Open Architecture Approach to Internet-Scale Context-Awareness

Robert Boyer W. G. Griswold
University of California at San Diego
{ rboyer, wgg } @cs.ucsd.edu
January 5, 2005

One Motivating Scenario

- "Prison without bars" (UK)
  - Paroled not imprisoned
  - Alert if parole restrictions are violated
  - Locationing device on ankle bracelet
    - Constantly reports position to w/in 2 meters
- Extend to Monitoring Criminal Gang
  - Condition of Parole – not to associate
  - On violation, notify field agent to respond
  - Parole field agent has lightweight edge device

Internet-Scale Context-Awareness

- Network enabled *sensors* everywhere
- Human imagination for data relationships

- Location Proximity
- Energy Management
- Financial Decisions
- Traffic Routing
- Health Monitoring
- Sensor Networks

Our problem generalizes to…

- Dynamic data – high data rates
- Multiple publishers – distributed

- Multiple variable relationships (e.g., $a < b$)
- Multiple publishers (e.g., $(Ax-Bx)^2 + (Ay-By)^2 < d^2$)

- Multiple subscribers – distributed
  - Individualized usage patterns in defining relationships
  - High overlap in data elements – low equality of interests (e.g., proximity relationships tailored by and for an individual)

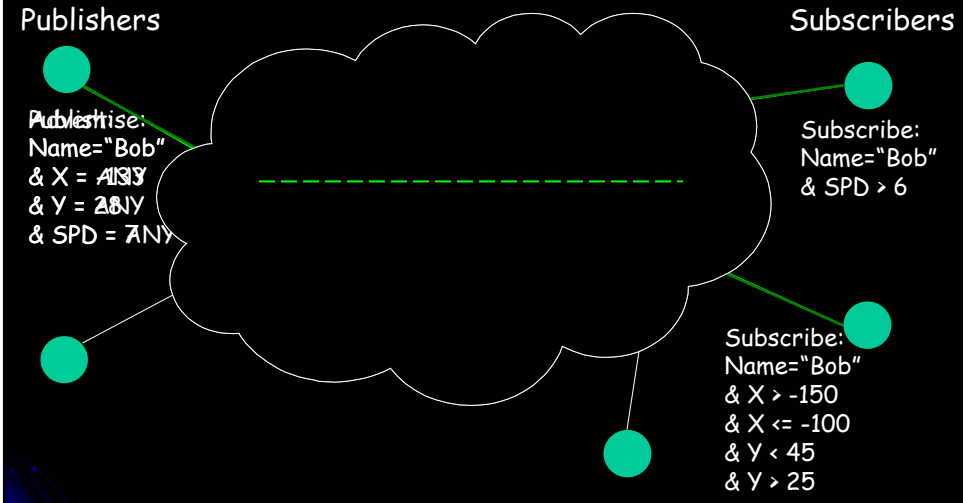
- General purpose, efficient, & scalable solution

Outline

- Motivation for ***new*** technical support
  - Content-Based Publish / Subscribe (CBPS)
  - But, was not designed for ***multi-publisher, dynamic data, attribute-relationships***
- Fulcrum – open implementation approach
  - System Design & Development
  - Avoids Expressiveness / Efficiency Tradeoff
- Evaluation

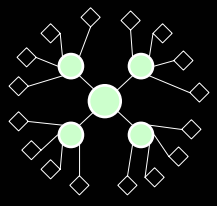
Content-Based Pub / Sub (CBPS)

Advertisements about… Subscriptions for… Publications of… ***Events***



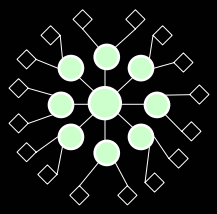
Uniform Distribution, Radial Network Average #hops approaches diameter

640 users (edge)



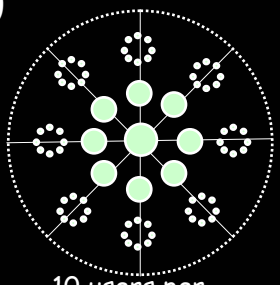
160 users per edge node with 4 edge nodes

Avg Hops = 3.5



80 users per edge node with 8 edge nodes

Avg Hops = 3.747



10 users per edge node with 8 edge nodes 8 2nd level nodes

Avg Hops = 5.71

Content-Based Publish / Subscribe

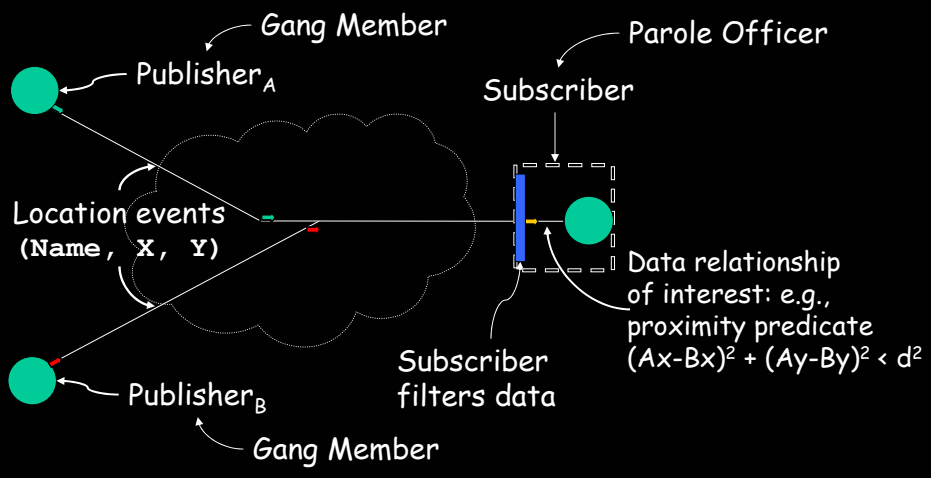
- Content-Based Routing
  - Associative Addressing
- Great separation of concerns
  - Plug-in more event brokers, pubs or subs
  - Contextually Relevant Filtering
- Efficient data distribution
  - Routing techniques limit dupes on a path
  - *Fast* event routing at each node of network
- Scalability – best for pubs with many subs

CBPS – Tradeoffs

- Limit expressiveness to gain efficiency
  Uses primitive types, relational ops, conjunctive predicates
  → To gain fast pattern matching at event broker nodes

- What about dynamic data relationships?
  - How do you do: price / earnings < 15
- What about multi-publisher situations?
  - How do you do: $(Ax - Bx)^2 + (Ay - By)^2 < 10^2$ m
  - How do you do: $Price_{SDGE} * Rate_{Home} < \$0.10$ / hr

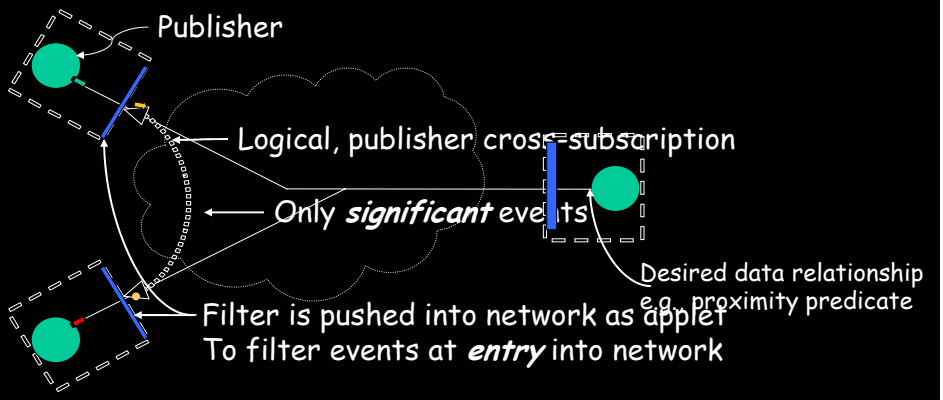
Current Evaluation at Consumer



Outline

- Fulcrum – open implementation approach
  - System Design & Development
  - Avoids Expressiveness / Efficiency Tradeoff
- Evaluation

Solution: Filter at Event Entry



Open Implementation

Allow programmer to share usage knowledge

An *Active Subscription* extends a predicate subscription with an implementation strategy instantiated as mobile code

subscribe(predicate)

subscribe(predicate, activeSubscription)

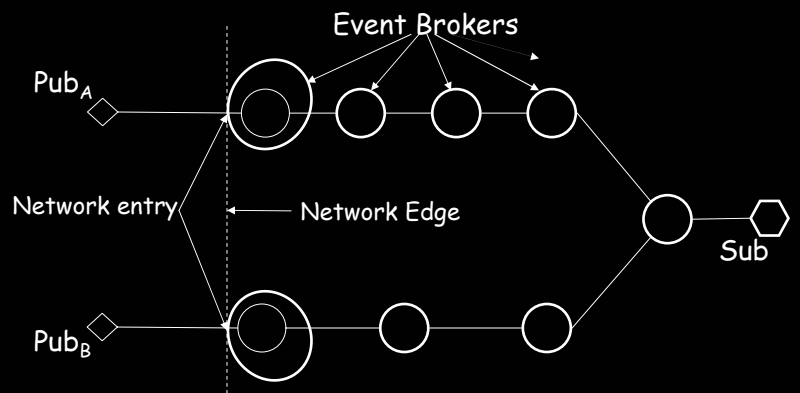
activeSubscription ::=

{ <*entryPredicate*><*appletParams*> }+ <*appletCode*>

The applet code is a first-class pub/sub entity

Subscriber submits an Active Subscription

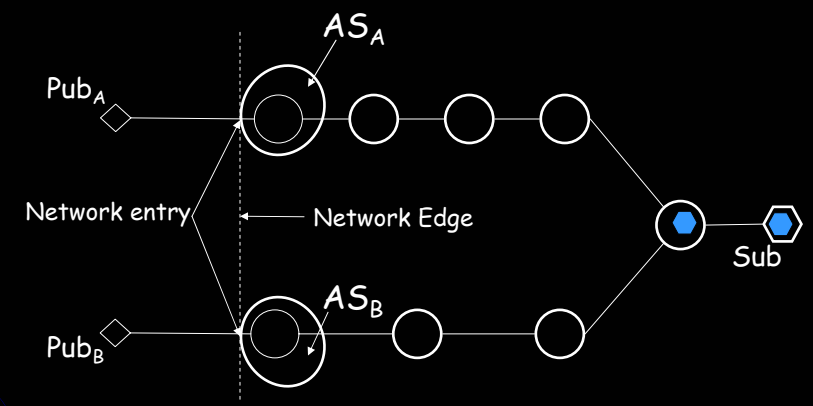
```
subscribe( P_proximity = { Name="AB" & "(Ax-Bx)² + (Ay-By)²" < 10² },
     AS_A = { < Name="A", X=ANY, Y=ANY >< "A", "B", rptTrue > },
     AS_B = { < Name="B", X=ANY, Y=ANY >< "B", "A", rptFalse > },
     <appletCode> ) ;
```



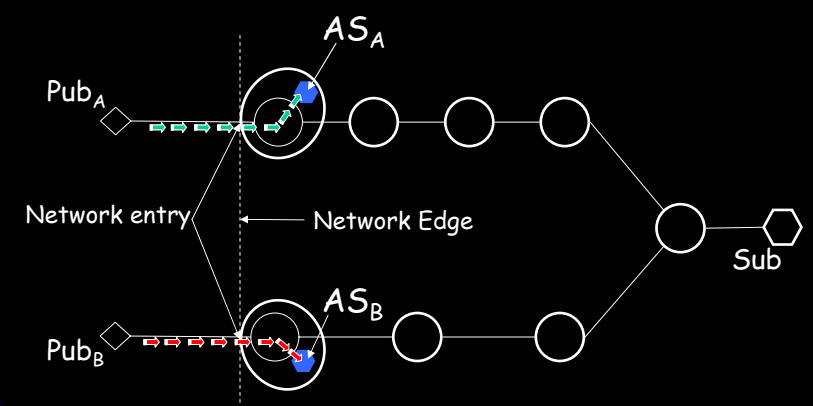
AS deployed to event-entry network edges

Network traversal uses standard associative addressing via *entryPredicates*
AS_A < Name="A", X=ANY, Y=ANY > **AS_B** < Name="B", X=ANY, Y=ANY >

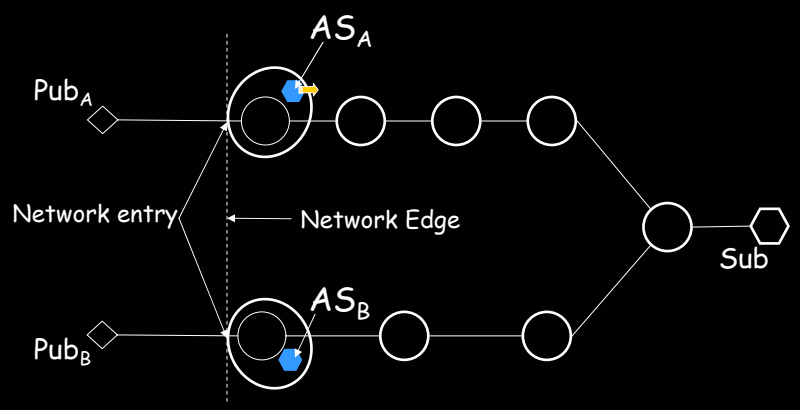
appletCode* is instantiated as AS_A and AS_B using *appletParams
AS_A < "A", "B", rptTrue > **AS_B** < "B", "A", rptFalse >



AS_A & AS_B directly subscribe to the original location events, thus logically *filter* events on network entry

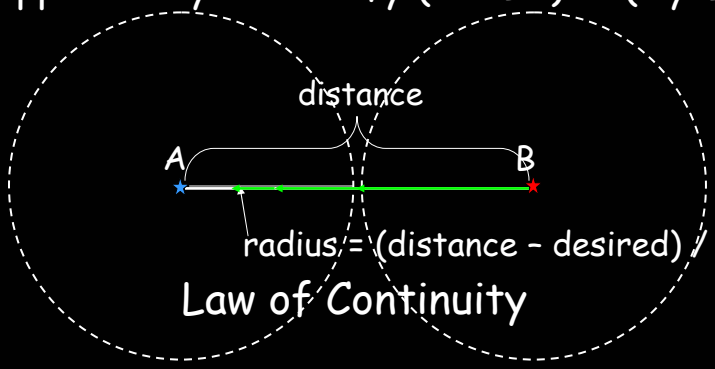


AS_A reports satisfaction of proximity event based on appletParams – i.e., rptTrue



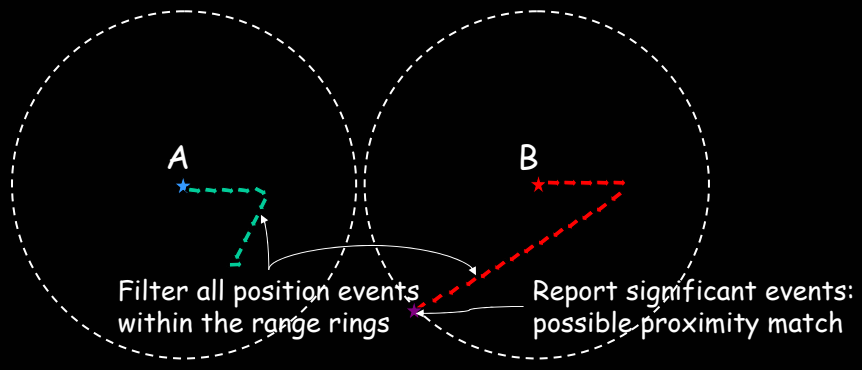
Active Subscriptions are Distributed Algorithms

Proximity: range ring represents first opportunity to satisfy $(Ax-Bx)^2 + (Ay-By)^2 < d^2$

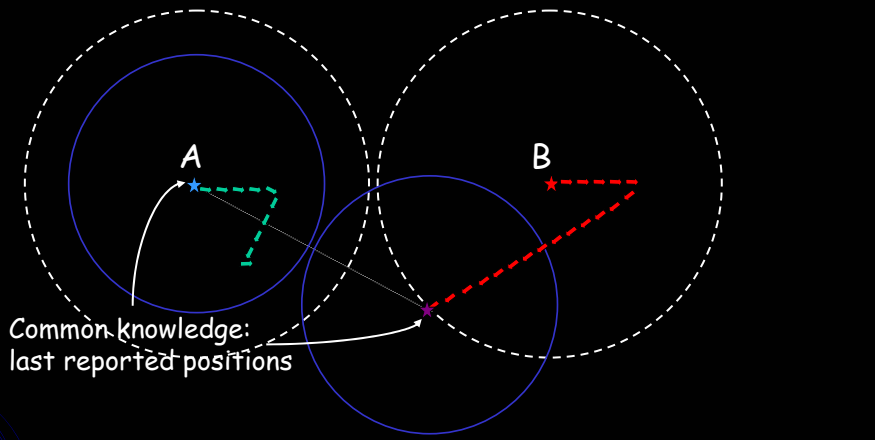


Example: determine gang member proximity

Movement information is semantically significant only when moving outside ring



Re-sync using shared knowledge Re-compute new range rings based on last reported locations



Increases Scalability

1. Does not propagate useless information
   - High data-rate events stopped at entry
   - Reduces network usage and CPU costs
2. Distributes processing costs to entry nodes
   - Can increase reporting rate, sensors, users, dist
3. Only semantically significant synthesized events exchanged between collaborators
4. Only single event passed on to subscriber

   Allows lightweight clients to use complex, multi-publisher dynamic data relationships

Evaluation - Data Setup

Data from UCSD's ActiveCampus experiments

643 users (anonymized for privacy)

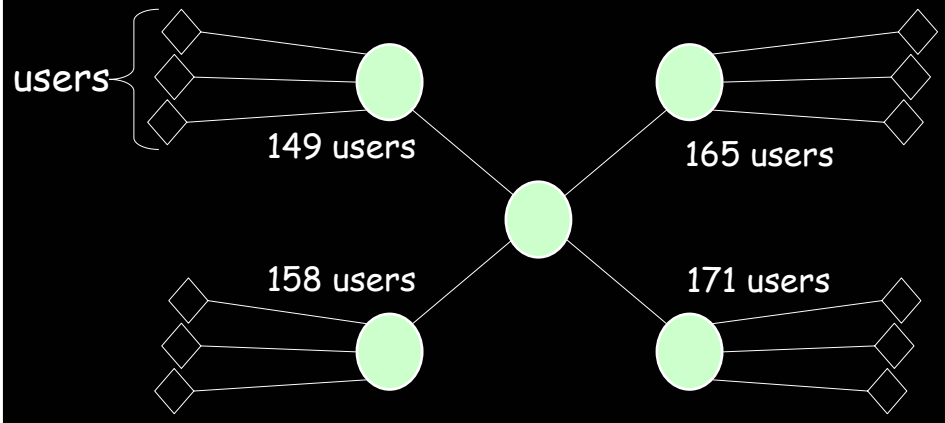
2,165 buddy relationships

604,800 seconds (1 week)

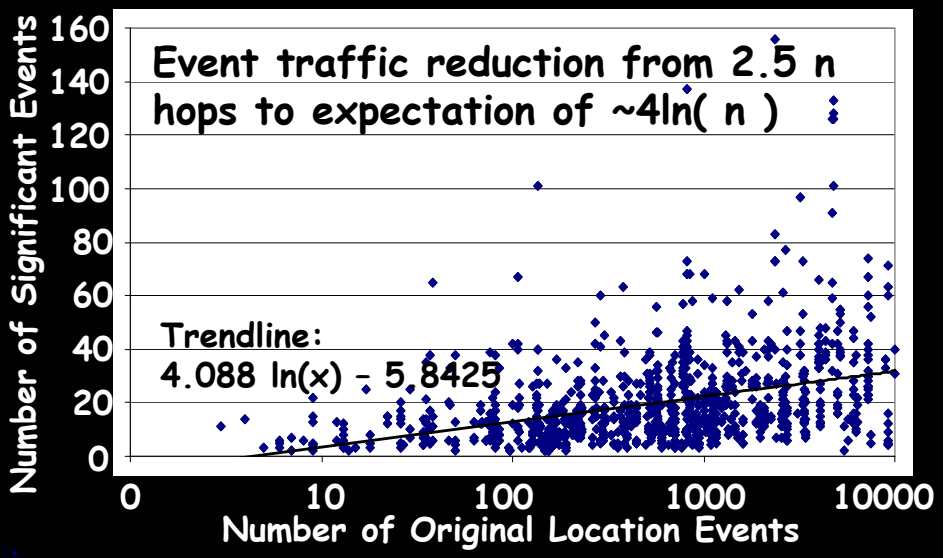
1-6 events per logged-in user per minute

360,067 location reports

Evaluation – Network Layout



Evaluation – Results



Future Work

- ActiveSubscription Composition
- Triggers: before, around, after (ActiveDB, AOP)
- Expose system context (e.g. sub to subscription)
- Collaborative Caching Infrastructure
- User Interface Development
  - Information Dashboard
  - Information Availability Query
  - Automatic Compilation ➔ (A + B + C) / D < (E * F)
- Usage Guidelines (e.g., effective strategies)

Conclusion

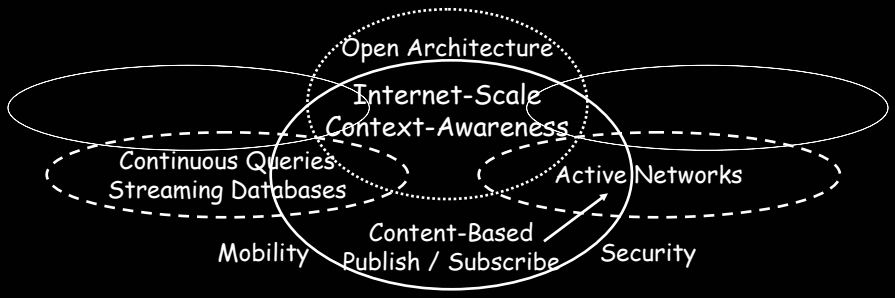
- Growing capability and desire for Internet-Scale Context-Awareness
- Existing technologies good but lack support for ***multi-publisher, dynamic data, attribute-relns***
- Open Implementation to enable info consumer to inject their contextual knowledge into network
- Developed Prototype Implementation – Fulcrum
  - Increases Expressiveness
  - Increases Efficiency
  - Improves Scalability

Backup Slides

Issues

- Reduce burden on large info providers (e.g., NYSE)
- Mechanisms to avoid DoS attacks
- Publisher or Subscriber mobility
- Content / payload security

Related Work



Content-Based Publish / Subscribe

SIENA	- Carzaniga, Wolf, et al
Solar	- Chen & Kotz – Dartmouth
Gryphon	- IBM T.J. Watson
Addressing Modes	- Hill & Knight – UVa

Context-Awareness

"Any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and application themselves"

[Dey, Salber, & Abowd 2001]

- 1st Generation – Device Awareness
  - Screen savers, Power Mgmt: screen, hard-drive, fan
- 2nd Generation – Personal Awareness
  - Single variable (e.g., location) against static backdrop
- 3rd Generation – Internet-Scale
  - Dynamic data (e.g., frequent changes in real-time)
  - Multiple variable relationships (e.g., a < b)
  - Multiple publishers (e.g., Ax < Bx)

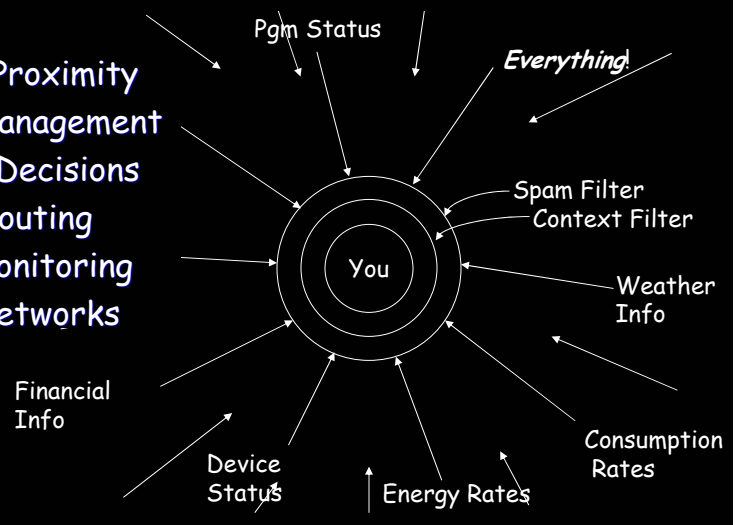
Evolution to Continuous Data

Imagine a world where every new application ***emits a continuous stream of messages*** that record every significant action. Imaging that routine documents, files and messages sent through the enterprise network and Internet are automatically scanned to extract meaningful data without having any impact on the sending or receiving applications. Imagine that sensors are placed strategically around the enterprise to observe what is happening in the physical world outside IT. Imagine what a business person could accomplish if he or she could ***tap into this wealth of information in a timely*** fashion. Imagine what problems could be anticipated and headed off before they caused damage.

[David Smith, Gartner Research, BIJ, May 2004]

Internet-Scale Context-Awareness

- Location Proximity
- Energy Management
- Financial Decisions
- Traffic Routing
- Health Monitoring
- Sensor Networks



Right Information – Right Person – Right Time

Radial Network – Average #Hops

```
Int       levelHops = 0 ;
double  levelExpected = 1.0 ;
double  hopsExpected = 0 ;
double  sumExpected = 0.0 ;
double  levelKids[] ; // init elsewhere

for( int i = 0 ;   i < numLevels ;   i++ )
{
   levelHops = (numLevels - i) * 2 ;
   if  ( 0 < i )
      levelExpected /= (levelKids[i] – 1.0) ;

   if  ( i + 1 < numLevels )
      levelExpected *= (levelKids[i+1] – 1.0) / levelKids[i+1] ;

   hopsExpected += levelHops * levelExpected ;
}
```