

The Furniture of Ubiquitous Computing

Toshiyuki Masui
Sho Hashimoto

Although various ubicomp technologies have been proposed for home environments, few people are enjoying such technologies in their daily life, due to the **lack of powerful software framework for building flexible applications for home**. We are developing simple and powerful ubicomp frameworks which can be used for building furniture-embedded networked devices which fit to home environments. Using our frameworks, **many devices can communicate** with each other by exchanging data shared on the Web server using standard **HTTP/WebSocket**. In this paper, we describe the concepts and implementations of the frameworks, and show how sparsely-connected devices can cooperatively be used for various tasks needed in home environments.

Setup Problem

Many of the research systems require **special hardware/software settings before usage**. If a user has to register network parameters for all the devices before he can use the system, he would not set the parameters again when the network configuration is changed.

Programming Problem

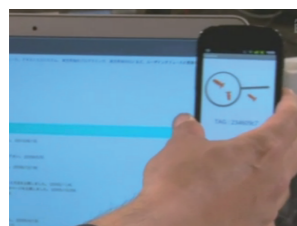
Parallel and distributed programming under unreliable communication channel is required for handling many devices at home, but such programming is difficult without a flexible and reliable parallel programming model.

Customization Problem

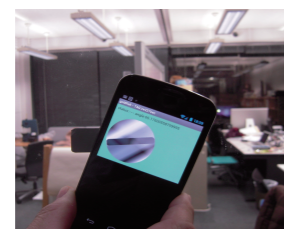
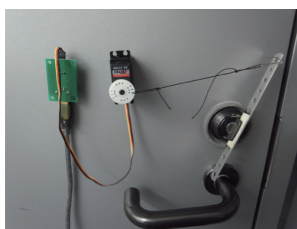
When a system is made up of tightly-coupled modules, changing behaviors and adding features is not easy. Most of the experimental systems are not designed to be flexible enough, and the whole system should be **redesigned when additional devices are required** or the configuration should be modified.

GoldFish

JavaScript framework for developing “Real-World GUI” using an Android Phone equipped with an NFC reader.



Real world copy-paste



Door unlocking

LindaBase

Web-based implementation of **Linda**.

Linda is made up of several primitives operating on data objects called “tuples” placed in the “tuple space” shared by multiple processes. A tuple is represented as an array of objects and handled from programming languages like **C, JavaScript, Ruby, etc.**

Send the value of Light sensor on Arduino using Ruby program.

```
ts = Linda::Client.new.tuplespace["myhouse"]
loop do
  light = arduino.analog_read 0
  ts.write ["sensor", "light", light]
  sleep 1
end
```

Read the Light sensor value using JavaScript on browser

```
var ts = new Linda().TupleSpace("myhouse");
ts.read(["sensor", "light"],
  function(tuple){
    alert( tuple[2] );
  }
);
```