CS 525M – Mobile and Ubiquitous Computing Seminar

The Anatomy of a Context-Aware Application

Presented by Devanshu Mehta

The Facts

- Paper by Harter, Hopper, Steggles, Ward, and Webster.
- Based on research at AT&T Labs and Cambridge University in UK.
- Published in 2001.

Last Week at the Hospital

- Based on what was possible with current technology
- Limited in scope due to IM, etc.

Typical Context-Aware Apps.

Need to know:

- Location of user.
- Locations of equipment.
- Capabilities of the equipment and networking infrastructure.

i.e. CONTEXT

What the paper is about

- To build a sensor-driven (sentient) platform for context-aware computing.
- Hence, enable applications to follow users as they move through a building.
- Only requirement for user is to carry a sensor tag.
- The platform builds a dynamic model of the environment using these location sensors and other resource information.
- Use of the platform is described through an example where a user’s current working desktop follows them as they move.

System Components

- A fine-grained **location** system to locate and identify objects.
- A detailed **data model** which describes essential real-world entities that are involved in mobile applications.
- A persistent distributed **object system**, which presents the data model in a form accessible to applications.
- **Resource monitors** to communicate status information to a central repository.
- A **spatial monitoring** service, which enables event-based location-aware applications.

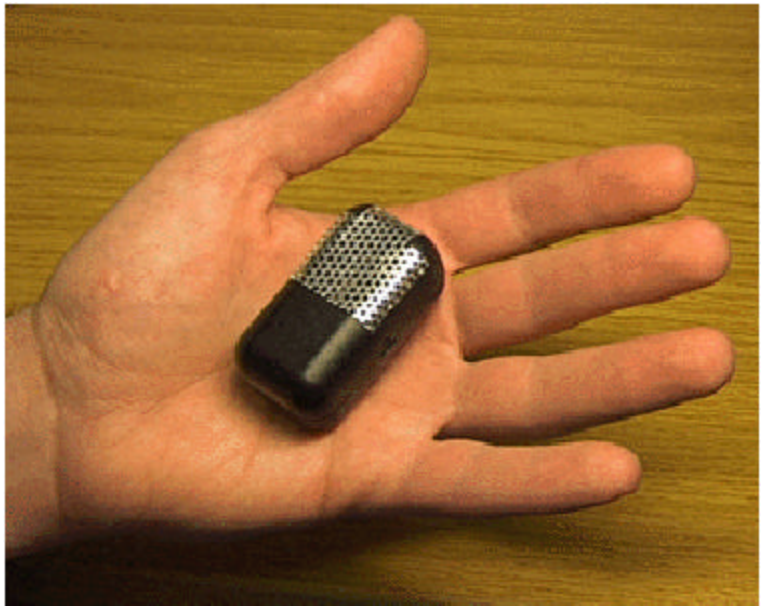
Location Sensing

Ideally:

- Provide fine-grain spatial info
- High update rate
- Unobtrusive
- Cheap
- Scalable
- Robust

Techniques:

- Radio-based (GPS) ✘
- Electromagnetic ✘
- Optical ✘
- Ultrasonic ✓



Bat:

- 5cm x 3cm x 2cm; 35g
- Contain radio transceiver, controlling logic and ultrasonic transducer.
- Has a globally unique identifier

Location Sensing

Ultrasound Receiver:

- Placed on ceiling
- Connected in daisy chain
- Base station periodically sends radio message to each Bat. Bat replies with short encoded pulse of ultrasound.
- Using speed of sound in air and times-of-flight of the pulse, the distance to the Bat can be found.
- If the distance to 3 or more non-collinear receivers is determined, the location in 3D space can be found (using multilateration).
- Errors due to reflections eliminated by algorithm.
- FACT: Real bats gain awareness of environs in a similar fashion!

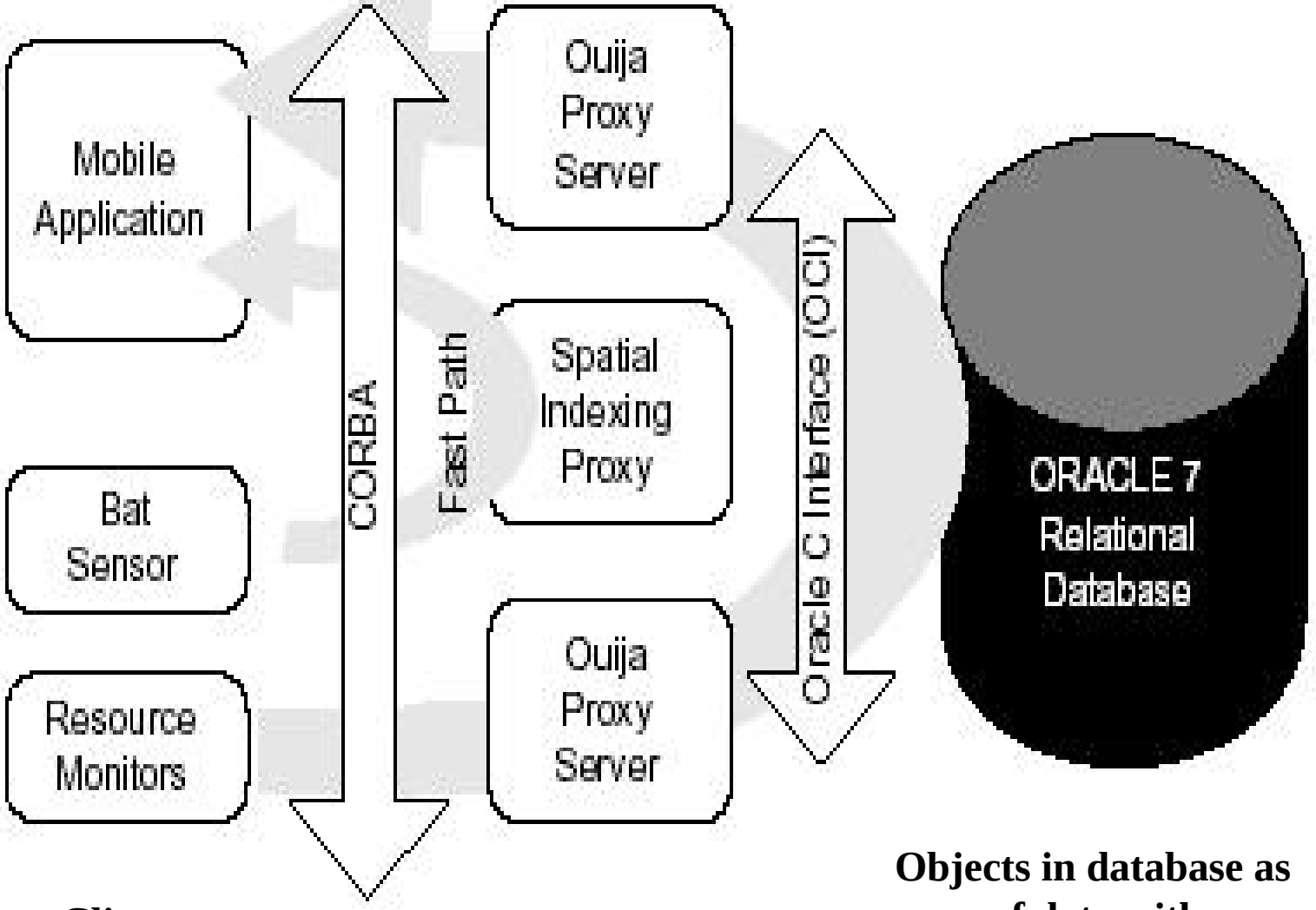
Location Sensing

- Mechanisms exist to detect when a Bat has left the locale and when a new Bat has entered the area.
- Using TDM ensures that nearby base stations do not interfere with each other.
- The location system developed:
  - could address up to 65535 Bats
  - over a 1000m^2 floor area
  - 2500m^3 volume
  - using 6 radio cells,
  - operating at 4 TDM channels
  - implying an aggregate total of 75 updates per second.

Modeling the Environment

- Uses an object-oriented model to describe entities in the real-world and their interactions.
- Represent people, computers, keyboards, displays, networks, telephones, furniture.
- Models types, names, capabilities and properties of all the real-world entities.
- The extent of the data model defines the limits of the system’s view of the world and consequently it’s domain of action.

Modeling Infrastructure

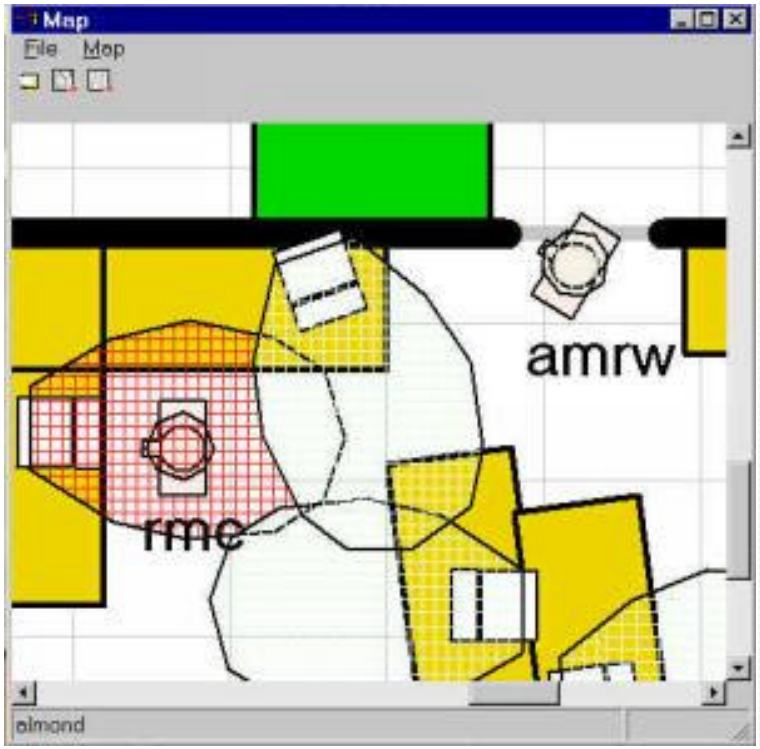


Updating the Model: Resource Monitors

- Installed on networked machines.
- Use OS system calls to get info about current status of machines.
- Periodically report changes in status to objects in database using CORBA interface
- Portable and work on many platforms.
- Do not impinge upon normal working of machine
- Implemented Types:
  - Machine activity e.g. keyboard activity
  - Machine resources e.g. CPU, memory usage
  - Network point-to-point bandwidth and latency

Spatial Monitor

- Spatial monitor transforms location data into containment relations.
- Objects can have one or more named spaces defined around them.
- A quad-tree based indexing method is used to quickly determine when spaces overlap, or when one space contains another, and applications are notified using a scalable event mechanism.



A map of an office, showing *visibility* spaces around computers, and *usage* spaces around people. The red shading indicates a containment state.

Programming with Space: General Facts

- Error correction is done using low-pass filters (random errors) and thresholds for max. velocities of objects (environmental ultrasound errors).
- Bats can request specific QoS; if not, default requests (2 sec for people; 5 for computers) are made.
- The system can also provide users with a browsable 3D model of the world which they can explore.

Active Badge System: Why not?

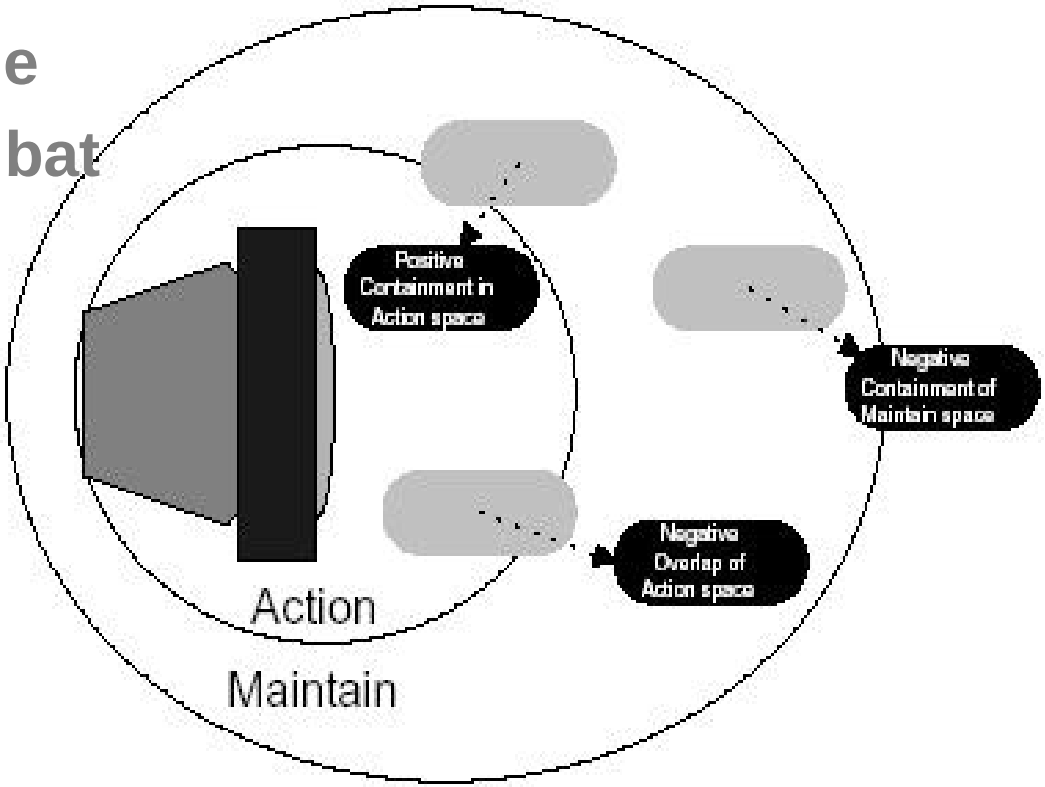
ActiveBadge is an existing system that can be used to locate the user and port user's X Windows System environment to current desktop.

Why this approach is not good enough:

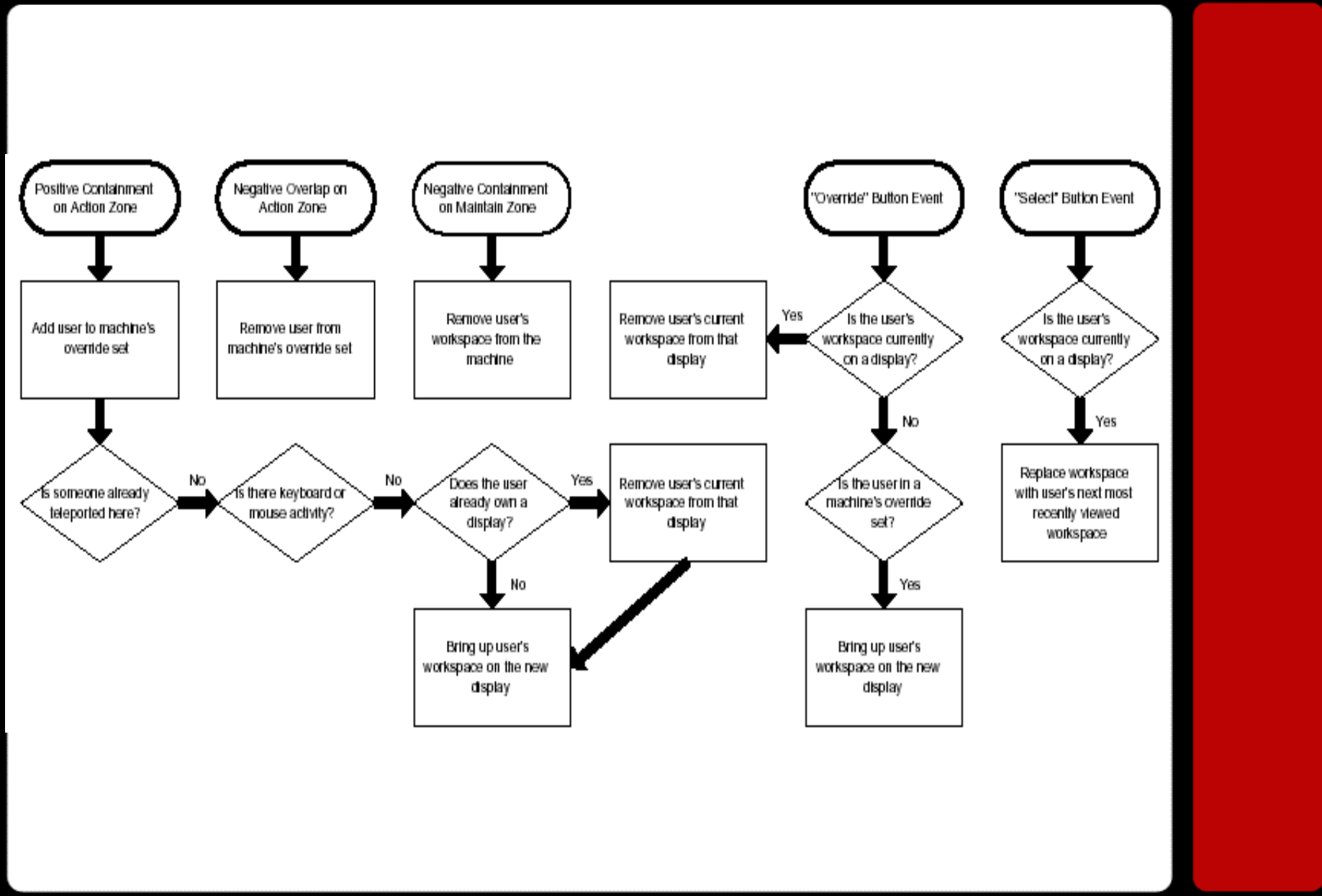
- Room-sized granularity- required cycling.
- Screen flashes to communicate- distracting!
- Does not have resource information- hence assumes all machines are working and ready.

Bat Teleporting

- **Create desktops which follow their owners around.**
- **Decisions based on:**
  - **Active Zone**
  - **Maitain Zone**
  - **Buttons on bat**



Bat Teleporting



Wrapping Up

Features of System:

- Event-driven nature
- Highly available world model
- Fine granularity of location sensing
- Resource monitoring allows for adjustments on basis of system capabilities and usage patterns.

Ultimately, the success of such a system will depend upon:

- Accuracy in location sensing
- Curing bottlenecks
- Scalability
- Convenience of bat

However, the paper provides great insight in:

- Constructing a workable sensor-driven system.
- Issues involved (especially spatial) when designing such a system.
- Directions for future work and applications of the system.

Final Words

Other Uses:

- “Follow Me” Video conferencing
- Context-aware information retrieval
- Smart Posters
- Resource Ownership

More information on current status of project: http://www.uk.research.att.com/spirit/