Università di Modena e Reggio Emilia

Autonomic Services for Browsing the World

F. Zambonelli & The Agents and Pervasive Computing Group

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

Outline

- Future pervasive computing scenarios
  - The “Browsing the World” concept
  - Key research challenges

- Our current researches directions
  - *“Overlay knowledge networks”* for the sensor networks continuum
  - The *“W4 model”* for handling contextual data
  - Towards a novel *situated agent model* and associated infrastructure

- Our preliminary prototype

Università
di Modena e
Reggio
Emilia

Motivations

- Computer-based systems and sensors will be soon embedded in everywhere
  - all our everyday objects
  - all our everyday environments
- Current deployments of pervasive services and sensor networks focus on special purpose systems, e.g.,
  - E.g., environmental monitoring and healthcare
- General purpose approaches are likely to emerge soon
  - Shared infrastructures of sensors, tags, smart-objects, cameras, Web 2.0 data
  - Generating general purpose data (“facts”) about the physical and the social worlds
  - Defining a sort of distributed digital “***world model***”
  - For general-purpose exploitation by “**browsing the world**” services

Agents and Pervasive
Computing Group

Examples of "Facts" in the World Model (1)

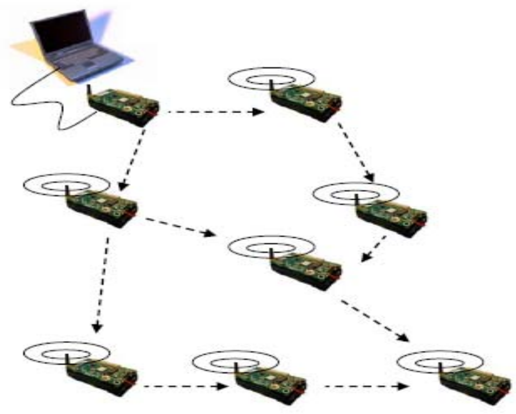
- Localization of a user
  - GPS, Mobile Phone Cell, WiFi
- Localization of a car
- Description of an object
  - RFID/barcode on the object
  - Camera recognizing the object and its characteristics (and possibly some visual tag)
  - As well as its location (possibly inferred from users' nearvy)

Università di Modena e Reggio Emilia

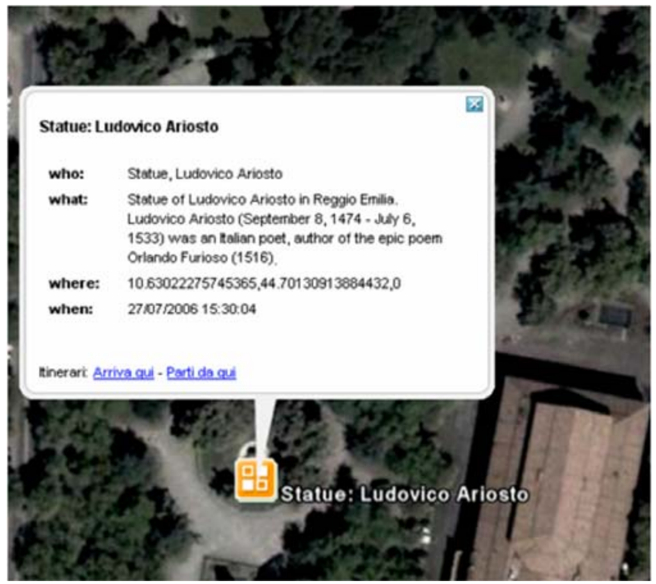
Agents and Pervasive Computing Group

Examples of "Facts" in the World Model (2)

- Environmental data from wireless sensor networks
  - Temperature, Movement, Sound, etc.
  - Typically enriched with temporal and geographical information



- Web 2.0 Information
  - Google Earth placemark
  - Geo-tagging
  - Blog fragments
  - Data extracted via Google API



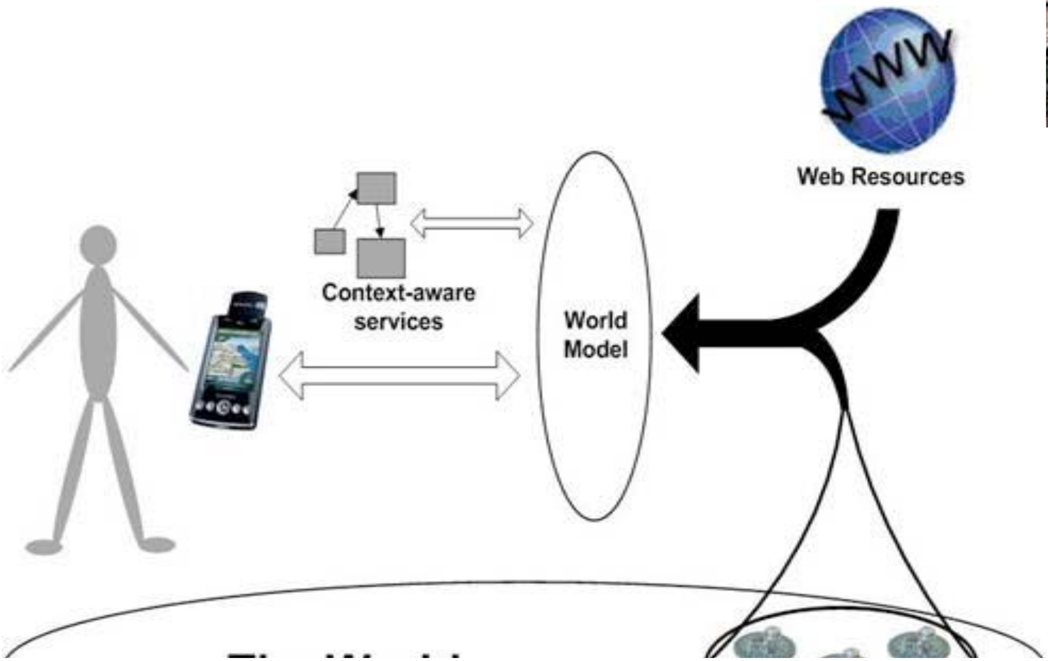
Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

“Browsing the World” Scenario

Exploit the world model to

- Implement services to help us interact with the physical world (e.g., personalized real-time maps)
- Have services coordinate with each other in a context-aware fashion to achieve global goals (e.g., traffic control systems)



PLEASE NOTE: Users and services are themselves part of the world, and thus must be part of the world model.

Data generators and data users are not necessarily

Why Self-organization?

- The complexity, openness, and dynamics of the scenario makes it impossible for humans to stay in the control loop
  - Services must autonomously organize their activities anc must self-adapt to the current situation of the world
  - Humans can intervene only on limited portions of the scenario

- But…..is the challenge really with services?

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

Mechanisms of Self-organization

- Direct interactions
  - Components (services/agents) interact directly with each other to reach common goals in a self-organizing way
  - Not suitable to open and dynamic systems (who are the others?)
  - Not suitable to situated activities (where most issues relates to what’s happening in the world, rather than to what the others are doing)
- Mediated interactions
  - Self-organization takes place via sensing/affecting a common environment (stigmergic interactions, as in ant colonies)
  - The environment has its own properties and processes which rules the behavior of the colony (diffusion and evaporation of pheromones)
  - Components do not need to know each other (ants are blind)
  - Self-organization is by definition situated
- In the latter case, then, the real challenge relates to **engineering the environment** rather than services
  - In our case, **engineering the world model**!!

Università di Modena e Reggio Emilia

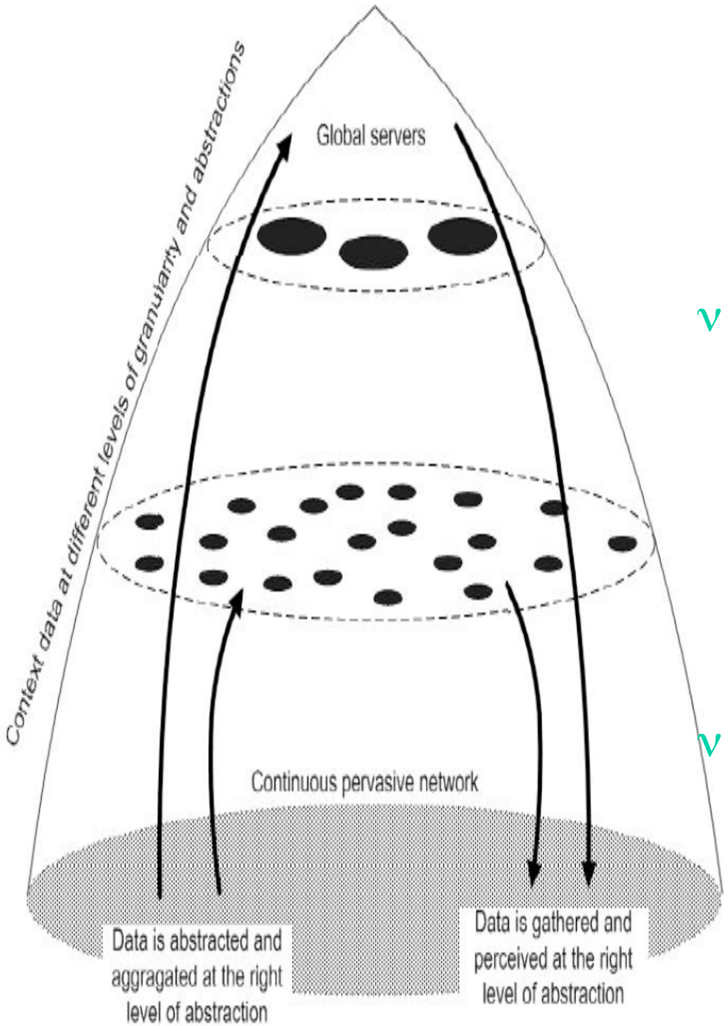
Engineering the World Model

- Extreme heterogeneity of data
  - Information coming from sensors, cameras, Web 2.0, etc.
  - Variable density of devices (potentially continuum)
- Massive amounts of data produced
  - No way to collect all data (potentially infinite) at a place
  - Need to aggregate, prune, evaporate, analyse data where it is produced
- Inherent dynamism and decentralization
  - Devices (and the associated data) come and go at any time
  - No way to control each devices due to decentralization
- Uncertainty and reliability
  - No user/services can be ensured the availability of specific data
  - Still services must be able to go on in any case
- The necessity arises to **exploit self-organization within the world model**!!!
  - Conceptual shift from self-organizing services to self-organizing data

Agents and Pervasive Computing Group

Università di Modena e Reggio Emilia

The Knowledge Continuum



- Data virtually generated in both
  - A sensor network continuum
  - A level of Web 2.0 services
  - At any level in between (e.g., cameras, loca servers, etc.)
- Data should flow up the pyramid
  - In aggregated forms, via proper self-organizing algorithms for data aggregation
  - To limit the amount of data (potentially infinite) to be managed by services (which have bounded rationality)
- Data should flow down the pyramid
  - To self-aggregate data coming from low-level sensors with data coming from higher-level ones (or from the Web)
  - To let agents exploit all available data in a uniform way and locally

Agents and Pervasive Computing Group

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

Our Current Research

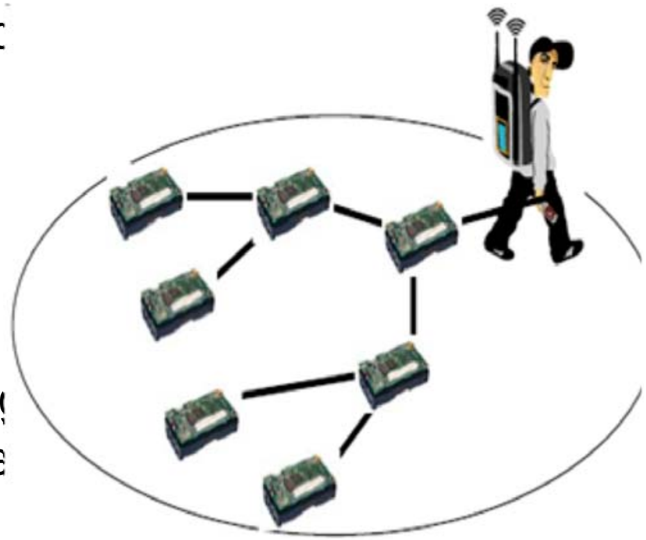
- Study general self-org algorithms and approaches to deal with data gathering and information extraction
- Identify a to represent context information as a basis for the building of an effective and usable world model.
- Define a general situated agent model and the associated infrastructure for autonomic communication services for browsing the world

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

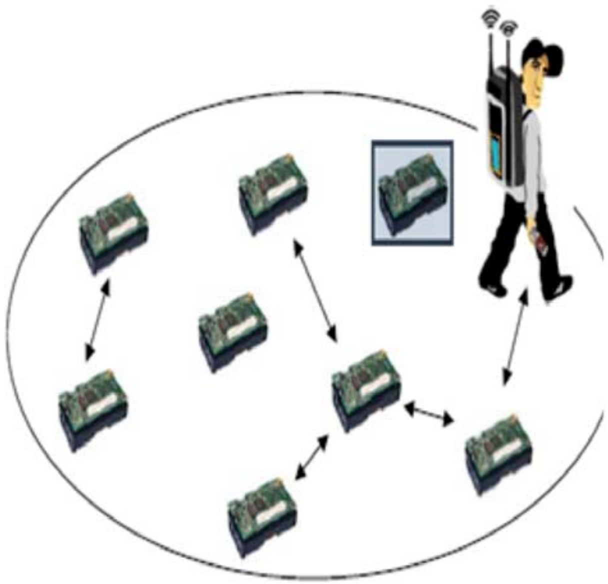
Data Gathering and Information Extraction: Usual Approaches

- Fully Application-driven
  - Collect and aggregate data *on demand*
  - E.g., build a spanning in the sensor network to route the collected data to a requiring node and aggregate a result
- Problems:
  - Highly expansive in terms of communication and energy, can also can be very slow
  - All the burden of understanding and interpreting data is on the clients and no actual world model is ever built
  - Direct connection with the physical devices (cannot handle the continuum problem)
  - In Weyns & Omicini's terms, there is no "engineering of the environment"



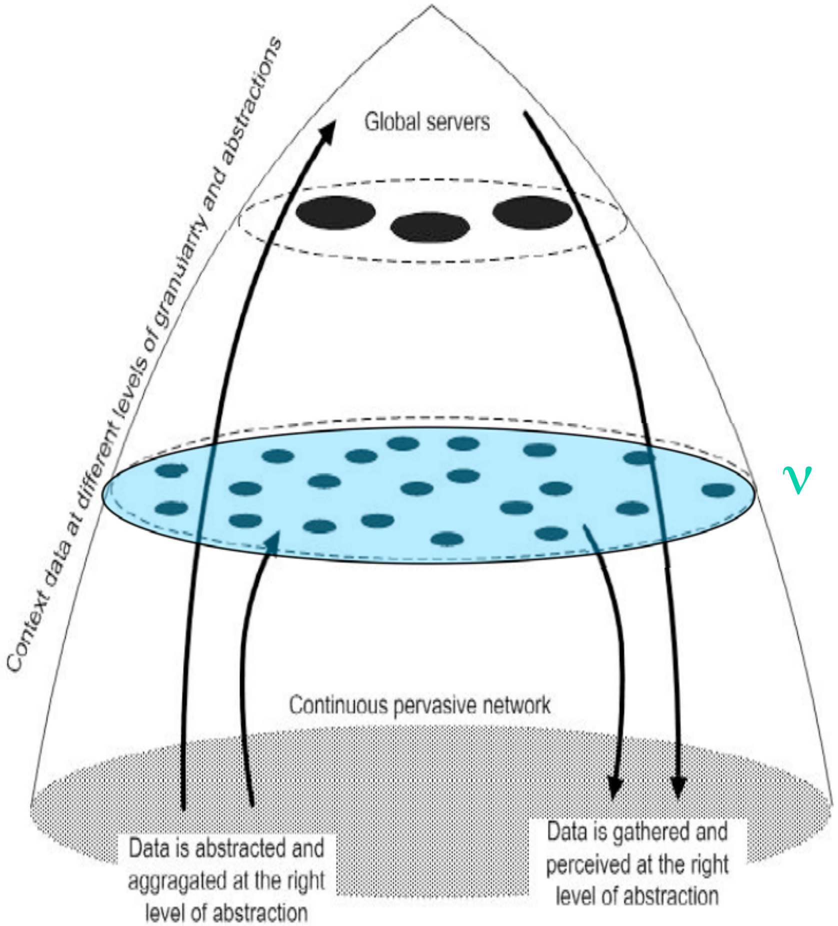
Data Gathering and Information Extraction: Our Approach

- Environment-driven:
  - Understand what the environment is
  - Automatically build specific views of the environment to later facilitate users/services in achieving context-awareness ◊ *engineering the environment!!*
- Key guidelines
  - Abstract from the device level to tackle heterogeneity ◊ the continuum abstraction
  - Enforcing self-organization to tackle decentralization
  - Enforce multi-scale observation levels via multiple knowledge views



Università
di Modena e
Reggio
Emilia

Building Knowledge Overlays over the "Continuum"



- We need to aggregate data from the continuum
  - Compact representation
  - Without losing relevant information
  - And indeed provide more information of "what's happening"
- Key idea
  - Identify aggregation regions
  - Enable "per region" views of specifi characteristics of the environment

Agents and Pervasive
Computing Group

Identifying Aggregation Regions

- Our "Region Aggregation Noise" (RAN) approach, considers the following:
  - A distributed algorithm is continuously running in the network as a sort of *"background noise"*
  - with the goal of partitioning the sensor network into regions characterized by similar patterns (as an overlay of virtual links)
  - Abstracting from the specific density/structure of the sensor network

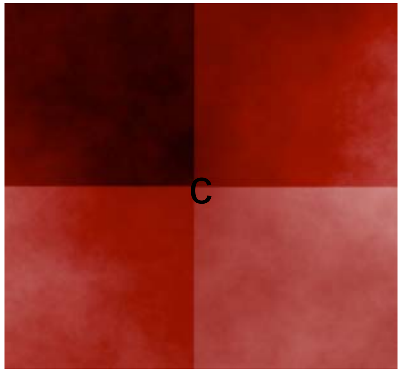
```
Do_forerever:
    Wait(t);
    neigh[] = Select_neighbor(num_neigh);
    Foreach(neigh[])
        Data = Exchange_data();
        Update_link(data);
```

```
Update_link:
if D(v(si),  v(sj)) < T {
    l(si,sj) = min(l(si,sj) + delta, 1)
} else {
    l(si,sj) = max(l(si,sj) - delta, 0)
```

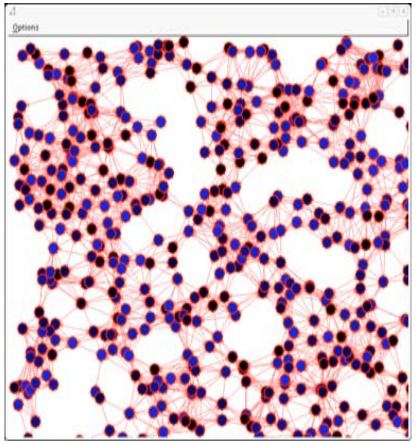
Università di Modena e Reggio Emilia

Simulated Region Formation

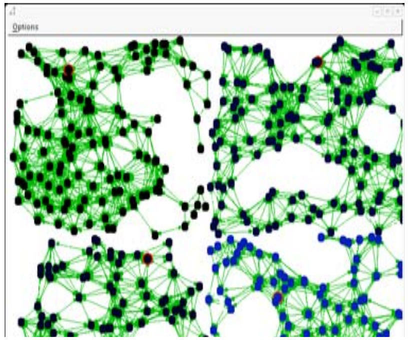
- Simulated sensors are embedded in an environment characterized by different patterns of sensed data



- At first they are not logically connected with each other



- Gradually they recognize regions
- Then a partitioning emerges based on the environmental patterns

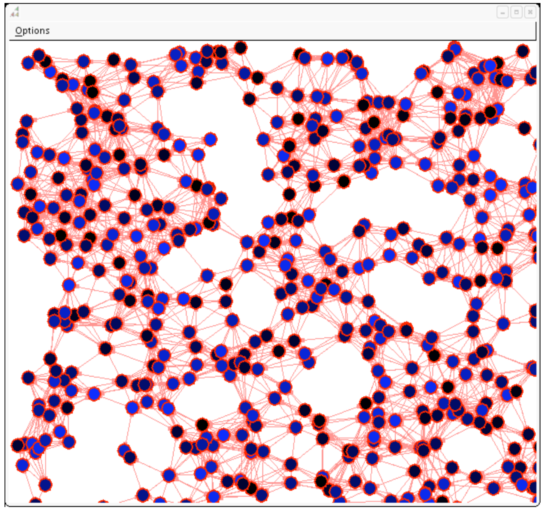


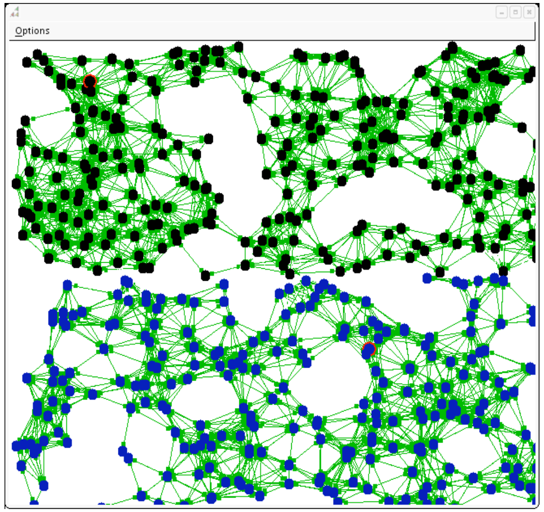
Agents and Pervasive Computing Group

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

Knowledge Organization



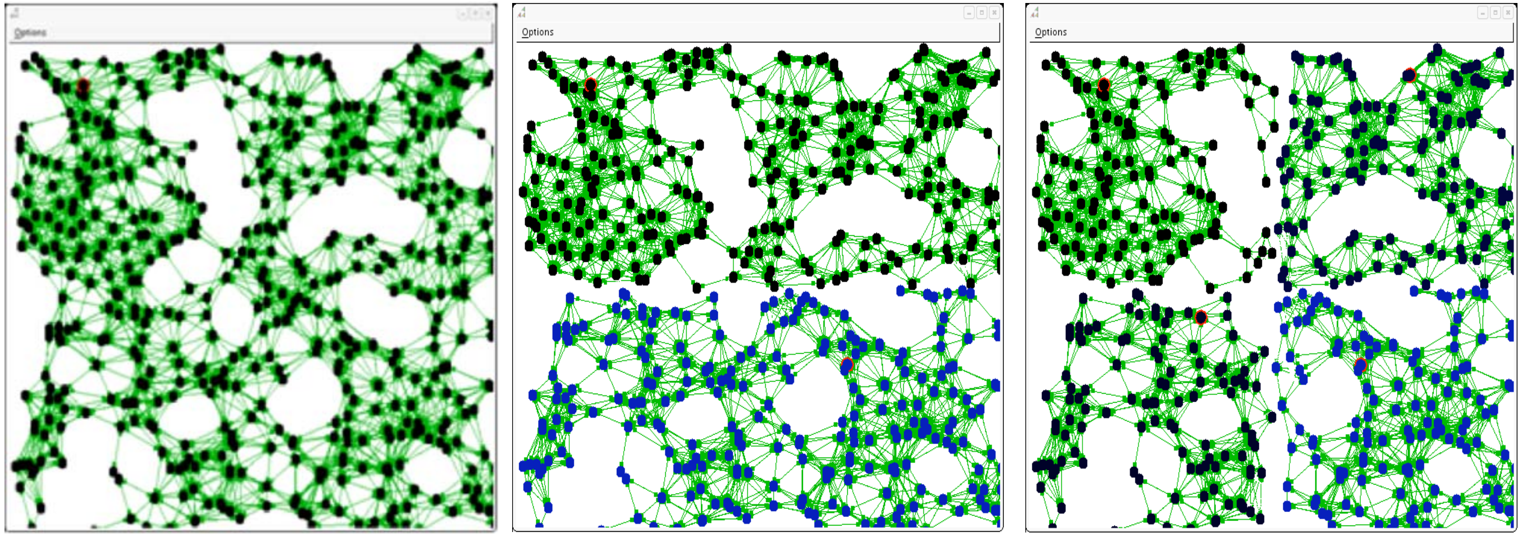


- The concept of region is derived by raw data.
  - Sensor nodes with “similar” readings organize themselves in regions.
  - This result could be considered a particular environment-driven ***knowledge view*** of the environment

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

Multilevel Knowledge Views



- Changing the sensibility level of the algorithm, different **knowledge views** are produced.
  - By this way different services can perceive the most suitable view to reach their goals
  - Multiple levels of observation
  - At higher levels of observation, a region can be

Aggregation on a “per region” Basis

- Identification of region is, per se, aggregated info about some characteristic of the environment
  - But we may also with to gather specific aggregated information
  - i.e., computing specific functions over a region
- Our “macro-programming” solution
  - Enable injecting of specific local self-org aggregation rules in devices (no matter how much and how dense they are)
  - Have these rules execute within the same basic aggregation scheme, at little or no additional costs
  - Have aggregated data be computed

Our Approach: Macro Programming Regions

- Once regions are formed,
  - *Aggregation of sensed data* can occur on a per-region basis
  - Simply by injecting a self-organizing (gossip-based) aggregation function on the network
  - That exploit the existing aggregation noise without incurring in additional communication costs
- Users/services can, on need, be provided with such aggregated data representing some “macro” property of a region at very limited costs

Example: Min Val in a Region

- Inject a simple function to calculate the minimum in a distributed way
  - Have this execute within the aggregation noise loop
  - Similarly for any needed computable data

```
Do_forerever:
    Wait(t);
    neigh[] = Select_neighbor(num_neigh);
    Foreach(neigh[])
        Data = Exchange_data();
        Update_link(data);
        if(connected) Local_aggregation();
Done
```

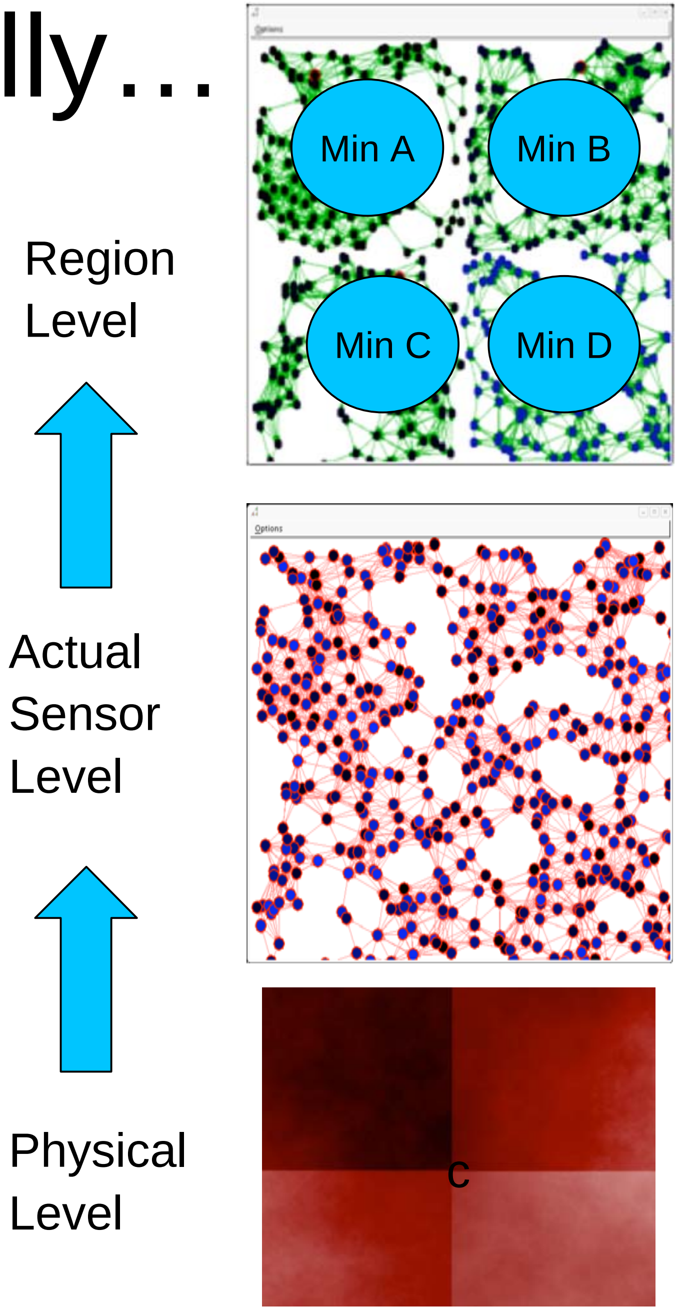
```
Local_aggregation:
    if(localMini>localMinj)
        localMini=localMinj
```

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

And Eventually…

- As in the general figure of the knowledge pyramid…
- We obtain a knowledge network describing some characteristic of the environment from different viewpoints (abstract regions level)
  - Working in a decentralized, self-organizing, self-adaptive way
  - Dealing with variable (potentially continuum) sensor densities



Data Representation

- Once we have algorithms and tools to gather information from the world
  - And we may have huge amount of data pieces representing differen aspects of the environment
- How can we provide a uniform representation of data
  - Coming from different devices
  - Expressing different levels of observations
  - Expressing different facts about the world
- Key guidelines
  - Keep it simple
  - Keep it intuitive
  - Keep it computable
- How do we usually characterize facts about the world?

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

The W4 Context Model

- A simple model in which context data is expressed by a four field structure: *Who*, *What*, *Where* and *When*.
- Someone or something (*Who*) does some activity (*What*) in a certain place (*Where*) at a specific time (*When*)
- Who is acting? What is he/she/it doing? Where and when the action takes place?

Who, What, Where, When

- **Who** is the subject. It is represented by a string with an associated namespace that defines the “kind” of entity that is represented.
  - “person:Gabriella”, “tag:tag#567”, “sensor-region:21”
- **What** is the activity performed. It is represented as a string containing a predicate-complement statement.
  - “read:book”, “work:pervasive computing group”, “read:temperature=23”.
- **Where** is the location to which the context relates.
  - (longitude, latitude),
  - “campus”, “here”
- **When** is the time duration to which the context relates
  - 2006/07/19:09.00am - 2006/07/19:10.00am
  - “now”, “today”, “yesterday”, “before”

W4 Atoms

- Gabriella is walking in the campus' park. An agent running on her PDA can periodically create an atom describing her situation.

Who: user:Gabriella
What: works:pervasive computing group
Where: lonY, latX
When: now

- Gabriella's PDA is connected with a RFID tag reader. A specific RFID agent controls the reader and handles the associated events

Who: tag:statue of Ludovico Ariosto
What: -
Where: lonY, latX
When: now

Simple W4 Queries

- Gabriella is walking in the campus, and wants to know if some colleague is near. She will ask (read operation):

> **Who:** user:*
> **What:** works:pervasive computing group
> **Where**: circle,center(lonY,latX),radius:500m
> **When:** now

- Analogously, Gabriella can ask if some of her colleagues has gone to work in the morning:

> **Who:** user:*
> **What:** works:pervasive computing group
> **Where**: office
> **When:** 2006/07/19:09.00am - 2006/07/19:10.00am

- Key features
  - Partial knowledge (not fully filled atoms)
  - Context-aware queries (“here”, “now”)
  - Possibility of generating atoms merging different sources
    - E.g., and RFID atom generated by merging tagID (who), DBMS data (what), GPS data (where), PDA data (when)
  - Device independence

The W4 Tuple Space

- Atoms can be stored in multiple distributed tuple spaces
  - Overall, these atoms and their relations represents a sort of "world model" expressing all known facts about the world
  - How to distributed data across multiple spaces is an open issue…
- Current (simplistic) approach
  - Cache locally at the collection point (e.g., users' PDA)
  - Forward to a centralized W4 tuple space
- Aggregation algorithms (as the region aggregation noise)
  - Take care of producing, aggregating, and relating atoms
  - And of providing specific views of the world at different levels of information

Self-organized W4 Knowledge Networks

- Knowledge atoms (W4, or whatever) can be related to each other
  - To represent relations between pieces of data and enable navigating related facts of the world
    - Spatial, temporal, or semantic relations
  - To support a better navigation of services in knowledge
  - And more “cognitive” forms of self-organization in services
- Can we exploit self-organization approaches?
  - Self-aggregation of data via semantic/temporal extension of the spatial region approach (cluster and link related data to define multiple and multilevel knowledge views)
  - Have data diffuse (data distribution) and evaporate (data obsolescence) the same as pheromone does in ant-based systems
  - Create knowledge structures that services can perceive as sorts of virtual force fields (context-aware data replication)
  - Matching data patterns and chemical-like reactions (creation of new knowledge)
- A lot of issues to be investigated
  - But ***what service model*** can take advantage of it?

What is a Situated Service for Browsing the World?

- Browsing the world services
  - Location-dependent queries, Social interactions, Real-time understanding of situations, etc.
- Autonomic communication services
  - Distributed cooperation, Ad-hoc communications, etc.

- Given the availability of W4 spaces and data (we assume that there are a multiplicitly of accessible “W4 space” where to store locally produced W4 tuples),
  - we have to code specific software components that can somewhat “query” the W4 tuple spaces for
  - Achieving context-awareness and adapt to the current situation
  - Navigating the world model and extract high-level information about situation occurring in the environment
  - In autonomy

Key Features of Services/Agents

- We require services the capability of
  - querying the W4 space for extracting info
  - injecting new knowledge atoms and/or new aggregation algorithms in the space

```
KnowledgeAtom[ ] read(KnowledgeAtom a);
void inject(KnowledgeAtom a, Behavior b);
```

- In addition
  - Services themselves will be represented by some sorts of atoms in the space
  - Services can thus indirectly interact via the W4 space, which is a sort of

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

Browsing the World Services vs. Ants

- Services produce and read W4 tuples
  - The same as ants release and sense pheromones
- Services indirectly interact via the W4 distributed environment
  - Stigmergy!
- The environment describe something about the environment
  - “There is another agent near here”…the same as pheromones do
- The environment is active
  - Aggregation and pruning of W4 atoms ◊ similar to diffusion and evaporation of pheromones

- We still have to fully unfold these issues, and define a simple and usable agent model…
  - Yet, we have a prototype infrastructure…

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

Browsing the World Architecture

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

Implementation

- The whole system has been realized using the Java language.
- **Web (global accessible) Tuple Spaces** has been implemented through a Postgres database with spatial extensions.
- The **Local Tuple Space** is simply implemented by a Java Vector.
- The RFID reader and the Mote Sensors are accessed via JNI and TCP/IP.
- The **“W4 Query Engine”** interrogates the web accessible tuple space through Sql and its postgis spatial extensions. Local tuple space instead makes use of String parsin and java algorithm.
- User interface is provided by:
  - Google Earth (for laptops) and Google Maps accessed via the Minimo browser (for PDAs).
  - Google KML Language
  - Jsp e JavaScript

Università di Modena e Reggio Emilia

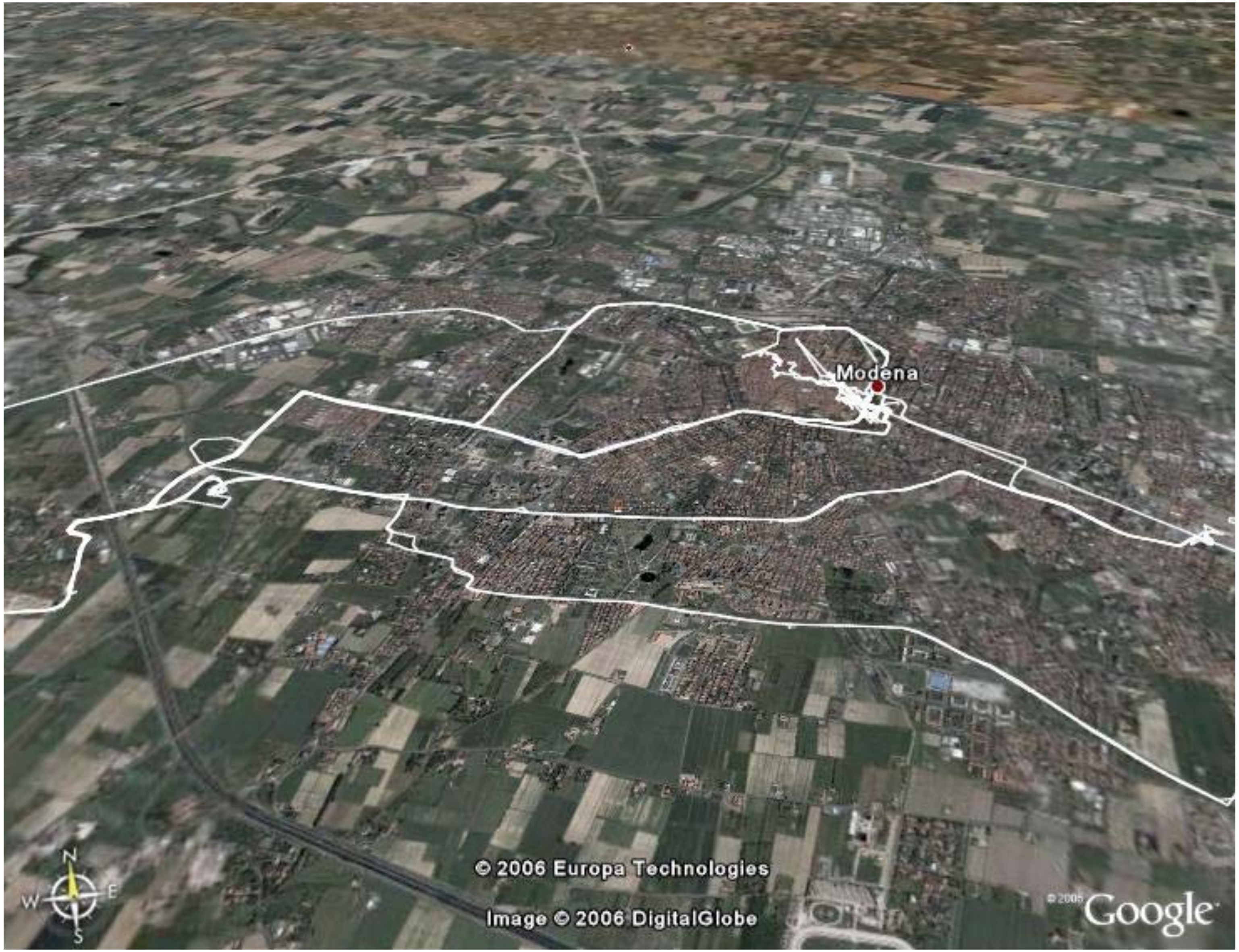
Journey Map

- a tourist wants to automatically build and maintain a diary of his journey:
  - track of all the user movements
  - access available tourist information stored in RFID tags attached to monuments and art-pieces

Who: rfid:*
What: *
Where: *
When: now

Agents and Pervasive Computing Group

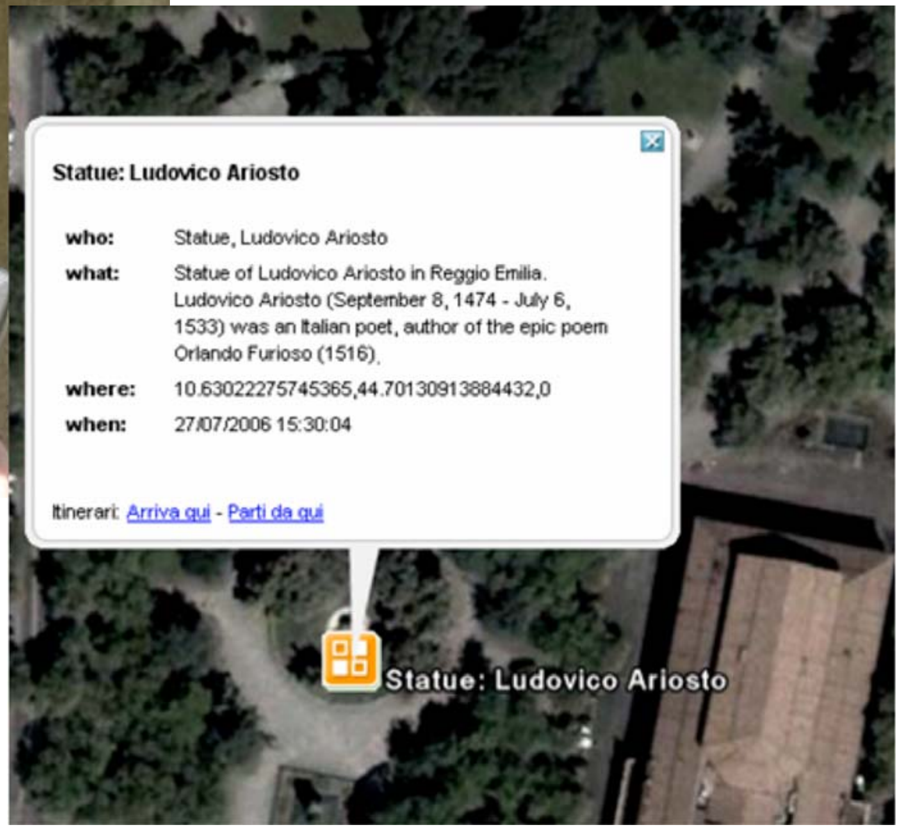
Journey Map


Modena
© 2006 Europa Technologies
Image © 2006 DigitalGlobe
Google
N
S
E
W

Università
di Modena e
Reggio
Emilia

Agents and Pervasive
Computing Group

Journey Map



Università
di Modena e
Reggio
Emilia

The People Map

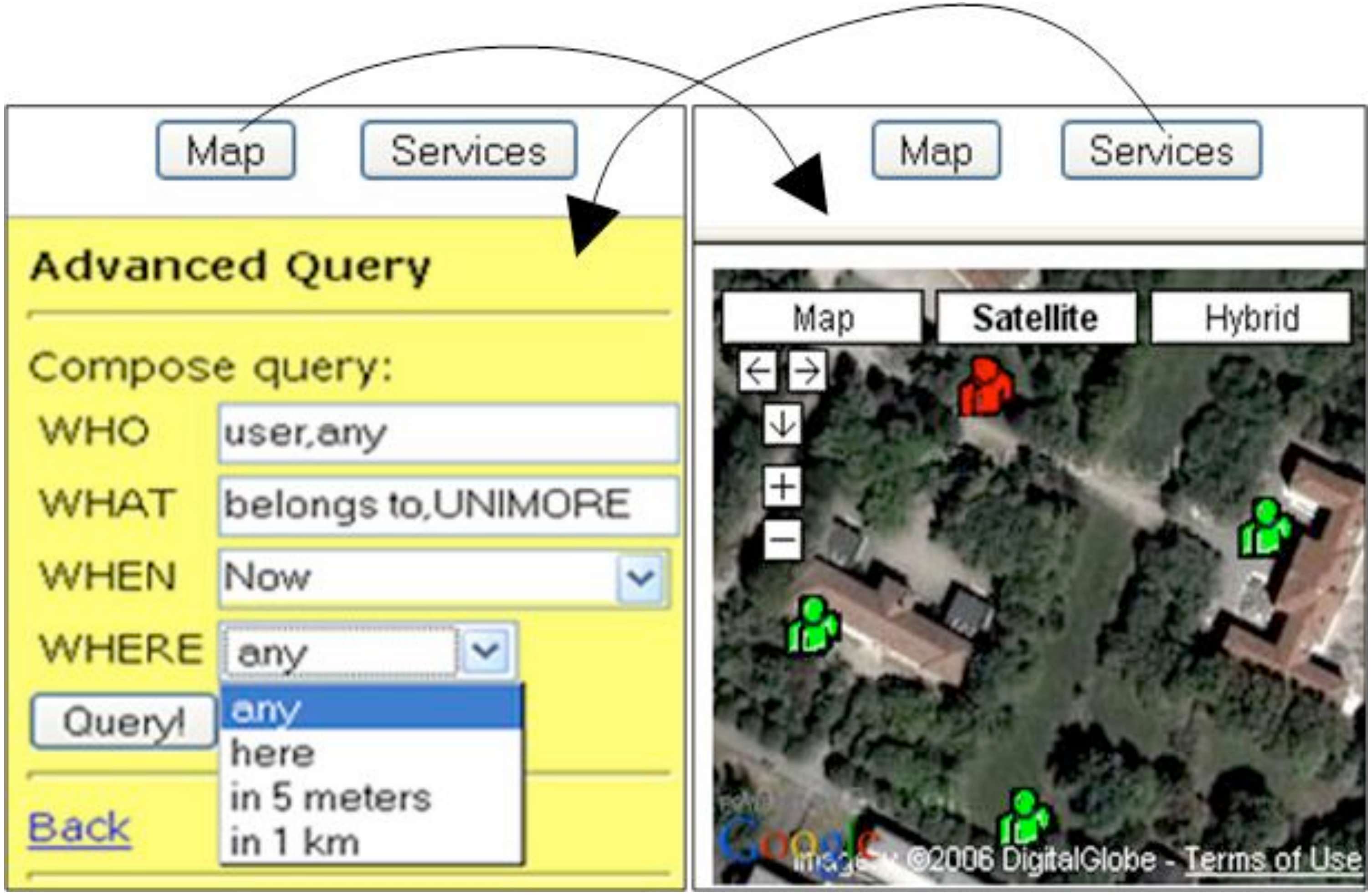
- Group of friends can share their actual GPS locations (represented as knowledge atoms) with each other.
- Collected knowledge atoms can be used to display users' locations on a real-time map (which can also highlight other interesting Web-retrieved information for the group, such as museums or hotels, depending on the specific interests of the group)

Agents and Pervasive
Computing Group

Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

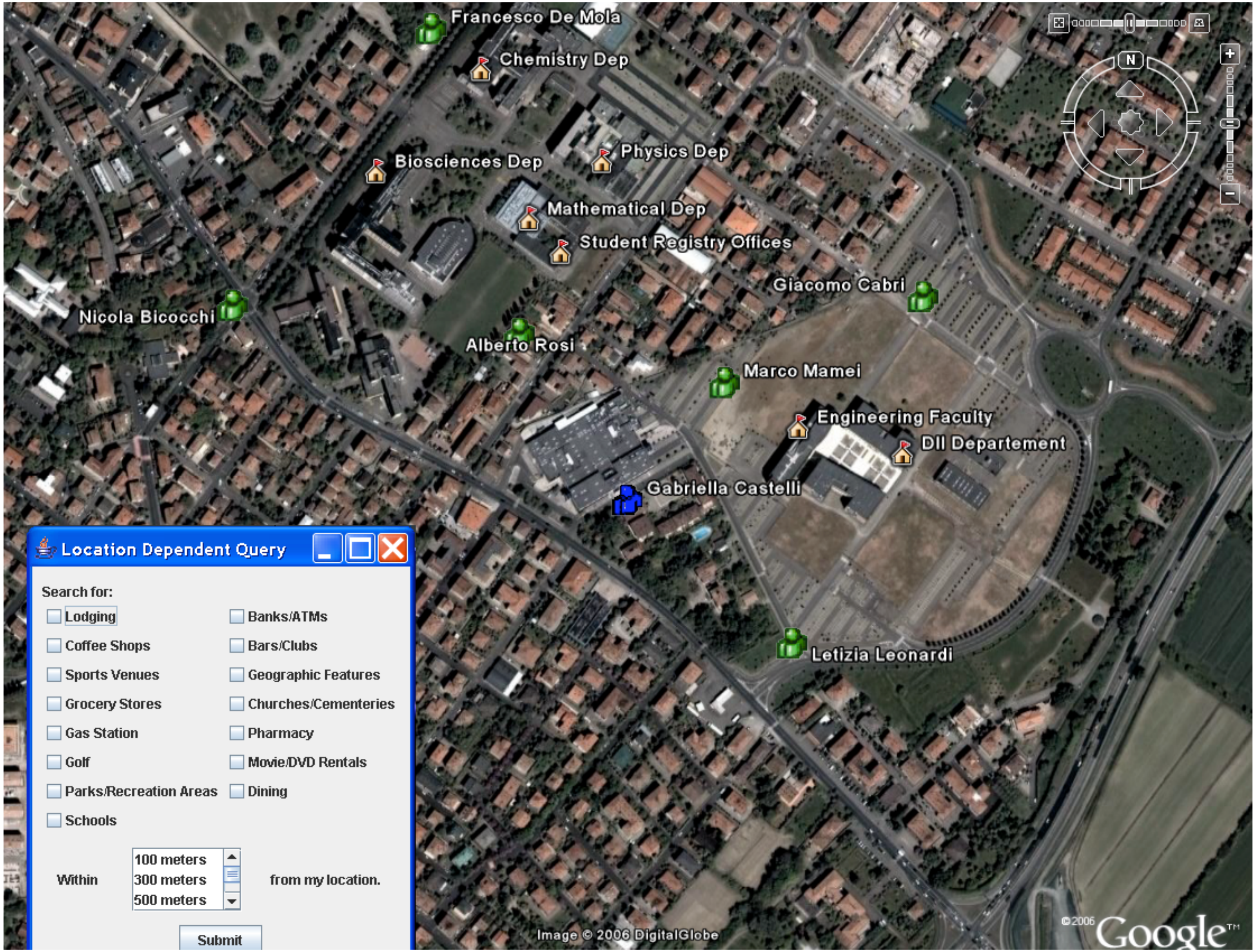
The People Map



Università di Modena e Reggio Emilia

Agents and Pervasive Computing Group

Real Time Map



Conclusions

- Future pervasive computing scenarios invites considering
  - A continuum abstraction for pervasive networks
  - Autonomic services for browsing the world
- Beside our preliminary and incomplete proposals, there is indeed need to study
  - Self-organizing algorithms for multilevel aggregation
  - Proper models for representing, accessing, and integrating heterogeneous contextual data
  - Suitable agent-based model exploiting the above for the provisioning of fully autonomic communication services

References

- N. Bicocchi, M. Mamei, F. Zambonelli, "Knowledge Network Mechanisms for Pervasive Computing Infrastructures", the Handbook of Mobile and Ad-Hoc Networking, CRC Press, 2007
- N. Bicocchi, M. Mamei, F. Zambonelli, "Self-organizing Spatial Regions for Sensor Network Infrastructures", 2nd IEEE Symposium on Pervasive and Ad-Hoc Communications, May 2007.
- G. Castelli, A. Rosi, M. Mamei, F. Zambonelli, "A Simple Model and Infrastructure for Context-aware Browsing of the World", 6th IEEE Conference on Pervasive Computing and Communications, March 2007

Many Thanks To:

- The CASCADAS Project
  - Componentware for Situation-Aware and Dynamically Adaptable Services
  - Funded by EU-FP6-FET: “Situated and Autonomic Communication Initiative”
  - www.cascadas-project.org
- You, for listening