Management of Dynamic Context Models with Data Stream Processing

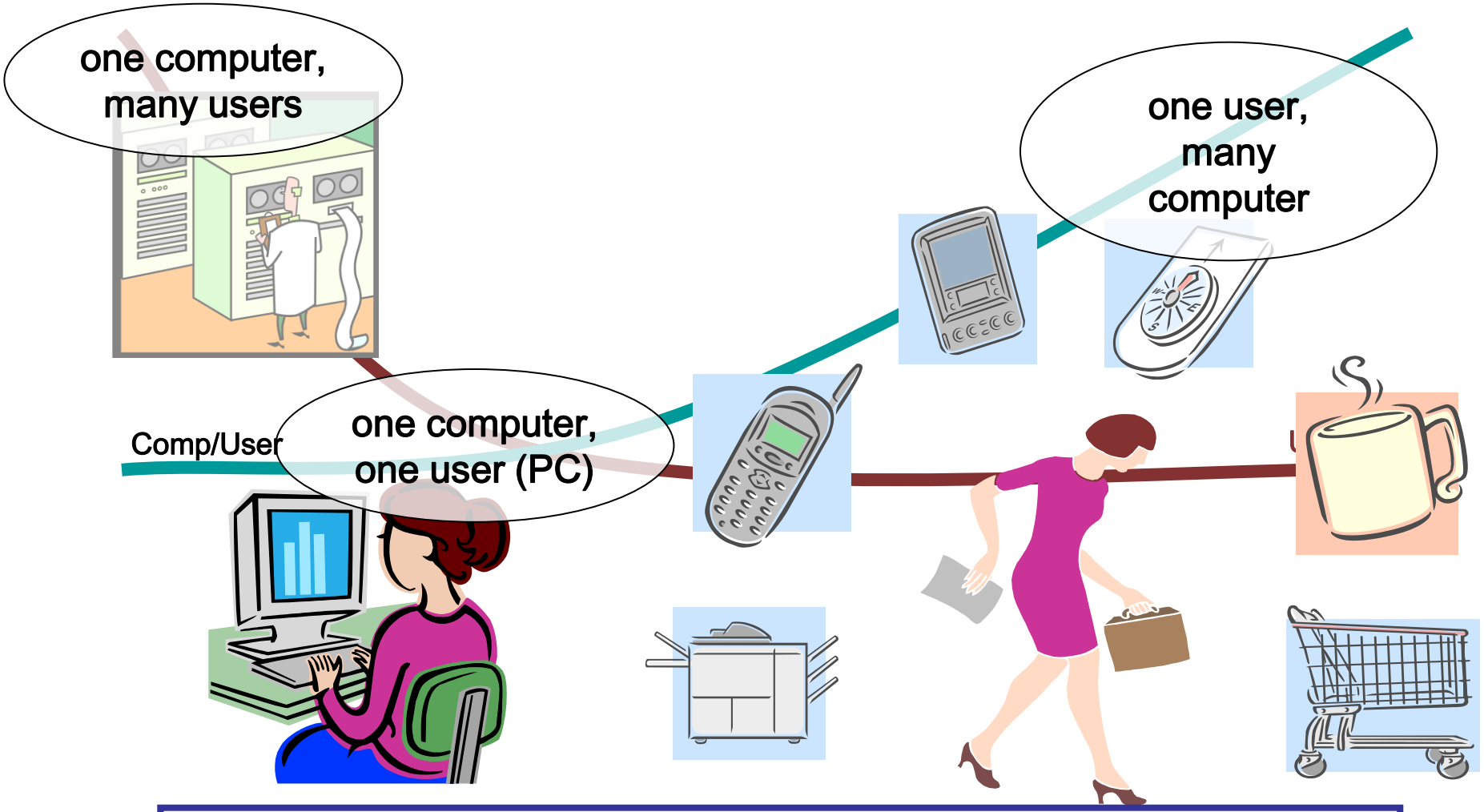
2nd International Workshop on Information Management for Mobile Applications in conjunction with VLDB 2012

Prof. Dr. Daniela Nicklas

Department für Informatik
Juniorprofessur Datenbank- und Internettechnologien
Carl von Ossietzky Unversität Oldenburg

dnicklas@acm.org

A Clear Trend

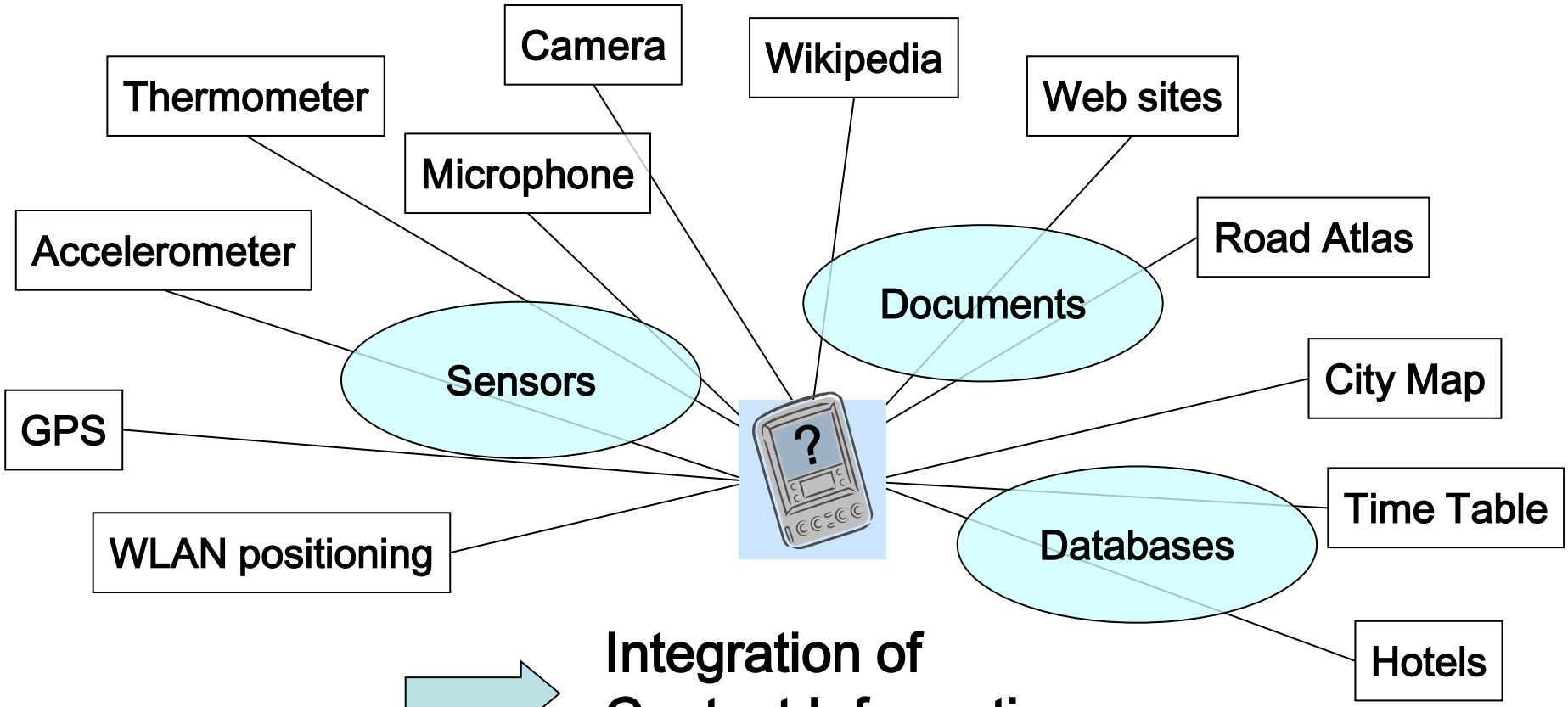


Overview

- Introduction
  - Context and context-aware applications
  - Data stream processing
- Context Management
- Dynamic Context Management
  - two application examples:
    - Sensor fusion for autonomous vehicles
    - Safe offshore operations

- Conclusions

Available Context Information



What is Context? -- The "Deyfinition"

Context is any information that can be used to characterize the situation of an entity.

An entity is a **person**, **place**, or **object** that is considered **relevant to the interaction** between a user and an application, **including the user and applications** themselves.

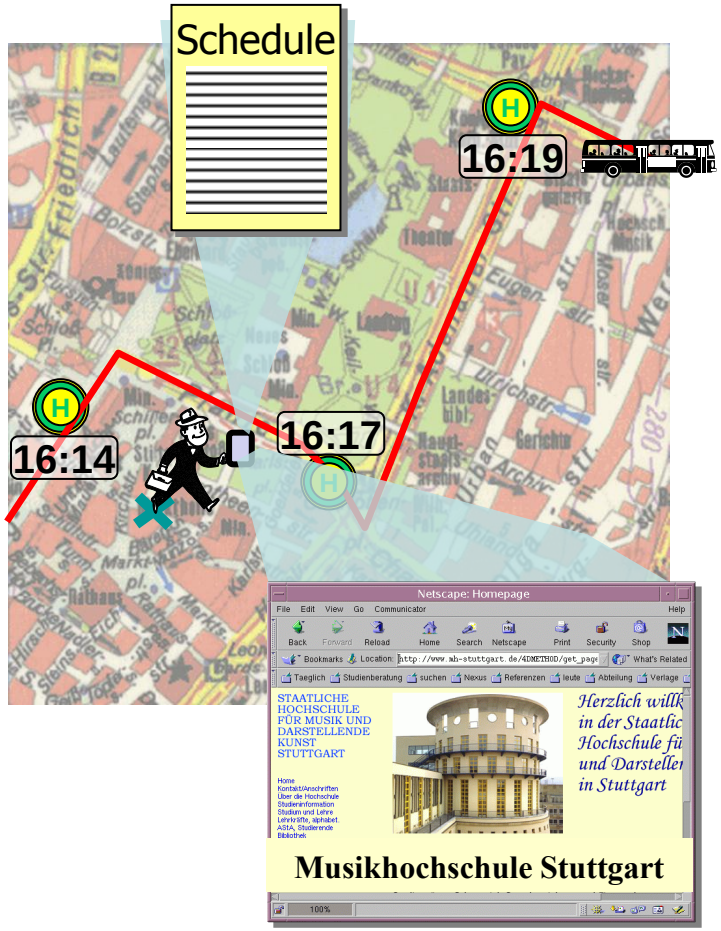
A system is context-aware

if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user‘s task.

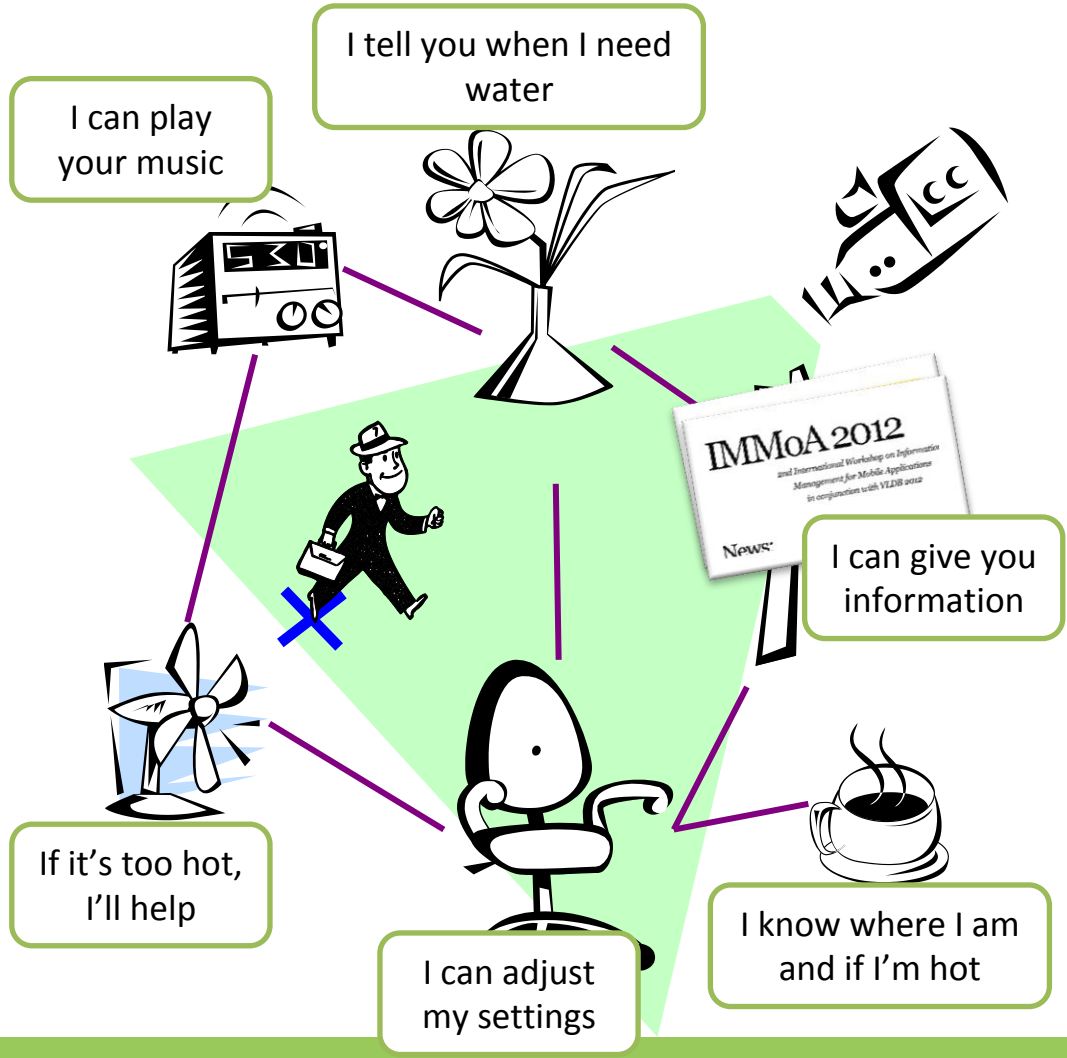
A. K. Dey. Understanding and using context. Personal and Ubiquitous Computing, 5(1):4–7, 2001.

Typical Context-Aware Systems

mobile computing



"smart" environments (AKA Internet of Things)



More examples … (of my current interest)

Crew
Congestion
Snow
Pedestrians
Wind
Goods
Cargo
Trucks
Processes
Parcels
Sea
Obstacles
Ships
Ground
Lanes
Products
Cars
Workers
Roads
Machines

Safe Off-shore Operations

Traffic monitoring

Logistics

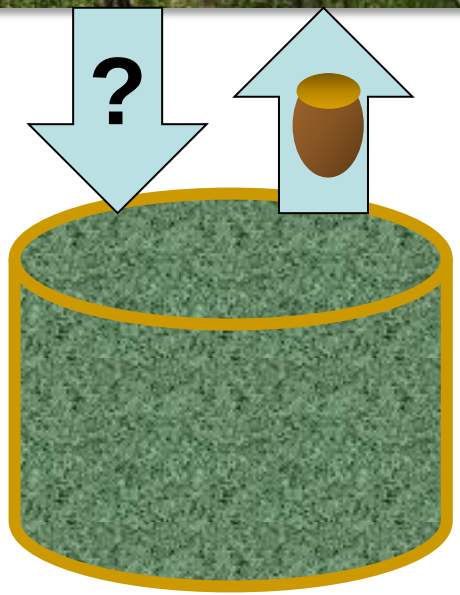
Application Areas

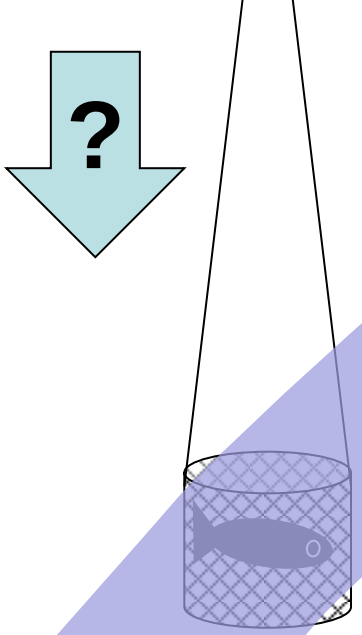
Vessel Traffic Management

Driver assistant systems

Smart Factory

… and many more!





Static and Dynamic Data Management …

Data Base Management Systems

- „store, then process“
- analytical queries
- one-time queries
- well-known technology
- user/application is active, data management passive

Data Stream Management Systems

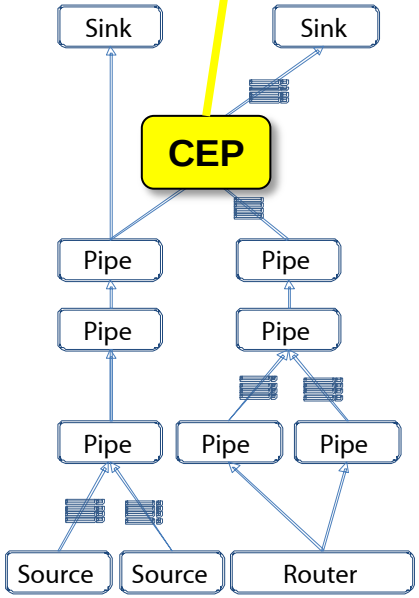
- „on-the-fly“ data processing
- real-time reaction on changes
- store only what needs to be kept
- register long-running queries
- user/application is passive, data management pushes data
- Complex event processing for temporal patterns

Features of Data Stream Management Systems

- Programming Abstraction
  - declarative: query
  - functional: flow graph
  - → enables optimizations
  - → better maintanance of systems
  - → using a DSMS on data streams is like using a DBMS instead of files
- Data flow vs. event bus (as in many CEP engines)
  - execution of data flow processes only data items that are needed by a query
- Parallel execution of operators in graph
  → no shared memory
- Data streams can be unbounded:
  - issues with sorting, joins, aggregation
  - → approximate answers
  - → window semantics

```
SELECT ego.pos
RANGE 10 secon
radar RANGE 15
WHERE ego.speed > 30 AND
radar.speed > 30
AND s2.pos - ego.pos < 15
```

Some DSMS provide CEP operators



However …

Data Base Management Systems

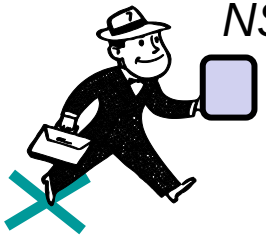
- ACID properties, persistant storage
- well-established semantics (rel. algebra)
- well-established optimizers
- often deeply integrated into enterprise IT infrastructure
- can also cope with dynamic data management (triggers, repeated optimized queries, continuous queries, compiled queries)

Data Stream Management Systems

- main memory processing
- often non-deterministic results (depend on timing of streams)
- no standard query semantics (often relational algebra with extras, or many user-defined operators)
- new system in IT infrastructure
- has often to be combined with static data management

Context Management

How to get the context?

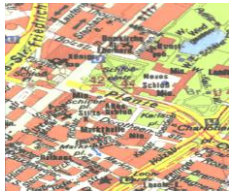


NSASG – not so annoying shopping guide

situation	user is accepting shopping recommendations			
related context	near by: San Pietro, Musei Vaticani, Borgo A. Angelo, shopping sites	opening hours	user's speed	user's timetable
	inside Piazza San Pietro	afternoon		
context	location of user	current time		
data type	WGS84 coordinates	GML Timestamp		
data	9,175; 48,7826	2005-10-18T20:47:00.000		
sensors	GPS-Sensor	click on map	…	system clock	gps clock

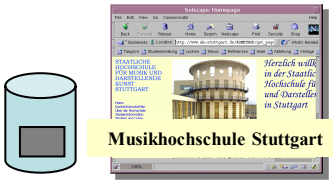
Context types

- Geographic context: map data
  - Streets, buildings, land marks, points of interest, ...
  - Data source: geographic information systems (GIS)
  - Stationary objects, rarely changing

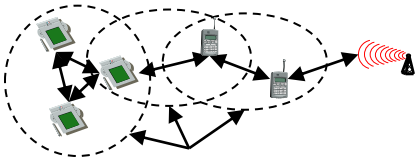


- Dynamic context: movement and change
  - People, vehicles,traffic situation, weather, ...
  - Data source: sensors

- Information context: digital world, "cyberspace"
  - web sites, documents, game objects, ...

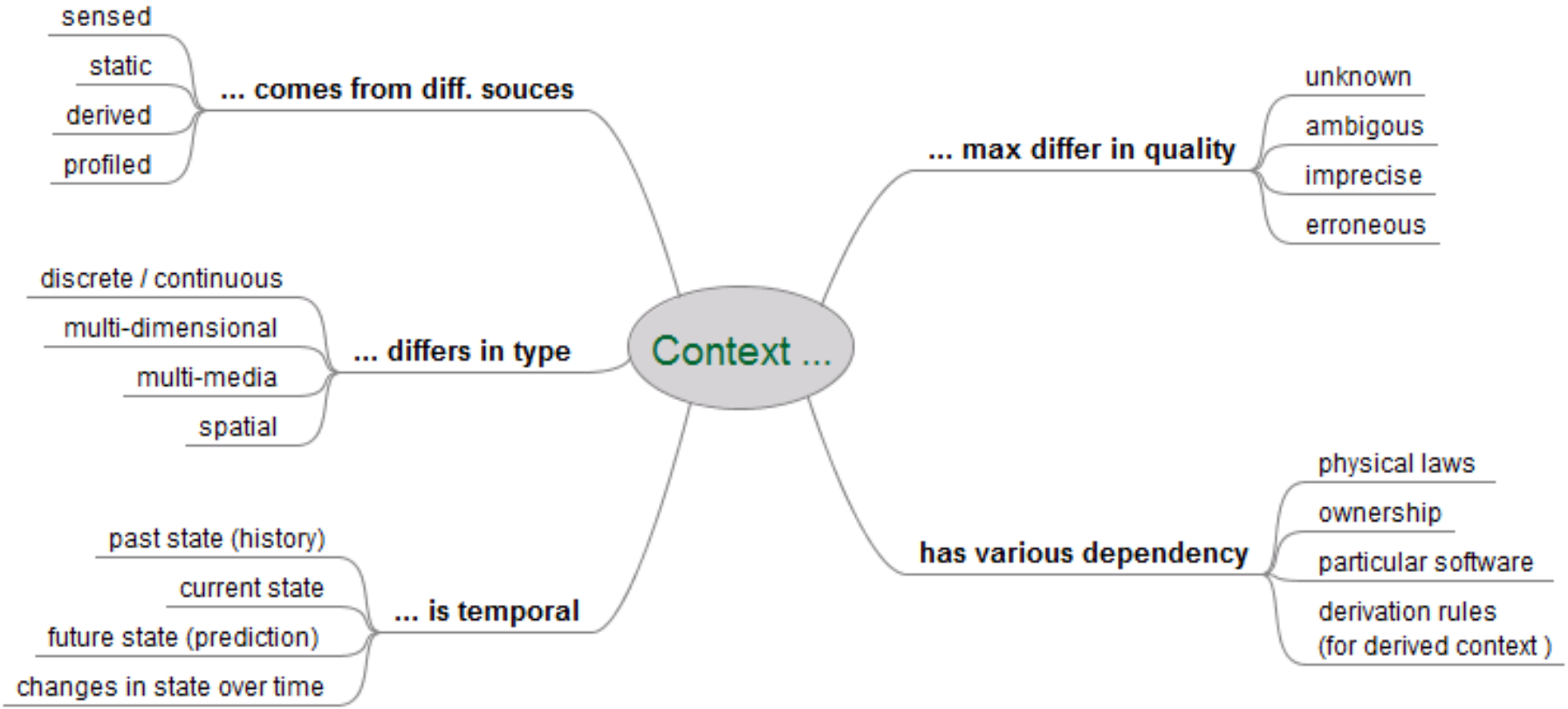


- Technical context: infrastructure
  - access networks, topology, services, (printer, projector, ...)



- User context
  - activities, plans, preferences
  - often derived from dynamic context (e.g., camera, accelerometer)
  - highly sensitive!

Context characteristics



Context characteristics

- Context information is heterogeneous
  - discrete / continuous
  - multi-dimensional
  - multi-media
  - spatial
  - sensed / static / profiled / derived
- Context information may differ in quality; may be
  - unknown
  - ambiguous
  - imprecise
  - erroneous

- Context information differ in
  - type
  - information quality
  - temporal characteristics

- Sensed Context: Low persistence
  - may be inaccurate, unknown, or stale
  - source of errors
  - sensor failures
  - network disconnections
  - delays (in communication or processing)

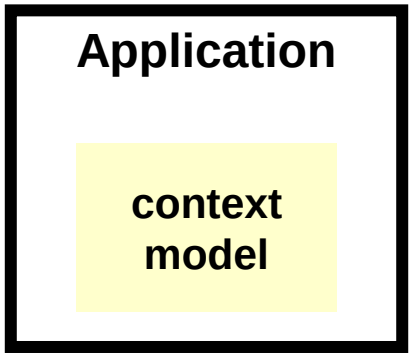
Context characteristics

- Context information needs temporal meta data
  - past state (history)
  - current state
  - future state (prediction)
  - changes in state over time

- Context has various dependencies
  - physical laws
  - ownership
    - who owns devices
    - which computers have a license to run
  - particular software
  - derivation rules for derived context

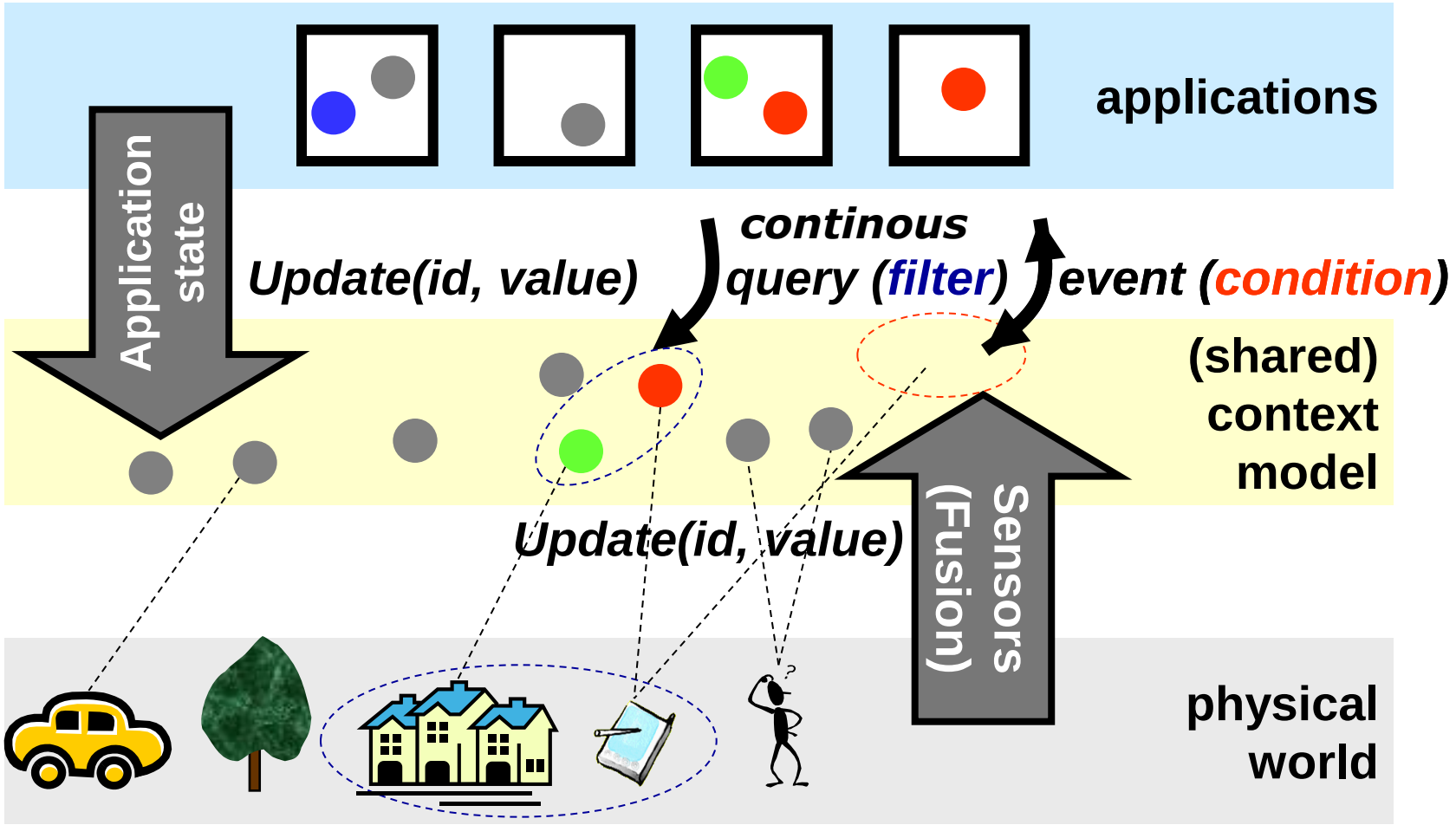
Local context model or shared

- Local context model:
  - context management for just one application
  - design of components up to the software developer, but it's a good idea to seperate concerns:
    - from sensors to data
    - from data to context
    - from context to situation



- Shared context model:
  - common context management for several applications
  - design of components depends on „sharability"
  - can save tremendous amount of development time if several applications work in similar/overlapping (by space / by content) contexts

Abstraction: Shared Context Model



Anatomy of a context-aware application

and „sharability“ of context information

Application-specific

Standards for aggregation methods

Application	what the user / **other systems / the world** see
Adaptation	how the application's behaviour changes
Situation	in what cases does adaption happen
Context	any information that can be used to characterize the situation of an entity ["Deyfinition"]
Data	data (e.g., from sensors, content) used by the system to determine context information

Easy to share

Good for „views“

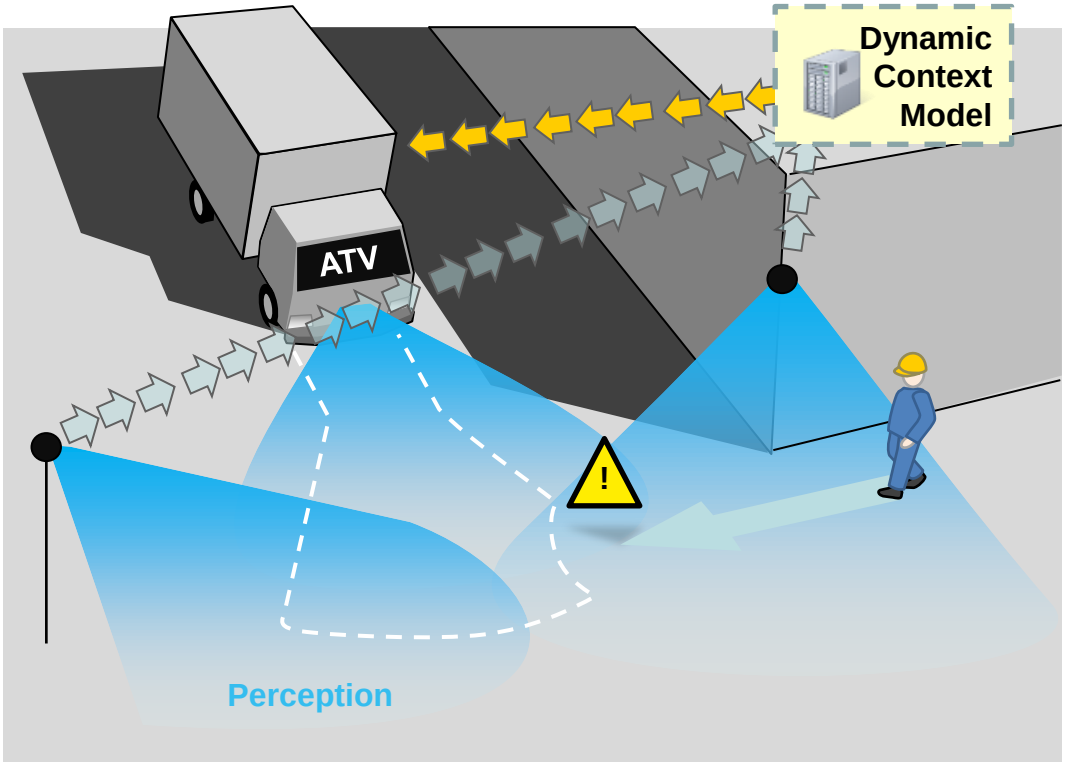
Enable domain-specific standards

Dynamic Context Managment

- within two applications

Sensor Fusion Example (project SaLsA)

- Context-Model Generation for Safe Autonomous Transport Vehicles
  - Autonomic vehicles perform mobility operations with walking speed to ensure safety
  - Faster mobility requires specific knowledge of the environment
  - Realization via a dynamic context model based on sensor data with uncertainties
  - Using Data Stream System (DSMS) for Sensor Fusion



Sensor fusion example: Sensors

- LMS100:
  - Indoor
  - Scanning Frequency: 25Hz / 50Hz
  - Resolution (Degree) : 0.5° - 0.25°
  - Operating Range: 0.5m … 20m
  - Field of View: 270 °
- LMS151:
  - Outdoor
  - Scanning Frequency: 25Hz / 50Hz
  - Resolution (Degree) : 0.5° - 0.25°
  - Operating Range: 0.5m … 50m
  - Field of View: 270 °

www.mysick.com

Sensor fusion example: Processing

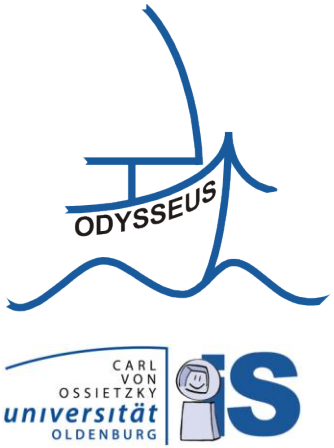
- Merge Operator:
  - Calculates the occupy probability of each cell in the occupancy grid along each laser beam using a Bayesian Updater.
  - Number of laser beams depend on application settings:
    - 540 Beams (0.5 °) / 1080 Beams (0.25°)
  - Resulting in up to 54,000 points per second per sensors
- Spread Operator:
  - Realized using a Kernel-Operation in OpenCV to summarize the probabilities of each grid cell in the log domain

- Current approach requires ~150ms for one occupancy grid using an old Intel Core 2Duo CPU.
- However, each beam can be processed in parallel.

Data Stream Management by Odysseus

- Flexibe Open Source Data Steam Management Framework

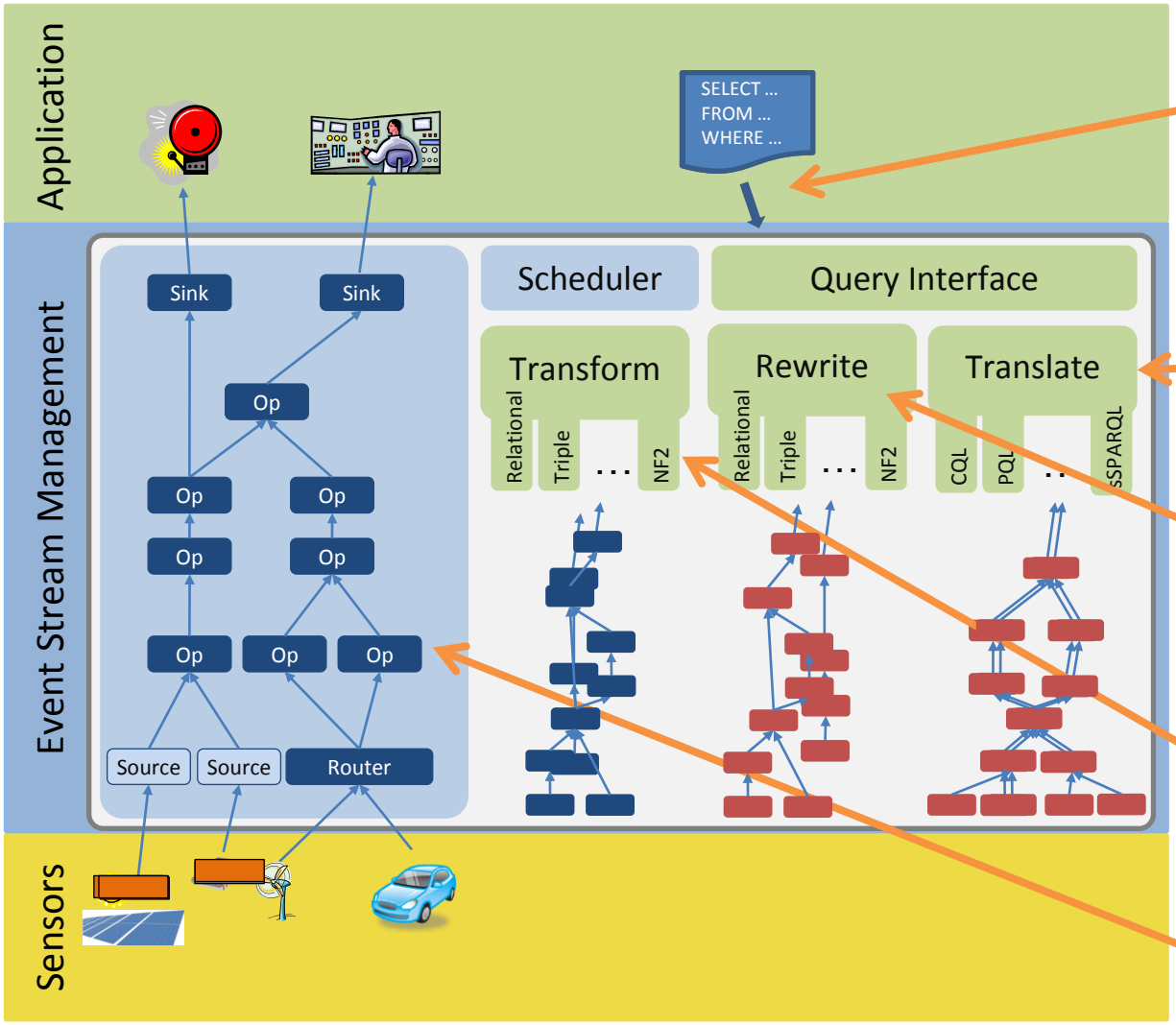
- Time intervals as stream model
  - Semantically defined and deterministic processing
  - System time independent
  - Robust against race conditions or bursts
- Built-in optimization techniques
  - Reduction of system load and latencies
- Framework architecture (OSGi)
  - Extensible for new requirements, operators, scheduling strategies, …
  - Adaptable even at runtime



Download and Information:
http://odysseus.informatik.uni-oldenburg.de

Apache 2.0 License

Odysseus DSMS



1. Application creates query
2. Translates the query into logical query plan
3. Optimizes the logical query plan
4. Transforms the logical into a physical query plan
5. Executes the query plan

Sensor Fusion Example with Odysseus

Operators:

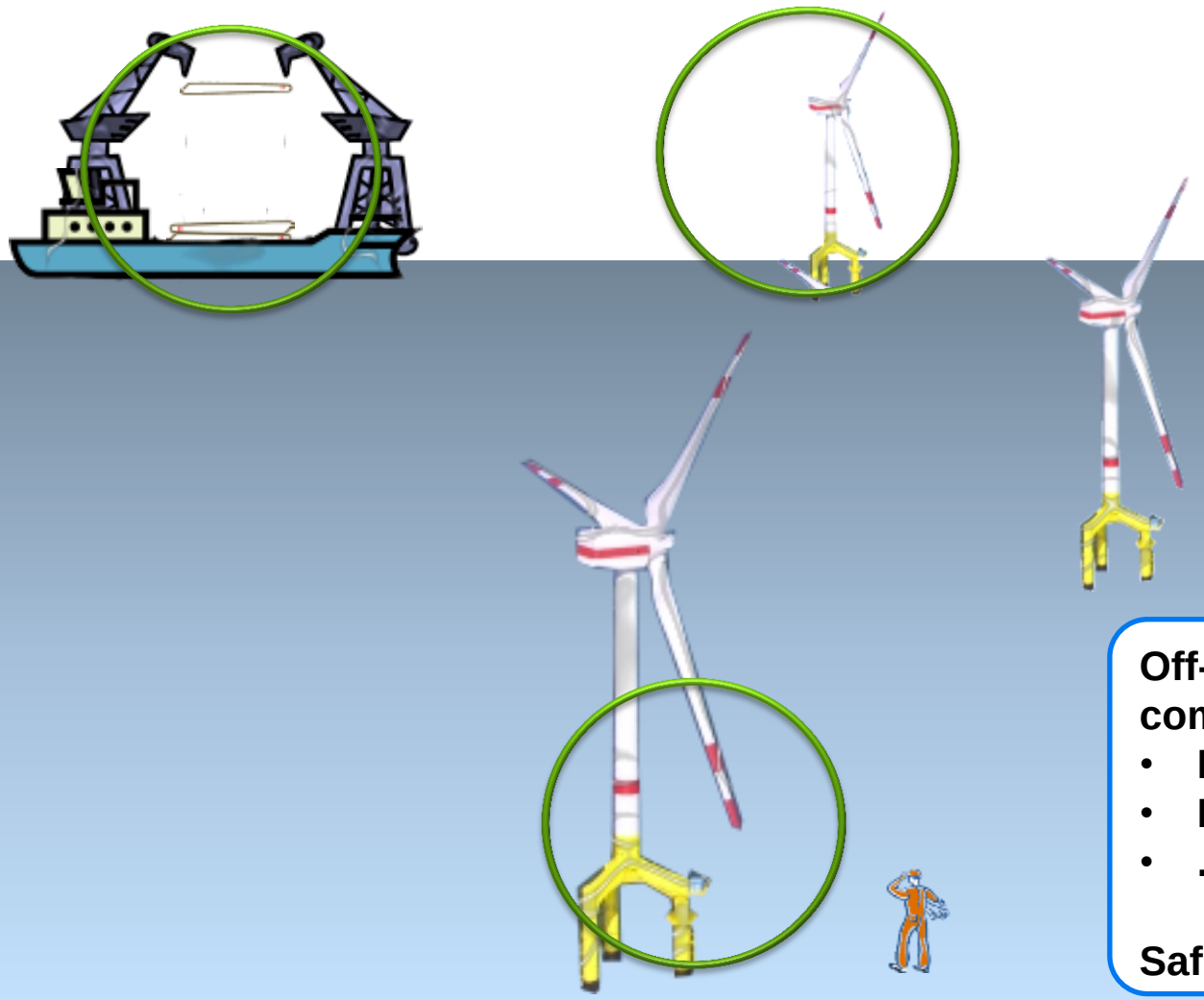
- Union Operator: Union and transmission of updated context models according to their timestamps
- Store Operator: Storing of new occupancy grid into the Context Store
- Merge Op…
- …ple …th their time …of validity
- 2 Laserscanner with up to 1080 beams (@0.25°), resulting in **54,000 points per second per sensors**

Example 2: Safe Offshore Operations (SOOP)

CARL VON OSSIETZKY universität OLDENBURG

OFFIS

& Project SOOP Sichere Offshore-Operationen

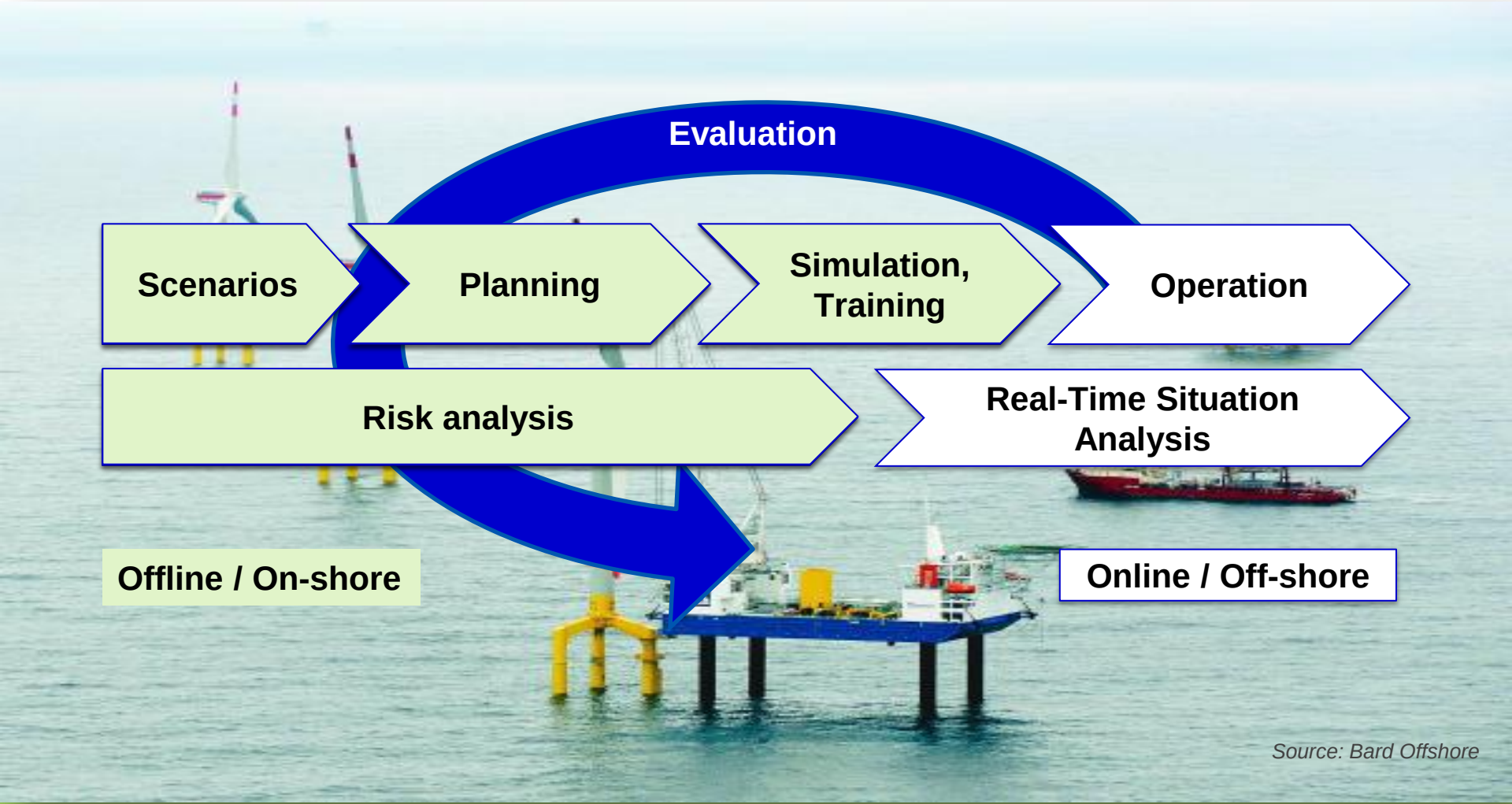


Off-shore operations are complex & risky!

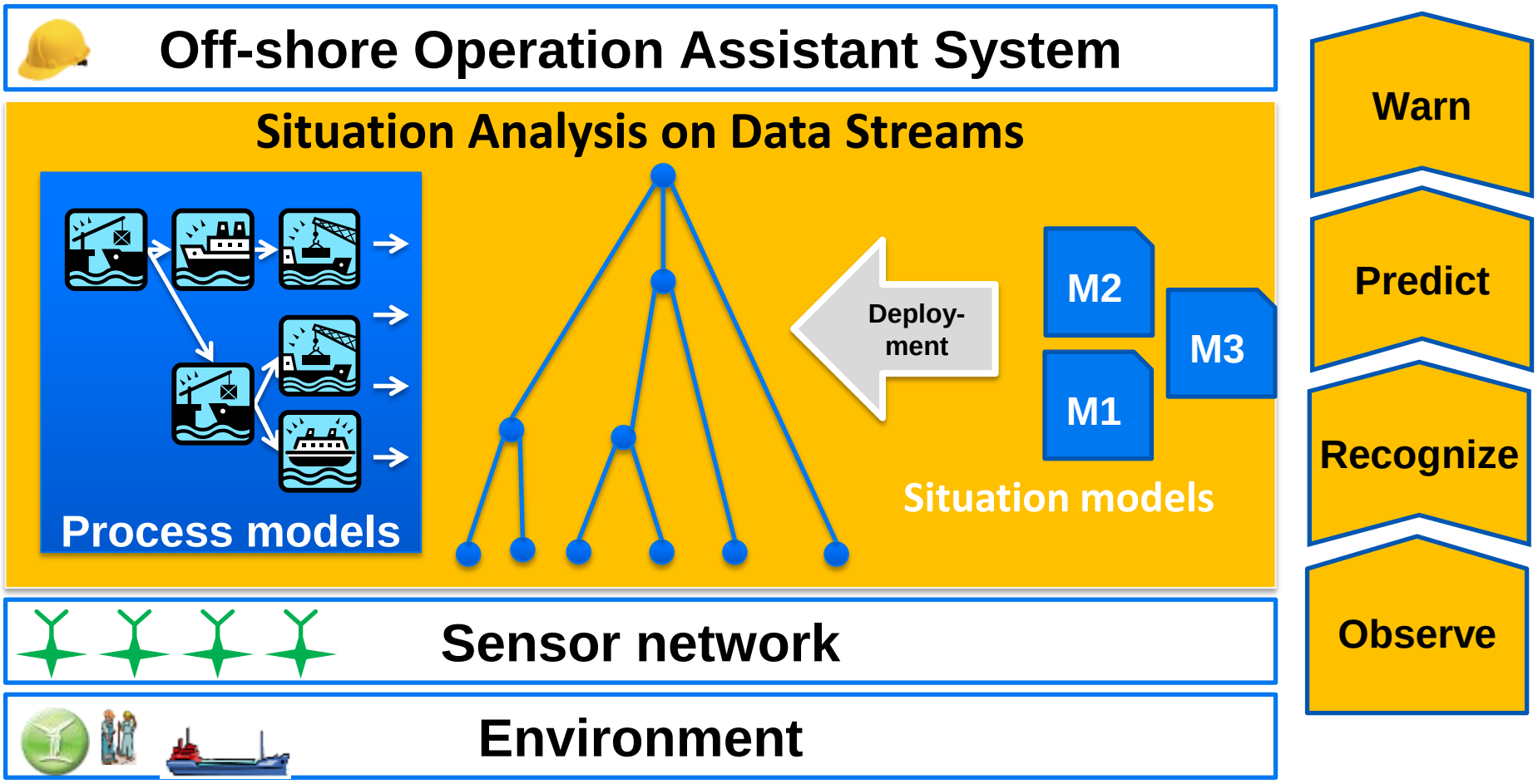
- **Harsh environment**
- **Long supply chains**
- **...**

Safety → Bankability

SOOP Overview



Real-Time Situation Analysis

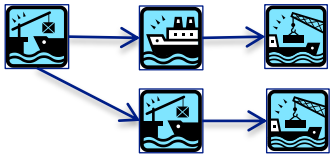


How to set up the Situation Analysis

What we got …

From Planning Phase:

Operation Model (Actors, Activities, Resources, Processes)



From Risk Analysis:

Hazard List

Context Model

A ship

(with some built-in sensors)

A bunch of sensors

What we need …

Off-shore Operation Assistant System

Visualize **Warn**

Data stream management

Dynamic Context Model

Stream Processing Plans

- For each sensor configuration
- For each mission

Warn

Predict

Recognize

Observe

sor gateway(s

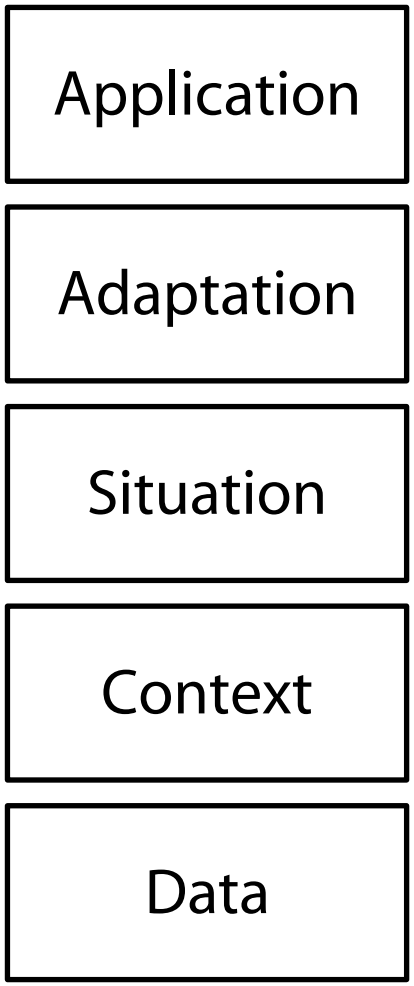
How to set up the Situation Analysis

How do we get there …

- Use generic system architecture from context-aware applications!
  - Data: typed data from sensors
  - Context: relevant observable parameters from the enrivonment
    - E.g.: crew.position, sea.level, crane.status, …
    - In SOOP: defined by the context model
  - Situation: relevant combination of context parameters
  - Adaptation/Application: warn crew

In SOOP:

- Non-critical: current activity (from process model), general status of ressources
- Critical:
  - Hazards („crew member overboard“)
  - Trends towards hazards („sea rising“)

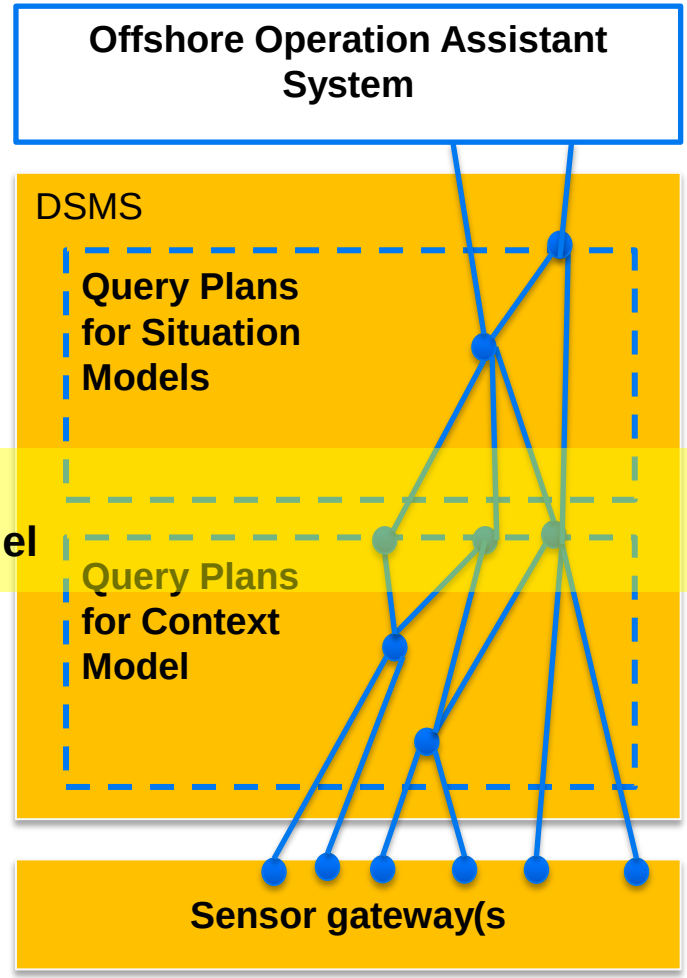


Situation models

- Situation definition:
  - Semi-formal description of the situation
  - Developed by domain experts
  - Uses system model (context variables)
- Situation model:
  - Executable query plan(s) based on context variables

higher level „view“ on sensor data / context model

- Context definition:
  - Semi-formal description of parameters in the system model
- Context model:
  - Set of executable query plans to determine context parameters based on sensor data



Conclusion

- Dynamic context models are needed in many (mobile) applications
- 2 example applications
  - sensor fusion for autonomous vehicles
  - environmental monitoring for safe offshore operations
- Challenges:
  - management of high update rates
  - sensor data (and thus context) quality, mainly
    - inaccuracy
    - staleness
    - existential uncertainty
  - modeling and maintanance of context models and situations
  - communication to mobile applications

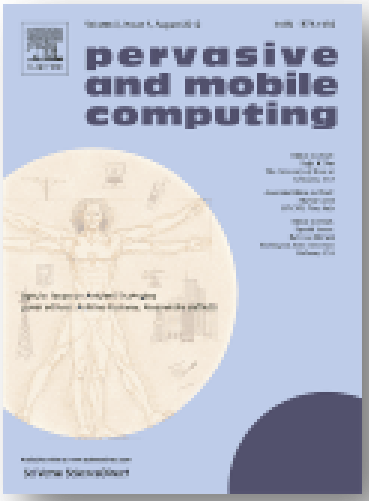
- Data stream management can be one tool to work on these challenges

http://nexus.informatik.uni-stuttgart.de/COMOREA

CoMoRea 2013: 10th IEEE Workshop on Context Modeling and Reasoning

San Diego, California, March 18-22, 2013

http://www.journals.elsevier.com/pervasive-and-mobile-computing



Special Issue on Information Management in Mobile Applications

Submission deadline: November 30th, 2012
Expected Publication of Special Issue: 2013

The **Pervasive and Mobile Computing Journal (PMC)** is a professional, peer-reviewed journal that publishes high-quality scientific articles (both theory and practice) covering all aspects of pervasive computing and communications.
Topics include, but not limited to:
… (a selection) …

- **Mobile computing systems and services**
- **Mobile grid and peer-to-peer computing**
- **Context-aware computing and location-based services and applications**
- **Service creation, discovery, management, and delivery mechanisms**
- **Middleware and agent technologies**

Impact Factor: 1.250
Imprint: ELSEVIER
ISSN: 1574-1192

Thank‘s for the fish!

… also to Christian Kuka (OFFIS), Marco Grawunder (Uni Oldenburg),
Jadwiga Indulska (University of Queensland)
and Nils Koppaetzky (Uni Oldenburg)