UNIVERSITÀ DEGLI STUDI
DI TRENTO

Middleware for Wireless Sensor Networks: An Outlook

Gian Pietro Picco

`disi.unitn.it/~picco`

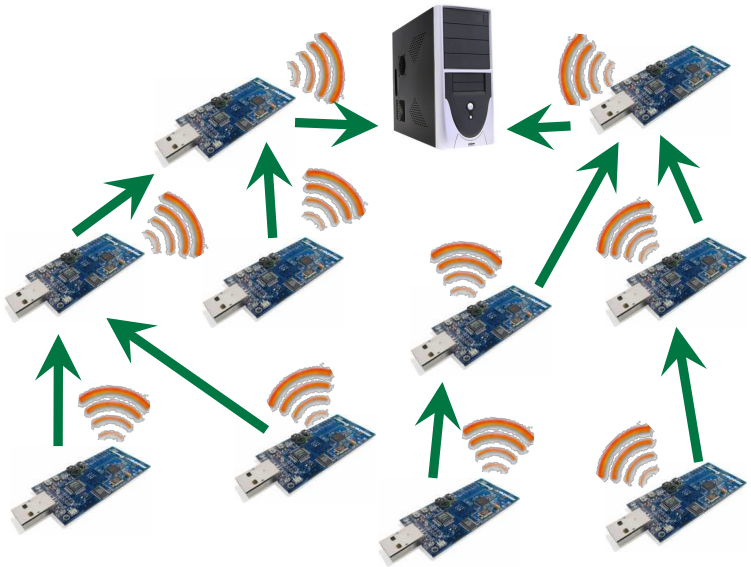
`d3s.disi.unitn.it`

Department of Information Engineering & Computer Science
University of Trento, Italy

joint work with **Luca Mottola**, SICS, Sweden

Wireless Sensor Networks

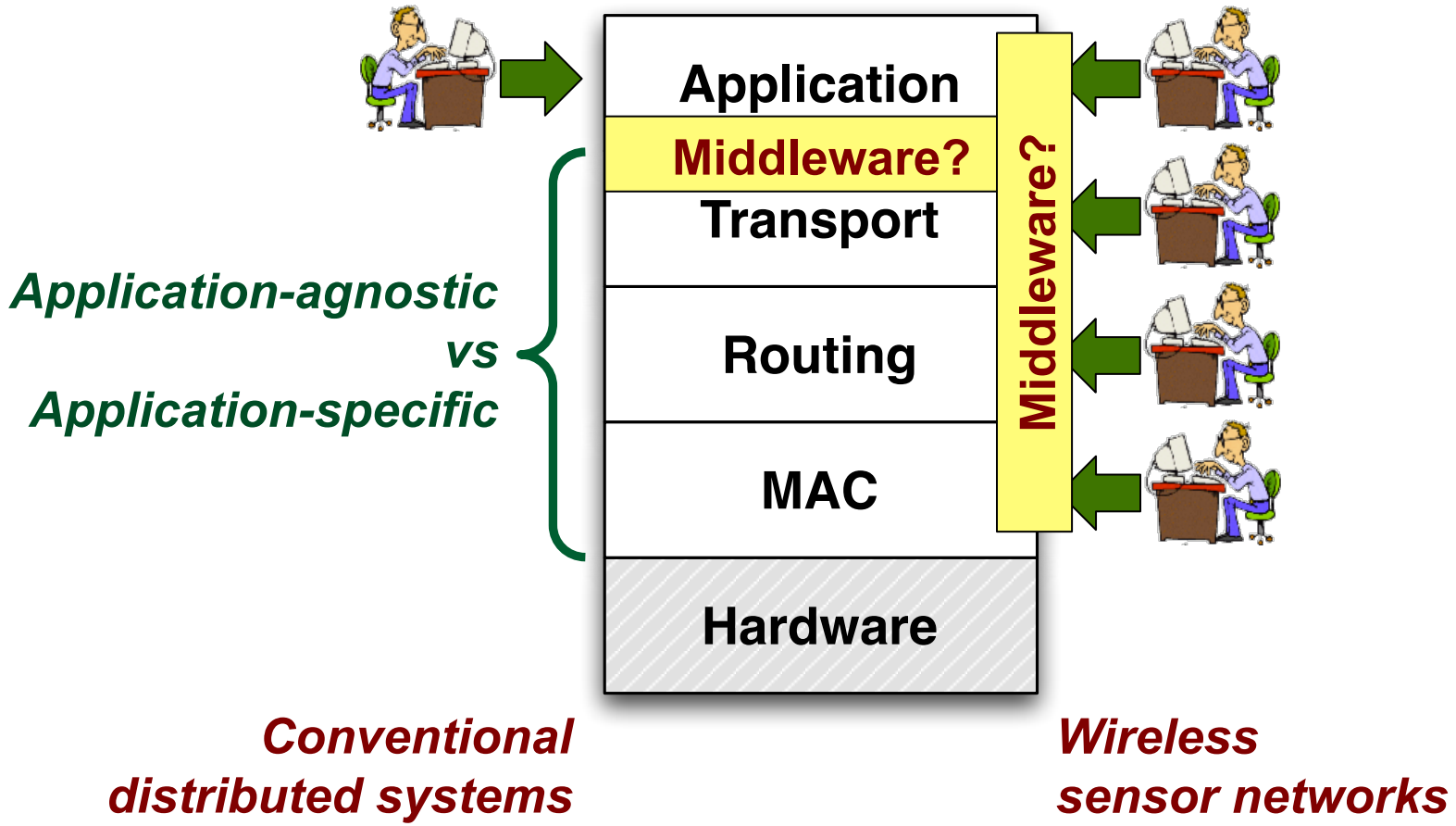
- Enabled by miniaturization of processing, communication, sensing and actuating devices
- Distinctive feature: ***self-organizing*** topology with ***multi-hop*** communication
  - transmit power ≈ distance^2
  - many cheap devices with short-range communication
  - more coverage with less energy (and no wires!)
- Challenges: ***reliability***, ***lifetime***



WSN Hardware and Software

- Example: Tmote Sky (TelosB)
  - TI MSP430 (16 bit RISC)
    - 8 MHz, 10 KB RAM,
      48 KB code, 1MB flash
  - Chipcon CC2420 radio
    - IEEE 802.15.4 compliant, 250 kbits/s
  - On-board antenna
  - Temperature, light, and humidity sensors built-in
- Software development is also stuck in the ’70
  - Applications developed directly on the OS layer
  - code-and-fix approach

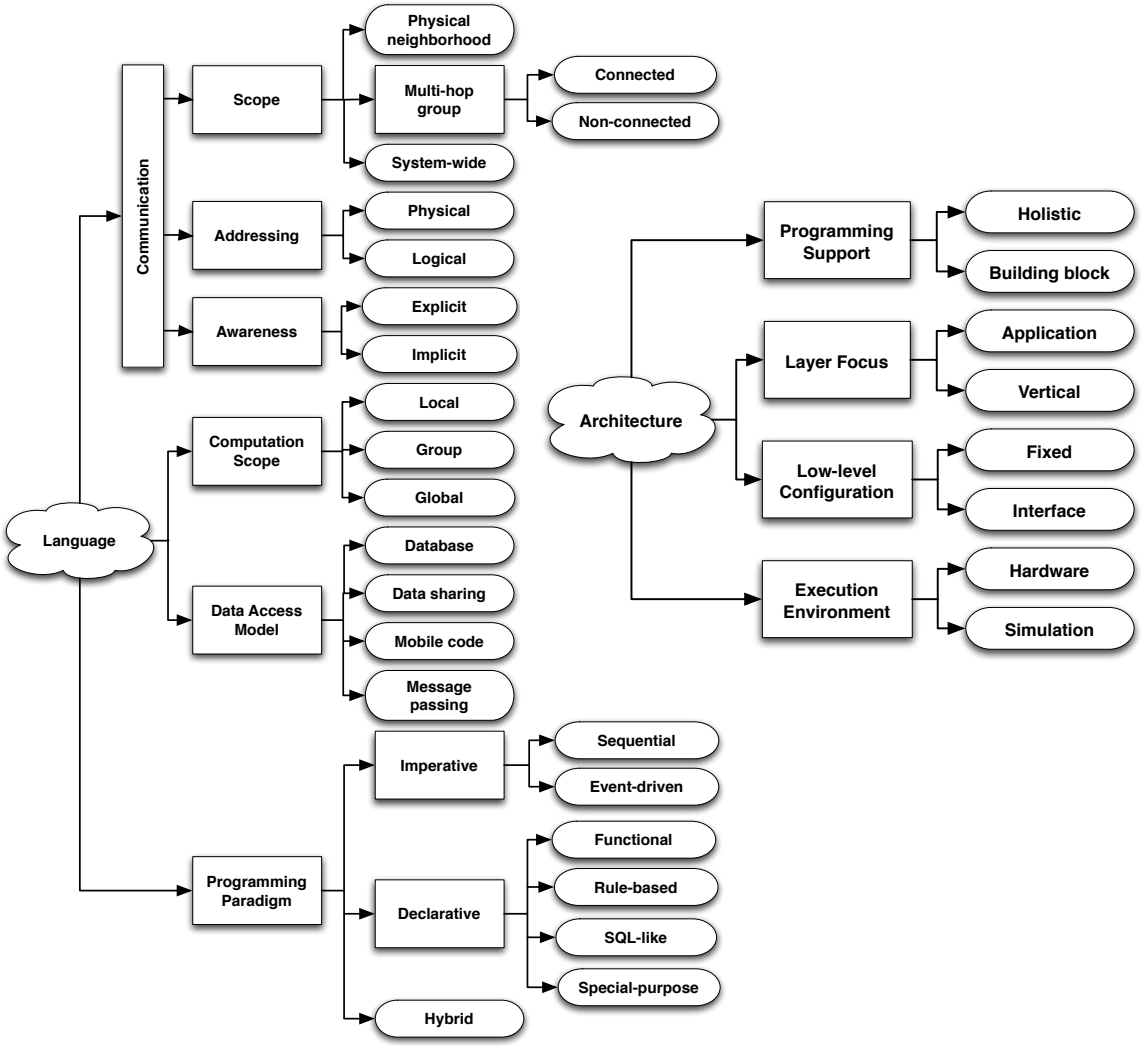
The Role of Software in WSNs



OS Libraries & Components

- ***Operating system libraries***
  - OSes for WSNs provide only very basic functionality
  - System services are effectively provided as an intermediate layer between the OS and the application
    - E.g., data collection and dissemination, time synchronization, localization, …
  - Example: in TinyOS these are built on top of Active Messages, which play a role similar to sockets
- ***Component-based middleware***
  - Focus on dynamic reconfiguration and deployment of application functionality
  - Example: RUNES middleware

Programming Abstractions

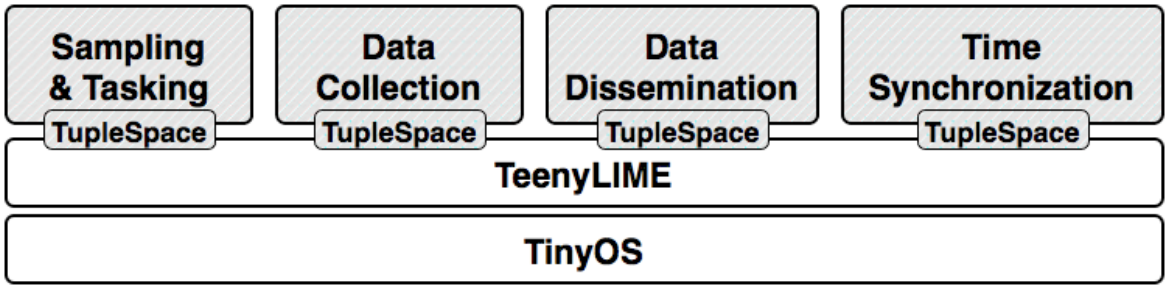


- Many approaches, covering a broad feature spectrum: A "solved problem"?
- In our survey:
  - 28 systems
  - 13 actually implemented on WSN hardware
  - Only 1 used in a real-world, long-term deployment

"Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art"
L. Mottola and G.P. Picco. In ACM Computing Surveys, (43)3, April 2011

TeenyLIME
… *and the Real-World*

- Replaces 1-hop message broadcast with 1-hop ***data sharing***
- Targets both ***application-*** and ***system-level*** functionality



"Programming Wireless Sensor Networks with the TeenyLIME Middleware"
P. Costa, L. Mottola, A. Murphy, and G.P. Picco. In Proc. of the 8th Int. Conf. on Middleware, 2007

Active Messages vs. TeenyLime

Or Why Abstractions are Useful…

```
bool pendingMsg;
TOS_Msg sendMsg;
event TOS_MsgPtr ReceiveInterestMsg.receive(TOS_MsgPtr m) {
  struct InterestMsg* payload = (struct InterestMsg*) m->data;
  if (isRecipient(payload, TOS_LOCAL_ADDRESS))
    insertInterest(payload->sender, payload->type, payload->threshold, payload->timestamp);
  return m;
}
event result_t TemperatureSensor.dataReady(uint16_t reading){
  if (!pendingMsg && matchesInterest(reading)) {
    atomic {
      pendingMsg = TRUE;
      struct DataMsg* payload = (struct DataMsg*) sendMsg->data;
      msg->sender = TOS_LOCAL_ADDRESS;
      msg->type = TEMPERATURE_READING;
      msg->value = reading;
      if
        p
    }
  }
  return SUCCESS;
}
event result_t SendDataMsg.send
  if (msg == sendMsg)
    pendingMsg = FALSE;
  return SUCCESS;
}
```

R,sizeof(struct AppMsg),&sendMsg)!= SUCCESS)

Acquires a temperature reading and brodcasts it to interested neighbors

Same functionality in TeenyLime

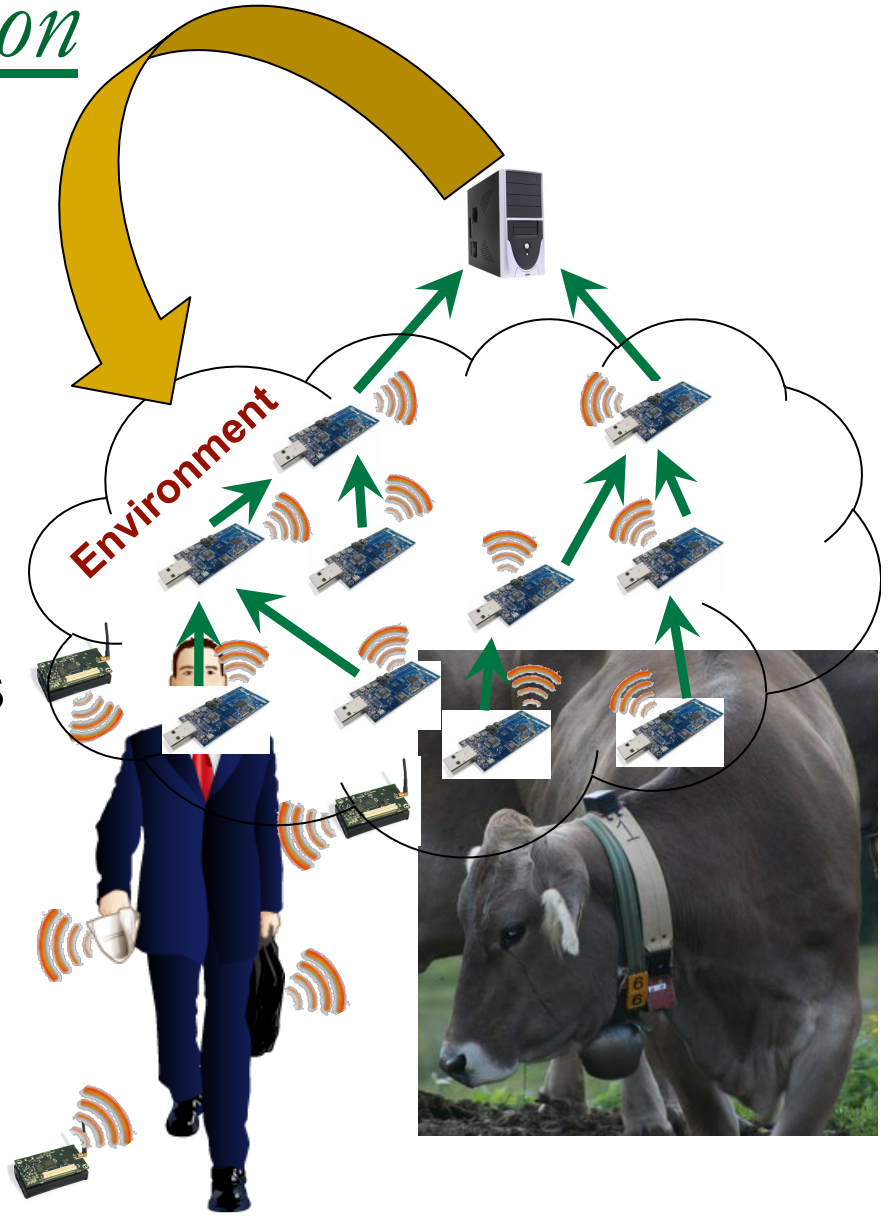
```
event result_t TemperatureSensor.dataReady(uint16_t reading){
  tuple<uint16_t, uint16_t> temperatureValue = newTuple(
                                actualField(TEMPERATURE_READING),
                                actualField(reading));
  call TupleSpace.out(FALSE,TL_LOCAL,&temperatureValue);
  return SUCCESS;
}
```

Open Research Challenges

Finding the Right Abstraction…

… *for the right application*

- Today's focus:
  - many-to-one interactions
  - sense-only applications in fixed-node WSNs
- Tomorrow:
  - many-to-many interactions
  - in-network intelligence
    - Cyber-Physical Systems
  - opportunistic interactions
    - mobile WSNs, Internet of Things



Finding the Right Abstraction…

… *for the right developer*

Scientist, engineer, domain expert

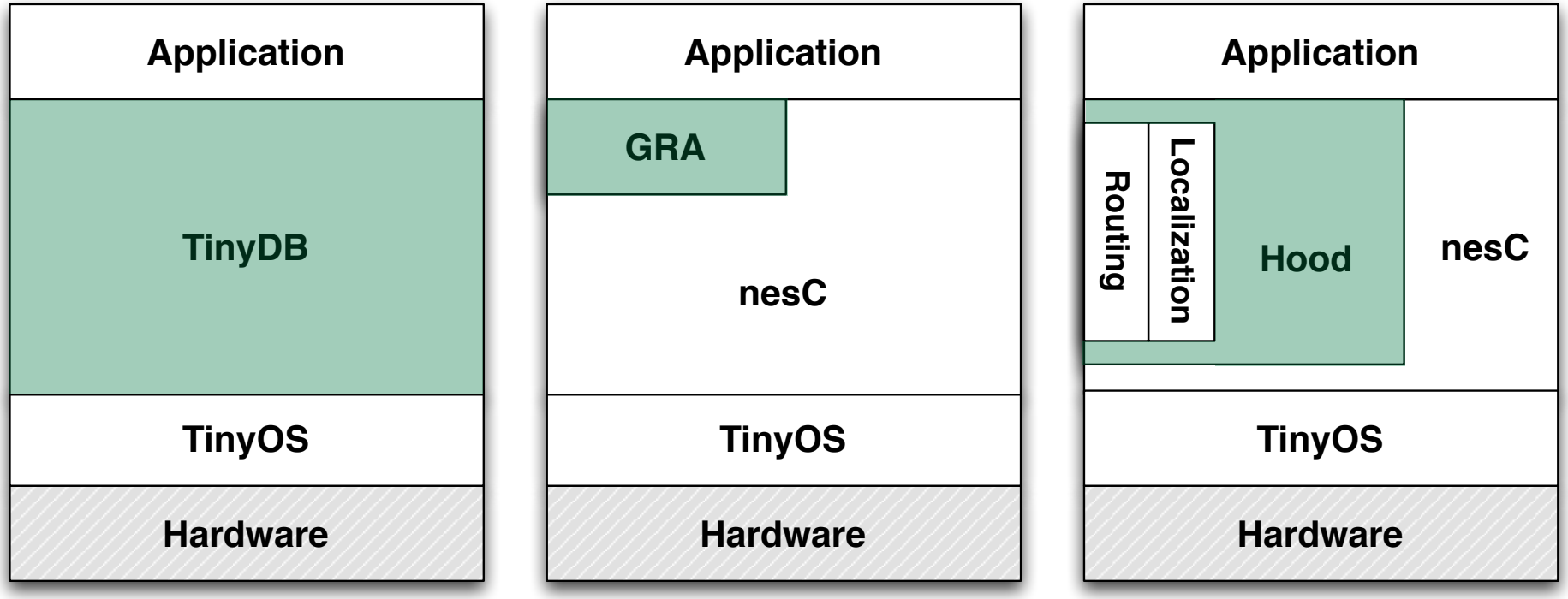
- Wants "good" data!
- Sees the network as a whole
- Knows (more than you'd expect) about basic IT

- Wants "good" yield!
- Sees the behavior of the individual nodes
- Can deal with complex languages/systems and their interaction

WSN geek

We need to support both!
(and probably someone in between)

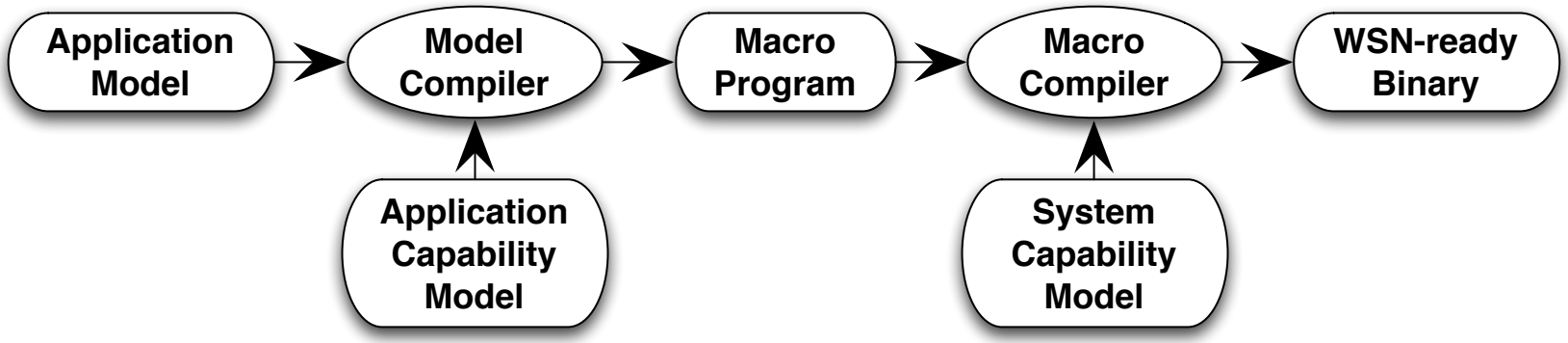
Language vs. Middleware, Monoliths vs. Components



- The “right” architecture is likely to depend on the intended developer

Joining the Flow
Integrating WSNs into the Mainstream

- At best, WSNs are seen as a monolithic “service”
- CPS and IoT scenarios are pushing WSNs to become part of mainstream business processes
- How to reconcile (and integrate) business process technology with the peculiarities of WSNs?



>makeSense
Easy Programming of Integrated Wireless Sensor Networks

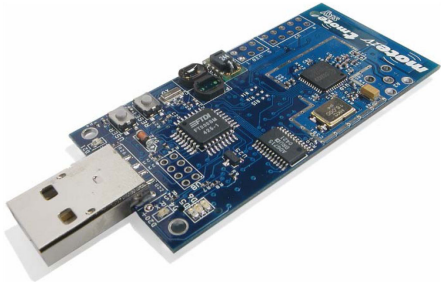
UNIVERSITÄT ZU LÜBECK
UNIVERSITÀ DEGLI STUDI DI TRENTO
SAP
SICS
acciona
SEVENTH FRAMEWORK PROGRAMME

Other Research Challenges

- ***Cross-layering***
  - Advocated as a must in WSNs, to guarantee performance in spite of scarce resources
  - In practice, no WSN middleware fully supports it
- ***Supporting multiple concurrent applications***
  - Will become more and more important as the Internet of Things becomes commonplace
- ***Enabling reliable, predictable implementations***
  - Provide well-defined failure modes
  - Middleware is the right level of abstraction for verification
- ***"How good is my middleware?"***
  - Often answered only in terms of performance
  - Impact on programming practice overlooked

Tactics vs. Strategy

… *Hardware and OS: Cozy or Adventurous?*



+ TinyOS =

?

- Experiment with more resource-rich platforms
  - Anticipating future hardware developments
  - “System” challenges vs. “abstraction” challenges
- Beware: unlike with PCs, it’s not just a hw upgrade
  - The software must be (heavily?) adapted, too

Tactics vs. Strategy

"Have You Ever Heard of ZigBee?"

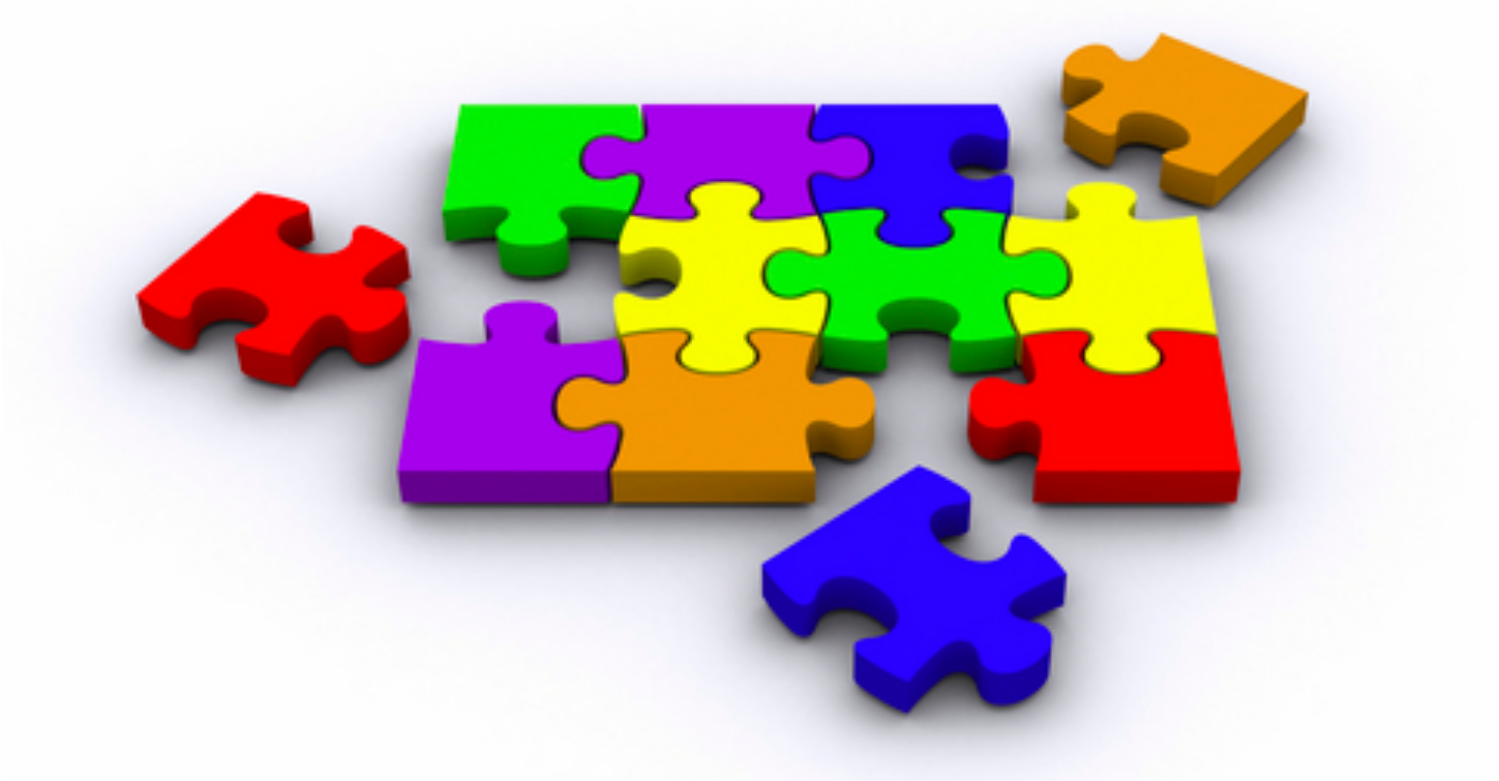
TinyOS

ZigBee® Alliance

- Conflicting goals
- Clean-slate research vs. standard compliance
- In which playground do you want to play?

Tactics vs. Strategy

Is Interoperability an Issue?



- No middleware provides support for multiple OS
- Interoperability is a not an issue … today

Let's Not Forget the Final Goal …

- To make an impact, middleware must sustain the challenges of real-world deployments
  - Just like networking protocols did…
- In-field validation is not enough:
  - tradeoffs between research speculation and practical issues must be resolved early on

A Different Mindset

- Designers of WSN middleware must “get their hands dirty” with the networking stack
- A broad blend of competences is required, currently scattered across different research communities…
  - … sometimes not speaking to each other
- A concerted effort is ultimately needed