A Component Architecture for an Extensible, Highly Integrated Context-Aware Computing Infrastructure

William G. Griswold, Robert Boyer, Steven W. Brown, and Tan Minh Truong
University of California, San Diego
ICSE 2003 - Portland, Oregon

Presented by: Justin Erenkrantz,
ActiveCampus at UCSD

- [http://activecampus.ucsd.edu/](http://activecampus.ucsd.edu/)
- Provide location-based applications
  - Also known as services
- Understand how such systems are used
- Focus on software systems
  - Geared for mobile devices
Growth presents challenges

- UCSD will add 10k students in 10 years
- How to facilitate a cohesive community?
- Students are increasingly busy
- Mobile technology is getting affordable
- Provide tools to help build communities
ActiveCampus Buddy

- Instant messaging client
- Annotated with location
- Display people nearby
- Display people online
ActiveCampus Map

- Shows current location
- Campus map overlayed
- Indicates building names
- Location of buddies
What is context?

- Situation is critical to context
- Tools can help determine context
  - Alidade: compass, prism, magnifier
  
  “Constitute the selection, superimposition, and rendering of representations of task-relevant context”
Needs for Software Architecture

- Add services easily
  - Anticipate future changes
  - Introduce separation of concerns
- Desire critical constraints
  - Do not sacrifice integration
  - Performance is critical
Goals for Extensibility

- Add new services and functionality
- Introduce new sensor input
- Incorporate new physical entities
- Represent locations multiple ways
- Use new classes of user devices
Building upon Context Toolkit

- Previous work by Dey and Abowd
- No useful architectural style presented
- Desire to have efficient communication
  - Context Toolkit may be too heavy
- Desire to produce integrated applications
  - Services change over time
ActiveCampus Architecture

- Centralized, layered system architecture
  - Computation by central server
  - Minimizes demands on portable devices
- Receive input from sensors (handhelds)
- Utilize web standards for display
  - Handhelds or desktops
Initial Architecture Layers

- **Data Storage**
- **Data Abstraction**
- **Object Correlation**
  - Mapping data to internal forms
- **Environment Proxy**
  - Transport to external devices
Problems with Architecture

- **Entity definitions saw churn and bloat**
  - Adding alternate representations hard
- **Services were not decoupled properly**
  - Interdependent chain of services
- **Performance was becoming unacceptable**
  - Database access became bottleneck
Revised Architecture

1. Entity Modelling
   - (Id_s, loc_s)
   - Raw

2. Situation Modelling
   - id_s → id_e
   - Object Refinement

3. Environment Proxy
   - (id_e, loc_n...)
   - Normal

4. Device
   - loc_n → map
   - Situation Refinement
   - id_n → svc
   - Meta
   - Transport (xml,wireless)
   - Sensor-Entity Reconciliation
   - Transport (html,map)
   - Service
   - GUI
   - caching
   - caching
Addressing Entity Bloat

- Intrinsic blurred with presentation
  - People may have the same screen name
- Performed entity normalization
  - Isolates only essential characteristics
- Object Correlation is Situation Modeling
  - Tries to determine what is happening
Achieving Low Coupling

- Services available for subject about object
  - John’s buddy service about Jane
- Services registered at startup
- Services provide standard interfaces
  - Defines compatibility between services
- Compatible services called when needed
Optimizing Performance

- Prior concerns may impact performance
- Two-level caching system deployed
  - Inter- and Intra-service caching used
- Allows for inconsistent and stale data
  - Location ten seconds ago is ‘fine’
- Allows minimization of communication
Impact of Architecture

- Isolate functionality in layers
- Add rules for combining components
- Present situational context to users
- Keys in on how services interact
- Support of new devices styles difficult
Conclusions

- Demonstration at UbiComp 2003
  - Opportunity to use around Seattle
- Still determining what styles work best
  - Understand tradeoffs in UbiComp
- Feedback and experience only answer