Self-organizing Services for Browsing the World: Challenges and Directions

F. Zambonelli
Agents and Pervasive Computing Group
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, cameras, Web 2.0 data
  - Collecting general purpose data about physical and social world
  - Defining a sort of distributed digital “world model”
  - For general-purpose exploitation by “browsing the world” services
**“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.

---

**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 and 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?
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!!

Engineering the World Model

- Extreme heterogeneity of data
  - Information coming from sensors, cameras, Web 2.0, etc.
  - Variable density of devices (potentially continuum)

- Inherent dynamism and decentralization
  - Devices (and the associated data) come and go at any time
  - No way to control each devices due to decentralization

- 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

- 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
The Knowledge Pyramid

- Data virtually generated in both
  - A sensor network continuum
  - A level of Web 2.0 services
  - At any level in between (e.g., cameras, local 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

- 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

Data becomes knowledge from the services’ viewpoint (something the services know about the world…)

Our Current Research

- Study and experience general abstractions, self-organization algorithms and associated infrastructures to engineer, via self-organization, the world model, so that it can act as a mean way to deploy self-organizing browsing the world services

- In particular:
  1. Algorithms to self-aggregate data to facilitate their exploitation by services → self-organizing spatial regions
  2. Identify a simple yet general-purpose mean to represent context knowledge, information, at whatever level it is produced, so as to build an effective and usable world model → the W4 model
  3. Define a general situated agent model and the associated infrastructure for autonomic communication services for browsing the world → the browsing the world infrastructure
Our Current Research

- Study and experience general abstractions, self-organization algorithms and associated infrastructures to engineer, via self-organization, the world model, so that it can act as a mean way to deploy self-organizing browsing the world services

- In particular:
  1. Algorithms to self-aggregate data to facilitate their exploitation by services → self-organizing spatial regions
  2. Identify a simple yet general-purpose mean to represent context knowledge, information, at whatever level it is produced, so as to build an effective and usable world model → the W4 model
  3. Define a general situated agent model and the associated infrastructure for autonomic communication services for browsing the world → the browsing the world infrastructure

Self-organizing Spatial Regions

- We need to aggregate data from the continuum
  - Compact representation, manageable by services
  - Without losing relevant information
  - And indeed provide more information of “what’s happening”

- Key idea
  - Identify discrete aggregation regions
  - Enable “per region” views of specific characteristics of the environment
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_forever:
    Wait(t);
    neigh[] = Select_neighbor(num_neigh);
    Foreach(neigh[])
        Data = Exchange_data();
        Update_link(data);
    Done
```

```
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)
    }
```

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
Aggregation on a “per region” Basis

- Identification of region is, per se, aggregated info about some characteristic of the environment
  - But we may also wish 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

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: Minimum Value in a Region

```plaintext
Do_forever:
  Wait(t);
  neigh[] = Select_neighbor(num_neigh);
  Foreach(neigh[])
    Data = Exchange_data();
    Update_link(data);
    if(connected) Local_aggregation();
  EndForeach
  if(localMin > localMinj) localMin = localMinj;
Done
```

Local_aggregation:
- if(localMin > localMinj) localMin = localMinj;
And Eventually…

- As in the general figure of the knowledge pyramid…
- We obtain a sort of “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

Our Current Research

- Study and experience general abstractions, self-organization algorithms and associated infrastructures to engineer, via self-organization, the world model, so that it can act as a mean way to deploy self-organizing browsing the world services

- In particular:
  1. Algorithms to self-aggregate data to facilitate their exploitation by services → **self-organizing spatial regions**
  2. Identify a simple yet general-purpose mean to represent context knowledge, information, at whatever level it is produced, so as to build an effective and usable world model → **the W4 model**
  3. Define a general situated agent model and the associated infrastructure for autonomic communication services for browsing the world → **the browsing the world infrastructure**
Data Representation

- Once we have algorithms and tools to gather information from the world
  - And we may have huge amount of data pieces representing different 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?

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.


What is the activity performed. It is represented as a string containing a predicate-complement statement.


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(lon,Y,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

W4 Self-organized Knowledge Networks

- W4 Atoms can be related to each other
  - To represent relations between pieces of data and enable navigating related facts of the world
  - To refer to individual sources of aggregated data and enable vertical navigation (shifting of observation level)

- Can we exploit self-organization approaches?
  - Self-aggregation of data via a semantic extension of the spatial region approach (cluster and link related data)
  - Have data diffuse and evaporate the same as pheromone does in ant-based systems

- Still to be investigated!
Our Current Research

- Study and experience general abstractions, self-organization algorithms and associated infrastructures to engineer, via self-organization, the world model, so that it can act as a mean way to deploy self-organizing browsing the world services

- In particular:
  1. Algorithms to self-aggregate data to facilitate their exploitation by services \( \rightarrow \text{self-organizing spatial regions} \)
  2. Identify a simple yet general-purpose mean to represent context knowledge, information, at whatever level it is produced, so as to build an effective and usable world model \( \rightarrow \text{the W4 model} \)
  3. Define a general situated agent model and the associated infrastructure for autonomic communication services for browsing the world \( \rightarrow \text{the browsing the world infrastructure} \)

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 multiplicity 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

- So it is a **software agent**
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

```java
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

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 \(\rightarrow\) 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…
Browsing the World Architecture

Web information

User centric

Local tuple space for W4

Web Accessible tuple servers

RFID Agent

Sensor Agent

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
Application: The 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

Application: Real Time Maps

- Friends/members of a rescue team can share their actual GPS locations (W4 knowledge atoms)
  - They can be used to see each other
  - And to coordinate their movements with each other and with the environment
Application: Real Time Maps

Conclusions and Future Work

Future pervasive computing scenarios invites considering
- A continuum knowledge pyramid for pervasive networks
- Self-organization of knowledge \(\rightarrow\) Engineering the Environment
- Autonomic services for browsing the world

Beside our preliminary and incomplete proposals, there is indeed need to study
- Self-organizing algorithms for knowledge aggregation
- Proper models for representing, accessing, and integrating heterogeneous contextual data
- Suitable “ant-like” service models exploiting the above for the provisioning of autonomic communication services