the devices. How do I tell a small digital display to show the temperature output from some small sensor? We need to [1] select and implement the device communication mechanism, and [2] devise a way to specify device behaviors and relationships (i.e.: to create a "device language" for specifying the behavior and relationships among devices so that together devices can form an integrated piece of technology.)

What's in it for Sun

Move from "the network is the computer" to "the networked computer is all of human technology." Greater demand for servers.

Service the rest of the emerging edge of the network.



In conventional technology (a), relationships are hard wired. With Jini (b) local code maintains a lease on a resource after a lookup has established the relationship. Epsilon (c) devices typically need not be identified with a lease on a resource or go through a code download. Because their relationship is held in software, Epsilon devices are flexibly reconfigured.

Technical Approach

0.1] Form ultra thin device survey group. Get up to speed on previous work (e.g.: tangible interface work [Ishi et. al.], and existing phycon (physical data token) projects.)

0.2] Create a catalog of devices (periodic table of device atoms) from which any technology can be made. This includes data types from/to each device. Another catalog of “controllers” (TINI/CerfCube, etc., that resides with the device to put it on the net.)

1] Build some small number of wired and wireless Epsilon devices. A device may typically support a standard TCP/IP stack and one or a few simple common higher level protocols, such as HTTP, and/or SMTP transport, and/or MIME and/or HTML encoding. E.g.: use Dallas Semiconductor TINI boards, and/or Linux SBC (Single Board Computer) such as the Intrinsyc CerfCube. Also use 80211b Compact Flash.

2] Create a device language for dynamically specifying behavior and device relationships. This may start out as a simulation of the Epsilon world, in which the Epsilon devices are depicted on-screen.

3] Target an integration of 1] and 2] with a compelling demonstration of dynamically reconfigurable Epsilon devices.

POSSIBLE EXTENSIONS

As targets of opportunity,

A] Location - how does a device or the software world know where each device is? Physical proximity might be useful way for users to indicate relationships between devices.

B] Security - becomes more urgent as ALL devices become epsilonic. Epsilon technology does all its work over the net. How do we answer who owns a device, provide certifiable identity, and provide privacy?

C] Quality of service.

Scenarios

1 - Fred has to be in the backyard gardening but expects a delivery person to drop by, ringing the doorbell. Fred picks up his cellphone, and slides it over the ID plate next to the doorbell, “telling” it to forward the doorbell ring to the cellophane’s Epsilonic speaker.

1.1 - Fred’s front door also has a simple intercom (think apartment). Luckily Fred’s cellphone is entirely Epsilonic, being dis-Integrated into a speaker plus microphone transceiver set. Passing the cellphone by the intercom’s ID plate rewires the intercom to communicate directly to the cellphone.

2 - Karen is a programmer who likes to customize things...like her alarm clock. She normally drives to work at 7:30 to avoid the 8:00 rush hour. But she has used an on-screen visual programming system to “subscribe” her alarm clock to the traffic sensors on the three back roads she uses, which sense an earlier build up of traffic, thus causing the alarm to sound at 7:15.

2.1 - Karen’s new schedule is more flexible, thus she needs to coordinate not only with the traffic but also the time of her first meeting of the day. If the traffic lightens up a bit, she can sleep later. She uses her Yahoo Epsilonic calendar service to stream appointment updates to her Clock-Ray service, allowing interesting adjustments to her sleep schedule.

3 - GM introduces their new Epsilon Coupe (Epsi for short) for the digeratti. In addition to its Telematic device, providing land-auto communications, it has dis-Integrated the speakers, knobs, dashboard, GPS, microphone, and universal phone/pda cradle-charger. The GPS unit has no display, using instead an Epsilon Interface to the virtual dash, which integrates in a Map display. The travel planner uses the auto speakers to guide the driver, muting slightly the background music being URL web-delayed from PBS's Fresh Air. The Epsilonic cellphone also uses the speakers as well as the microphone for hands free operation. The Telematic box subscribes to the latest Epsilon Services, which have recently added the ability for the passenger to use PDA Pen Input to navigate within the Map system rather than using the driver's touch screen virtual dash access. Similarly the back seat displays are rewirable to exhibit Game, TiVo/VCR, and TV properties, all with no tuner, instead using WebServices.

4 - Mary Media has recently achieved Nurdvana with her new Bose-Epsilon system. The ScreenRay monitor is used for home theatre, TV, and Web display. She downloads into her RemoteRay controller a Graffiti input which can "drive" the TV mode via channel number or name, as well as program title. The speakers are used for the audio system, the WebConference work-at-home system, and the video systems. The 802.11 speakers are easily proxied to the phone system so that the entertainment mutes for calls.

Resources and Time line

2 or 3 people,

4 months Simulator:

Explored platform on which to build the Device Language back end. (For specifying behavior and relationships.) Tentative device language characteristics identified.

One device (hacked thermometer?) with its device language back end.

Study group completed.
Catalogs done
IP: Patent disclosures.

8 months Three devices interacting, reconfigurable.
White paper on architectural findings.

12 months Open Projects deployment / release.
6 - 9 devices actually working.
Device language done.

16 months Paper (e.g.: CHI 2003) talks.
Next phase scoped out.

20 months Compelling demo completed.

6. Tech Transfer Target

IN SUN:
Consumer and Embedded.
Collaboration Industrial Design (MPK UCD group)
Forte Tools: Device programming environment.

OUTSIDE OF SUN:
Professional Services.
(PS interfaces to appropriate customer industries)

7. Mesh with Labs Portfolio / Themes.

DeviceNets theme.
Thin Client.
Supernets may be important enabling technology.

Possible Java aspects: for example, a Java Device Bean
(an EJB of some sort?)