

Cricket: A Location-Support System for Pervasive Mobile Computing 9807-04

Progress Report: January 1, 2001—June 30, 2001

Hari Balakrishnan and John Guttag

Project Overview

WIND is a system of middleware and protocols that will enable new applications in dynamic, mobile networks of devices. The main focus of WIND is in designing and implementing *context management* techniques. With the technologies developed in WIND, applications can adapt their behavior to their physical location using the Cricket location-support system, to the other services in their environment using the Intentional Naming System (INS) for resource discovery, to “vertical” mobility between a variety of wireless networks using the Migrate mobility architecture, and to changing network conditions using the Congestion Manager (CM) module.

Progress Through June 2001

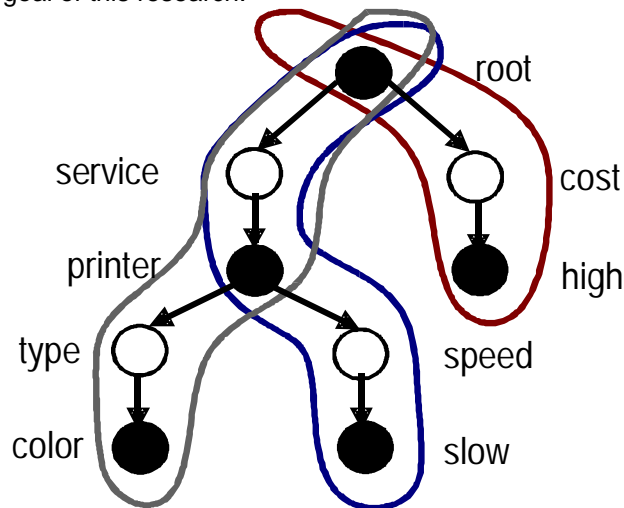
1. Stream migration architecture for pervasive computing: We have designed, implemented, and demonstrated, during the Oxygen partners meeting, the ability for live streams to follow users as they move around in a building. This uses two key Oxygen technologies: Cricket and INS, together with a modular framework for a variety of media streams. Furthermore, these technologies have also been used by other groups within MIT (and also other Oxygen partners) for new location-aware applications.

2. Design of INS/Twine (Scalable intentional resource discovery): INS is a peer-to-peer system where the set of nodes in a community collaborate with each other to provide a system-wide service. The big challenge facing both INS and other existing discovery systems (e.g., Jini, IETF's Service Location Protocol, Universal Plug-and-Play, Salutation, and various other research proposals) is that they do not scale well, or if they claim to, they make unrealistic assumptions (such as, for example, that network proximity is equivalent to geographic proximity, which is clearly not true).

In INS, the hierarchical attribute-based descriptions of services and queries form a “name-tree,” shown in the figure below. The unshaded nodes of the tree correspond to attributes, while the shaded nodes are the corresponding values. This hypothetical example describes a color printer service that prints pages slowly at high cost. Notice that the attributes “type” and “speed” are nested within the context of the printer “service”. One can see that this can be generated from a more traditional XML description that looks like this:

```
<service> printer
<type>color</type>
<speed>slow</speed>
</service>
<cost>high</cost>
```

Our new idea to achieving scalable resource discovery is to partition the set of name trees amongst the different resolvers in the network. Rather than using a static partition, which does not adapt well, we divide up each service advertisement into strands, where a strand is a path from the root to a leaf. We hash each strand to a unique value to ship the strand along the overlay network of resolvers to a node whose unique ID is closest to the hash. Essentially the same operations, with some enhancements, are used to resolve queries. Our preliminary simulations suggest a large number of ways to accomplish this basic idea, in terms of routing and query mapping strategies. These different approaches have radically different scaling properties, however. Understanding this at a fundamental level is a major goal of this research.



Research Plan for the Next Six Months

1. Implementation and experiments with Twine/INS for scalable intentional resource discovery.
2. Designing and implementing navigation applications with the Cricket/Oxygen software compass, which provides "heading" (orientation) information to applications.