outline

1. Introduction to pervasive environments
2. Motivation
3. Data management challenges
4. Design and Implementation
5. Conclusions and future work
1. introduction – the brave new world

- Devices increasingly more
  {powerful ^ smaller ^ cheaper}

- People interact daily with dozens of computing devices (many mobile):
  - Cars
  - Desktops/Laptops
  - Cell phones
  - PDAs
  - MP3 players
  - Transportation passes

→ Computing is becoming pervasive
the brave new world

- Many standalone devices, no interaction, no collaboration
traditional mobile computing

- Mobile devices = traditionally standalone
  - All required information present locally
  - Synchronization through “cable” connection

- Wireless connectivity = new way for data exchange
  - Cellular networks, satellites, LANs and short range networks

- Mobile devices now able to connect to the Internet
  - Client end-points in client/proxy/server model
  - Initiate actions
  - Receive information from servers

→ Mobile devices have wireless connectivity
before client/proxy/server model

→ Mobile devices treated as (dumb) clients in client/proxy/server information management models
client/proxy/server model

Mobile devices treated as (dumb) clients in client/proxy/server information management models
ad-hoc networking technologies

• Ad-hoc networking technologies (e.g. Bluetooth)
  • Main characteristics:
    • Short range
    • Spontaneous connectivity
    • Free, at least for now

• Mobile devices
  • Aware of their neighborhood
    • Can discover others in their vicinity
  • Interact with peers in their neighborhood
    • inter-operate and cooperate as needed and as desired
    • Both information consumers and providers

→ Ad-hoc mobile technology challenges the traditional client/server information access model
before peer-to-peer model

→ Peer-to-Peer Approach to Information Management
peer-to-peer model

→ Peer-to-Peer Approach to Information Management
pervasive environment paradigm

• Pervasive Computing Environment

  1. Ad-Hoc mobile connectivity
     • Spontaneous interaction

  2. Peers
     • Service/Information consumers and providers
     • Autonomous, adaptive, and proactive

  3. “Data intensive” “deeply networked” environment
     • Everyone can exchange information
     • Data-centric model
     • Some sources generate “streams” of data, e.g. sensors

→ Pervasive Computing Environments
• static and moving sensors
• database clouds with caching and computing functionality
• profiles and context knowledge

→ Hot topic
2. motivation – street scenario

- Cars, street signs, traffic lights and road-side businesses can exchange latest location-dependent information
motivation – conference scenario

- Smart-room infrastructure and personal devices can assist an ongoing meeting: data exchange, schedulers, etc.
imperfect world

• In a *perfect* world
  • everything available and done automatically

• In the *real* world
  • Limited resources
    • Battery, memory, computation, connection, bandwidth
      ➔ Must live with less than perfect results
  • Dumb devices
    ➔ Must explicitly be told What, When, and How
  • “Foreign” entities and unknown peers

• So, we really want
  
  **Smart, autonomous, dynamic, adaptive, and proactive methods to handle data and services…**
3. what is pervasive environment?

- A "type" of mobile distributed database / distributed file system
  - 4 orthogonal axes (Dunham and Helal, 1995)

- Challenges inherited from (common with) client/proxy/server models in wireless networks
  - Distributed data, low bandwidth and frequent disconnection
  - Query processing and optimization
    - location transparency and awareness
  - Caching
  - Replication
  - Name resolution
  - Transaction management
challenges – is that all? (1)

1. Spatio-temporal variation of data and data sources

   • All devices in the neighborhood are potential information providers
   • Nothing is fixed
   • No global catalog
   • No global routing table
   • No centralized control

   • However, each entity can interact with its neighbors
     • By advertising / registering its service
     • By collecting / registering services of others
2. Query may be explicit or implicit, but is often known up-front

- Users sometimes ask explicitly
  - e.g. tell me the nearest restaurant that has vegetarian menu items

- The system can “guess” likely queries based on declarative information or past behavior
  - e.g. the user always wants to know the price of IBM stock
3. Since information sources are not known a priori, schema translations cannot be done beforehand

- Resource limited devices
  - so hope for common, domain specific ontologies 😊

- Different modes:
  - Device could interact with only such providers whose schemas it understands
  - Device could interact with anyone, and cache the information in hopes of a translation in the future.
  - Device could always try to translate itself
    - Prior work in Schema Translation, Ongoing work in Ontology Mapping.
4. Cooperation amongst information sources cannot be guaranteed

- Device has reliable information, but makes it inaccessible
- Devices provides information, which is unreliable
- Once device shares information, it needs the capability to protect future propagation and changes to that information
challenges – is that all? (5)

• Need to avoid humans in the loop
  • Devices must dynamically "predict" data importance and utility based on the current context

• The key insight: declarative (or inferred) descriptions help
  • Information needs
  • Information capability
  • Constraints
    • Resources
    • Data
    • Answer fidelity

• Expressive Profiles can capture such descriptions
challenges – is that all (6)

- Use of profiles
  - Cherniak et al, 2002
    - Profiles = data domains + fixed utility values
    - Not specifically targeted to pervasive environments, but very applicable to them

PROFILE Traveler

DOMAIN

R = www.hertz.com
S = +shuttle +logan +downtown
D = +directions +logan +boston +downtown

UTILITY

U (S) = UPTO (1,2,0)
U (R [#D > 0]) = 5
4. Our data management architecture

MoGATU

- Design and implementation consists of
  - Data
    - Metadata
    - Profiles
  - Entities
    - Communication interfaces
    - Information Providers
    - Information Consumers
    - Information Managers
MoGATU P2P model representation
MoGATU P2P model representation

- Information Generator Y
- Information Provider X
- InforMa
- Routing Information
- Information Provider X
- Information Generator Y
- Bluetooth Interface
- 802.11b Interface
- User Interface

MoGATU P2P model representation
MoGATU – metadata

• Metadata representation
  • To provide information about
    • Information providers and consumers,
    • Data objects, and
    • Queries and answers
  • To describe relationships
  • To describe restrictions
  • To reason over the information

→ Semantic language
  • DAML+OIL / DAML-S

• http://mogatu.umbc.edu/ont/
MoGATU – profile

- Profile
  - User – preferences, schedule, requirements
  - Device – constraints, providers, consumers
  - Data – ownership, restriction, requirements, process model

- Profiles based on BDI models
  - Beliefs are “facts”
    - about user or environment/context
  - Desires and Intentions
    - higher level expressions of beliefs and goals

- Devices “reason” over the BDI profiles
  - Generate domains of interest and utility functions
  - Change domains and utility functions based on context
MoGATU – entities

- Communication Interface
  - Network abstraction
    - Routing / Discovery not concerned with underlying network
  - Registers and interacts with local Information Manager (InforMa)
    - InforMa still aware of the network attributes

- Information Provider
  - Subset of a world knowledge
  - Registers and interacts with local InforMa

- Information Consumer
  - Access to information through local InforMa
  - Registers and interacts with local InforMa
MoGATU – information manager (1)

- One Information Manager (*InforMa*) per device
  - Various types based on device strength and will

- Indexing / Discovery
  - Through advertisements and solicitations
  - Relying on local cache
  - No fixed schema or ontology

- Advertisement
  - Implicit discovery
  - Advertise local (and even remote) providers / answers
  - Local providers *must* register with local *InforMa*

- Solicitation
  - Explicit discovery
  - Asks for remote providers / answers
MoGATU – information manager (2)

- Routing
  - Data-based table routing
  - Promiscuous mode
    - Uses cached advertisement information
    - Matching using CLIPS

```
informa_route_query(f, t, query) {
    if (local(t))
        if (answer = cached_answer(query) && valid(answer)) return answer
        if (answer = contact_local_info_provider(f, t, query)) return answer
        else error(no_answer)

    if (intercept_foreign_queries)
        if (answer = cached_answer(query) && valid(answer)) return answer;

    if (willing_to_forward)
        if (nexthop = lookup(t)) return forward_to(nexthop)
        else
            if (local(o)) forward_to_random(o, d, query)
            else error(no_destination)
        else error(forwarding_denied)
        error(no_answer)
}
```
MoGATU – information manager (3)

- Caching
  - Caches incoming data messages
    - DAML+OIL encoded information in profiles and data objects
  - Hits based on reasoning over associated metadata
    - Using CLIPS for reasoning
  - Replacement policies
    - Traditional LRU and MRU
    - Traditional LRU and MRU + profile-based space pre-allocation
    - Semantic-based
      - Space pre-allocation based on context and profile knowledge
      - Dynamic utility values for each cache entry based on context and profile
      - Using CLIPS
MoGATU – implementation

- Implemented
  - Communication interface for Bluetooth and Ad-hoc 802.11
  - Providers for gas, weather, traffic information, etc.
  - Consumers representing cars and PDAs
  - InforMa with preliminary support for
    - Query routing
    - Query processing
    - Semantics-based data caching
    - Limited notion of user profiles
    - Simple pro-active profile-initiated interaction
MoGATU – experiments

• Conducted experiments to evaluate

  1. How context and profile knowledge affect cache pre-allocation
  
  2. Performance of the semantic-based cache replacement algorithm against LRU and MRU approaches
  
  3. Networking aspect of the framework
     • Transmission time
     • Routing
MoGATU – simulation settings

• One day activity of a person
  • Starts at 8AM in Annapolis
  • Travels to Washington D.C. for 10AM meeting
  • Lunch at noon
  • Travels to UMBC for 2PM meeting
  • Shopping from 4:30PM until 5:30PM
  • Dinner at 8PM in Annapolis

• Her PDA has some profile knowledge
  • Limited information about the schedule
  • Plus other information about preferences and requirements
MoGATU – simulation settings

• Information sources
  • Represent cars, street lights, buildings, subway and people
  • At every time instance some sources may be available

• Available information
  • Different type (8)
    • Directions, traffic, gas, parking, merchandise, dining, subway, and anything else
  • Different utility value
    • Dynamically computed by each Information Manager based on context and profile knowledge
Ex1: cache allocation

1. How does profile-based pre-allocation compare to measured LRU cache allocation?

   a) Recorded cache content at every minute of the 12-hour simulation period while using traditional LRU for cache replacement

   b) Computed how context and profile knowledge affects cache pre-allocation

   c) Computed how a prior omniscient knowledge would affect cache pre-allocation

→ Without using the additional knowledge, some important data were not cached
   • LRU did not cache any subway data
   • LRU kept on caching restaurant data after the lunch was over
Ex1: cache allocation

Measured cache allocation when using LRU only

Computed cache pre-allocation using context and profile knowledge

Computer cache pre-allocation when everything is known apriori

Time of the day
8:00AM 9:40AM 11:20AM 1:00PM 2:40PM 4:20PM 6:00PM 7:40PM
Ex2: single queries with varying update

2. How does each cache replacement algorithm perform given varying update periods and single queries only?

- Update period from 1 to 128 minutes
  - Device’s preferred refresh rate to prolong battery life
  - A rate at which information providers appear/disappear
  - Most data has 10-minute lifetime

- Person asks 1 to 4 unique queries during each activity
  - While driving, one query about traffic, one for a gas station, etc.
  - 54 queries total
Ex2: single queries with varying update

![Graph showing cache hit success rate over cache update period (min). The graph compares different cache policies: LRU, LRU+P, MRU, MRU+P, and S+P. The x-axis represents cache update period in minutes, ranging from 1 to 128. The y-axis represents cache hit success rate, ranging from 30% to 100%. Each policy has a distinct line indicating its performance over time.]
Ex3: repeating queries with varying update

3. How does each cache replacement algorithm perform given varying update periods and REPEATING queries?

• Update period from 1 to 128 minutes

• Person asks same queries once every 5-minute period during each activity
  • While driving from 8AM until 8:45AM, the person asks for traffic update once every 5 minutes = 9 queries
  • 374 queries total
Ex3: repeating queries with varying update

![Graph showing cache hit success rate over cache update period (min)]
Ex4: repeating queries with constant update

4. How does each cache replacement algorithm perform given a constant update period and queries repeating at different intervals?

- Update period is fixed 5 minutes
  - More realistic scenario
  - Device can reflect context changes every 5 minutes to preserve resources

- Person asks same queries once every N-minute period during each activity
  - N from 1 to 128 minutes

→ Semantic-based approach stays in 90% range
Ex4: repeating queries with constant update

![Graph showing cache hit success rate over query period in minutes. The graph compares different cache management policies: LRU+P, LRU, MRU, MRU+P, and S+P. The x-axis represents the query period in minutes, ranging from 1 to 128, and the y-axis represents the cache hit success rate in percentage, ranging from 30% to 100%. Each policy has a line indicating its performance over the query period.](image-url)
Ex5 and Ex6: on system performance

• Transmission time
  • Used laptops and iPAQs equipped with WLAN and Bluetooth cards
  • Sending 1kB “query” from A to B and receiving response
    • over Bluetooth = 4.56s RTT
    • over 802.11 = 27ms RTT

• Routing
  • Existential experiment with again laptops and iPAQs
  • Multiple hop routes with dynamic topology
  • Used the data-based routing
  • Worked but need better measurements
5. conclusions and future work

• We have
  • Presented a need for DM in pervasive environments
  • Defined issues associated with possible DM solutions
  • Designed and described a preliminary implementation including
    • Query routing
    • Query processing
    • Semantics-based data caching
    • Limited notion of user profiles
    • Simple pro-active profile-initiated interaction

• Still need (future work)
  • Formal models for expressing the BDI-based profiles
  • Better routing and caching algorithms
  • More advanced (and faster) reasoning engines
  • Transaction support
questions