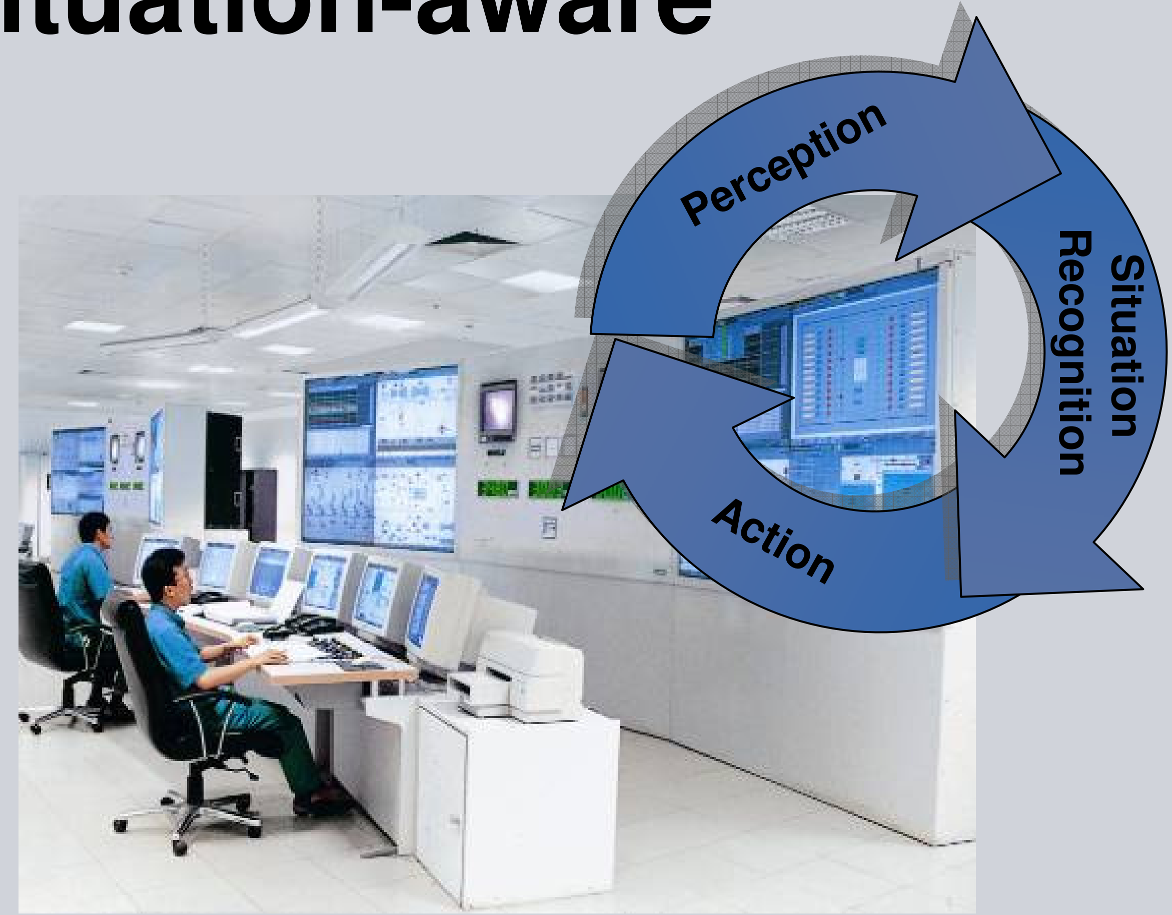
Reasoning in Situation-aware Applications

Steffen Lamparter

Siemens AG
Corporate Technology
Autonomous Systems Group



Application Areas of Situation-aware systems

Monitoring of complex technical systems for proactive maintenance and diagnostics

Examples:

- Maintenance of railroad infrastructures
- RFID-based incident management in production and logistics

Monitoring of human behavior to recognize critical situations

Examples:

- Surveillance
- Ambient Assisted Living

Fossil Power Generation

Renewable Power Generation

Power Transmission

Power Distribution

Water Technologies

Lighting (OSRAM)

Mobility

Solutions for Industry

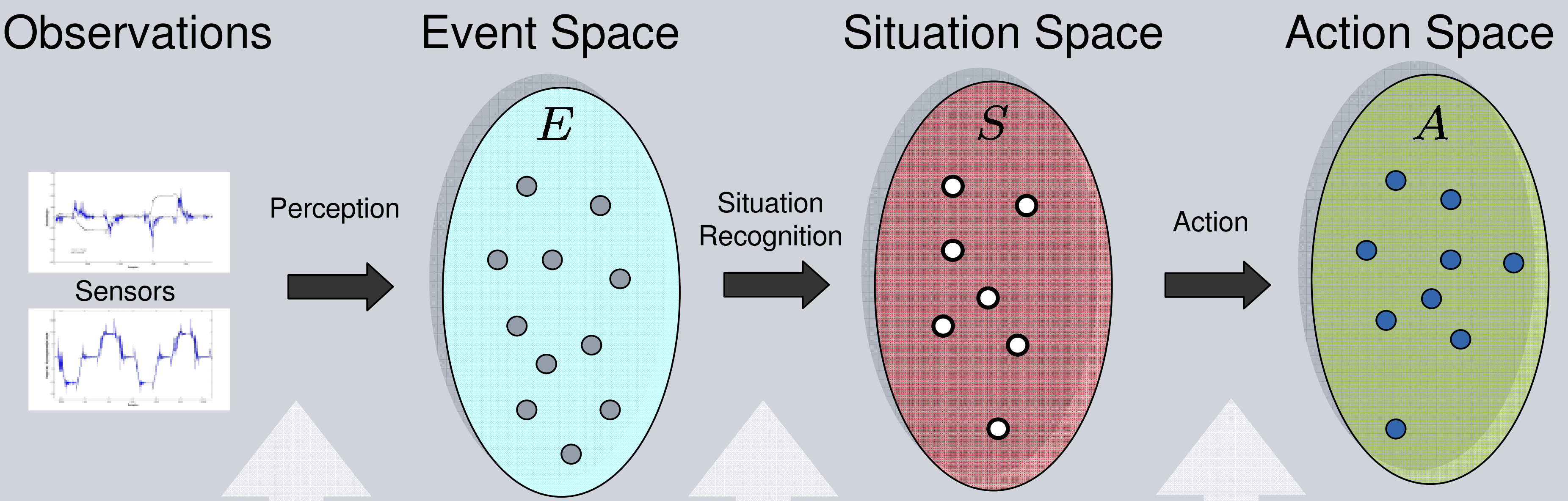
Healthcare

IT Solutions and Services

Building Technologies

Situation-aware Applications: Architectural Blueprint

SIEMENS



Observations

Event Space

Situation Space

Action Space

Sensors

Perception

E

Situation Recognition

S

Action

A

Which events are observed by a sensor?

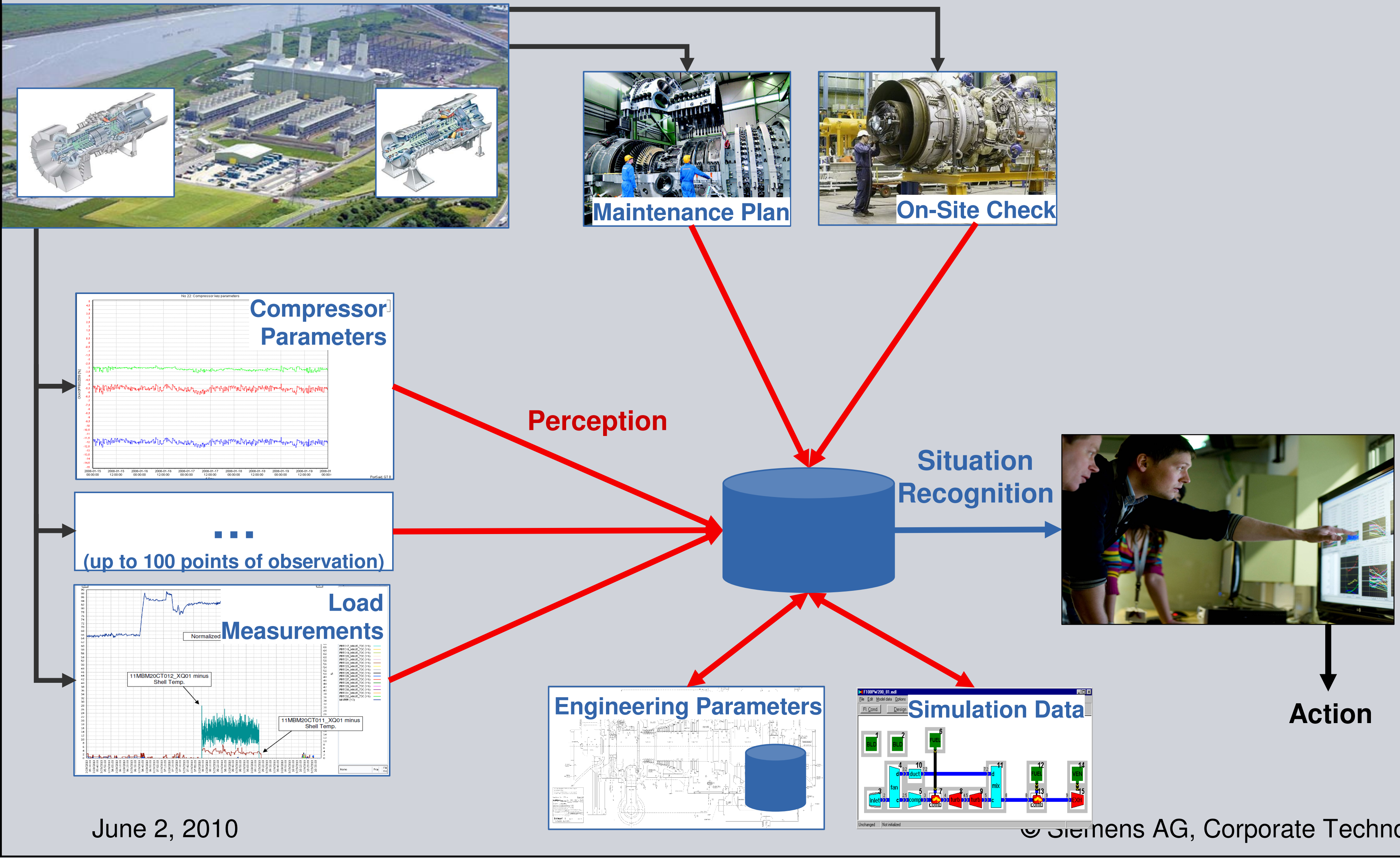
Which situations can be identified by a set of observed events?

Which actions have to be executed in a certain situation?

Reasoning based on Semantic Models

Example: Proactive Maintenance for Complex Industrial Processes

SIEMENS



Maintenance Plan

On-Site Check

Compressor Parameters

...

(up to 100 points of observation)

Load Measurements

Perception

Situation Recognition

Engineering Parameters

Simulation Data

Action

Normalized
11MBM20CT012_XQ01 minus Shell Temp.
11MBM20CT011_XQ01 minus Shell Temp.

Situation-aware Applications: Challenges for Semantic Technologies

SIEMENS

Situation Recognition Algorithms for Event Streams

- Efficient situation understanding for stream data necessary
- Situations depend on temporal as well as causal relations between events

Distributed Situation Recognition

- Events come from various sources and locations
- Processing and communication in real time

Intelligent Sensors with Situation Recognition “on board”

- Situation understanding algorithms are executed embedded on device/sensor
- Tractability on resource restricted hardware

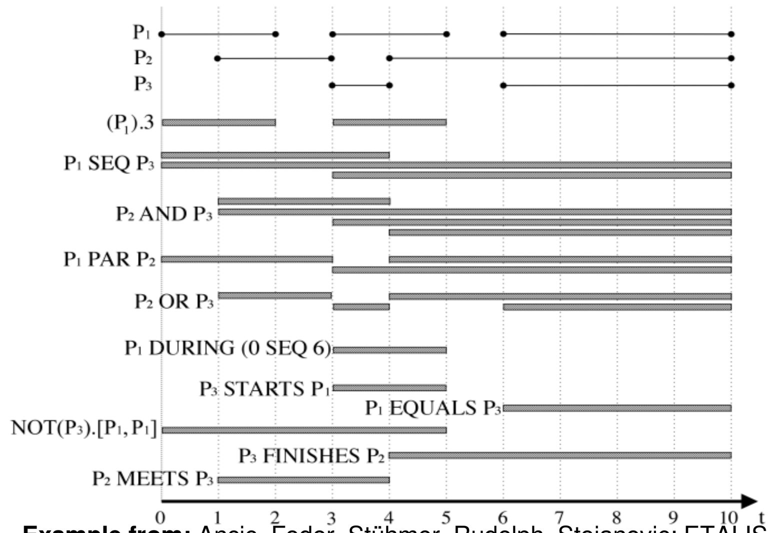
Towards “on-board diagnostics”

Situation Specification: Event Correlation Language

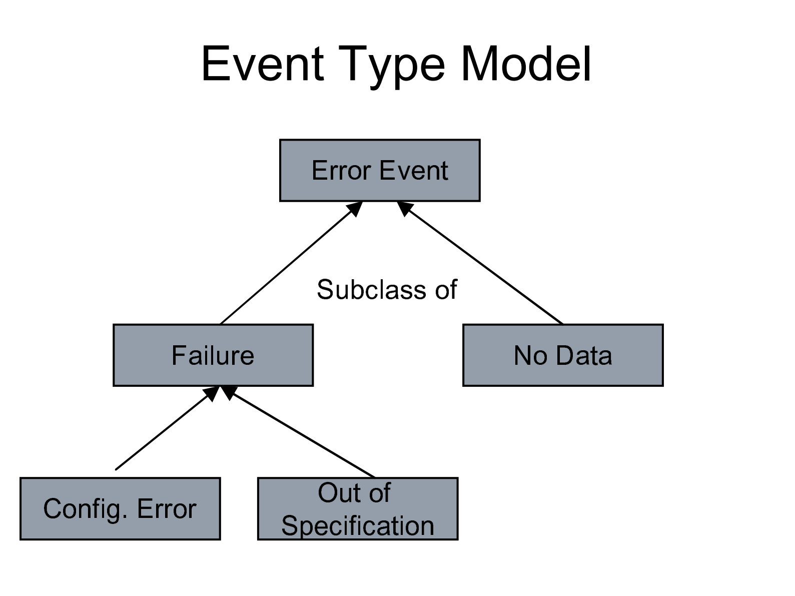
SIEMENS

Situation = (temporal, causal) correlation between events

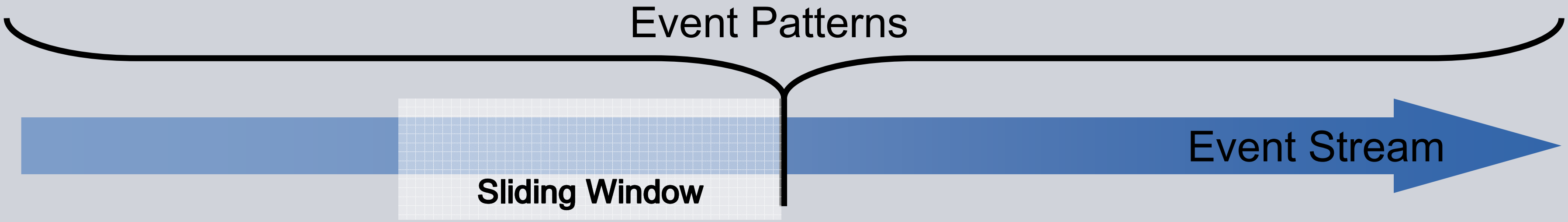
Situations



Example from: Ancic, Fodor, Stühmer, Rudolph, Stojanovic: ETALIS: A Rule-based System for Complex Event Processing, 2010



Events

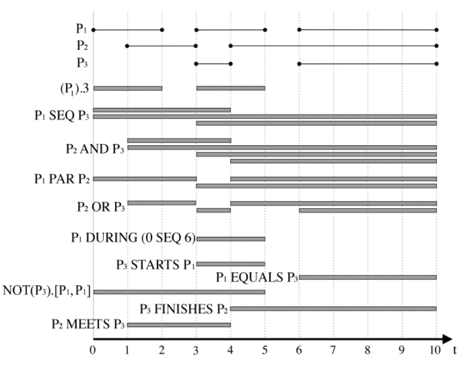


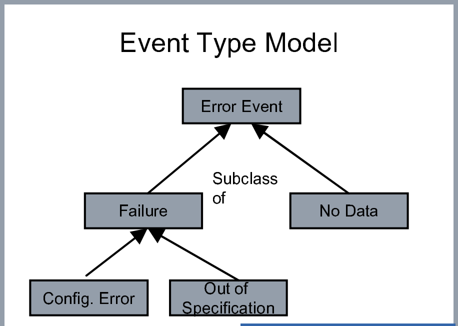
SIEMENS

Rule-based Event Processing

Current Approach: Complex Event Processing using Rule Engines

Event Correlation Language





OWL 2 RL

Situation

EventPattern → *Situation*

Rule Engine

Prolog, Clips, Drools

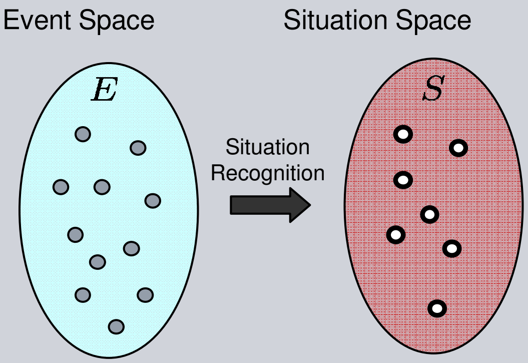
Various approaches in literature for

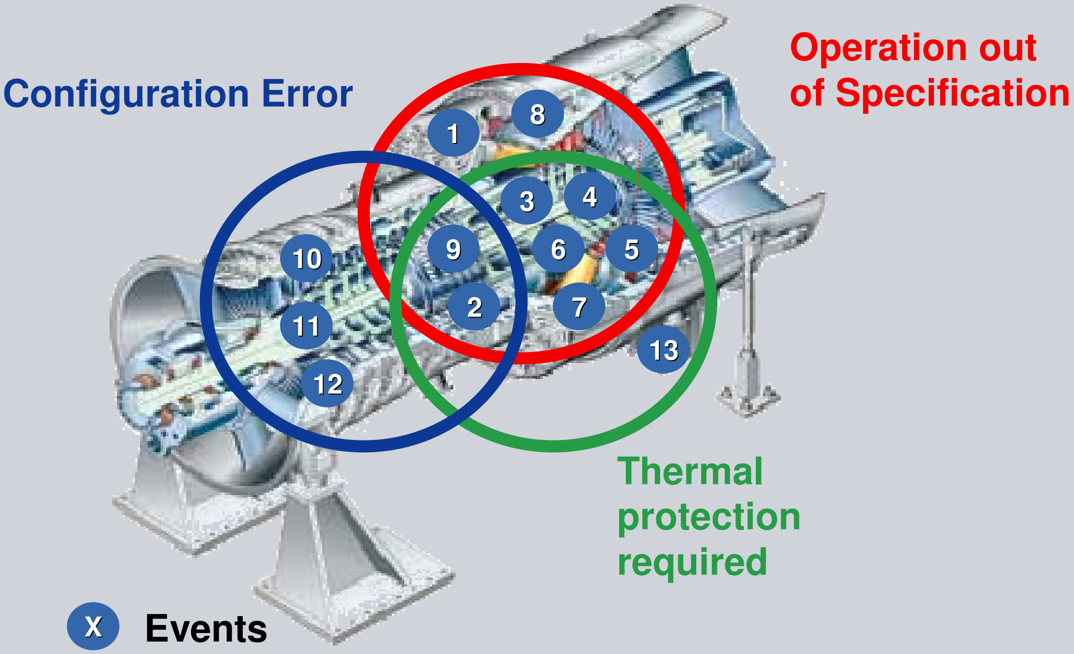
- semantics of temporal language constructs, e.g. RETE extensions
- Consumption policies to implement sliding window over events

Reasoning for Situation Understanding

Deductive Reasoning for Situation Understanding

- Requires full information about relation between events and situations
  - *EventPattern* → *Situation*





Reasoning for Situation Understanding

Problem: Incomplete Information

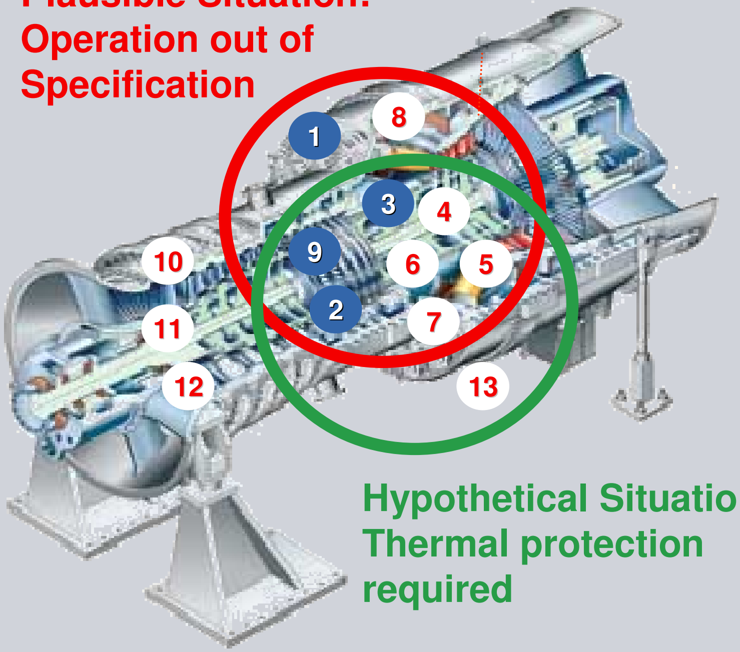
- Not applicable to predictive maintenance
- Reasons for the particular situations are often unknown

Current Approach

- Introduce classification of situations: certain, plausible and hypothetical situations
- Rank plausible and hypothetical situations according to heuristics as well as explicit preferences
- Support for simple event patterns only

Some challenges

- Abduction reasoning for situation understanding with incomplete information
- Probabilistic reasoning for handling uncertain information
- Inductive reasoning enables iterative improvements of (deductive) situation understanding



Plausible Situation: Operation out of Specification

Hypothetical Situation: Thermal protection required

X Observed event

X Unknown

Situation-aware Applications: Challenges for Semantic Technologies

Situation Understanding Algorithms for Event Streams

- Efficient situation understanding for stream data necessary
- Situations depend on temporal as well as causal relations between events

Distributed Situation Understanding

- Events come from various sources and locations
- Processing and communication in real time

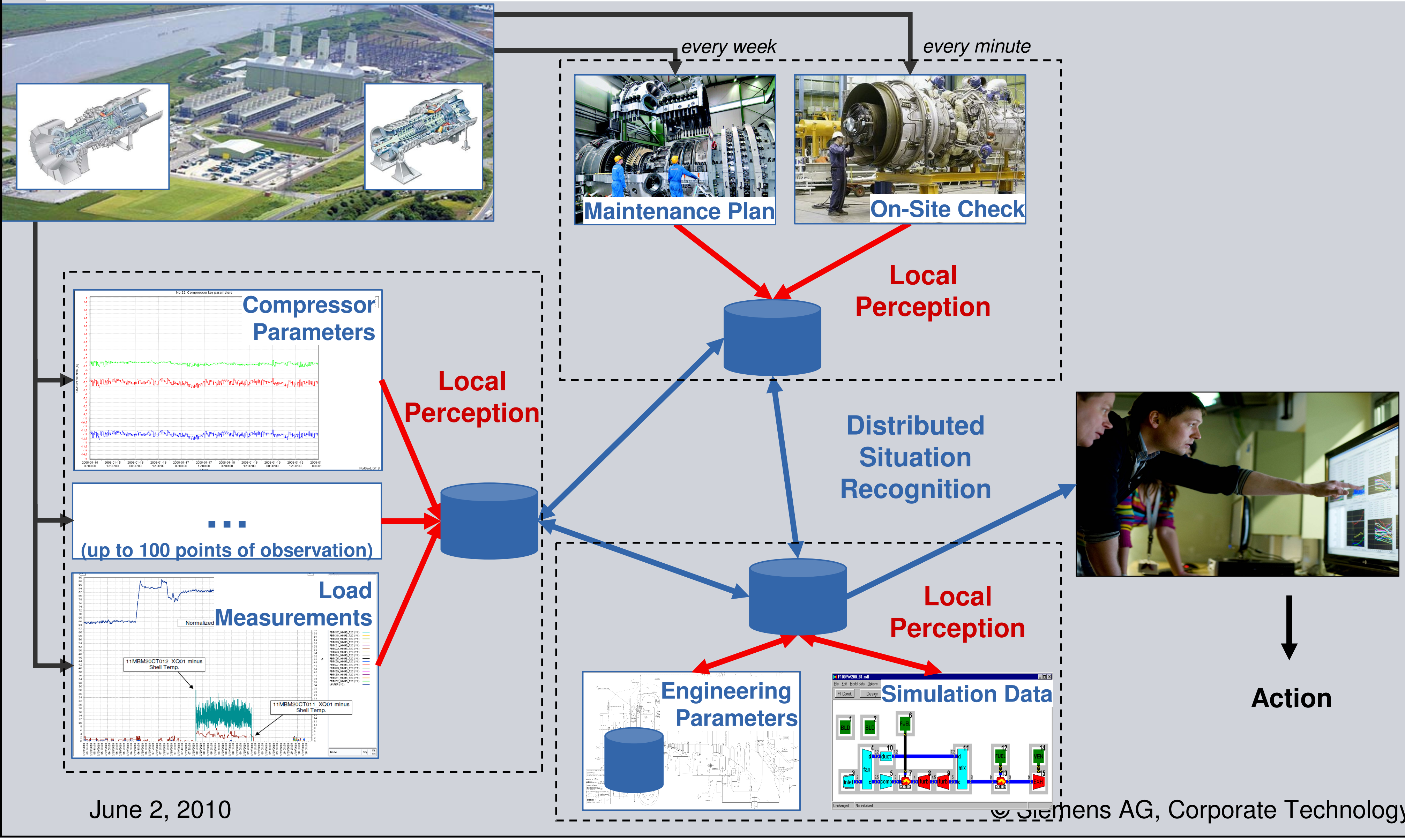
Intelligent Sensors with Situation Understanding “on board”

- Situation understanding algorithms are executed embedded on device/sensor
- Tractability on resource restricted hardware

Towards “on-board diagnostics”

Example: Proactive Maintenance for Complex Industrial Processes

SIEMENS



every week

every minute

Maintenance Plan

On-Site Check

Local Perception

Compressor Parameters

...

(up to 100 points of observation)

Load Measurements

Local Perception

Distributed Situation Recognition

Local Perception

Engineering Parameters

Simulation Data

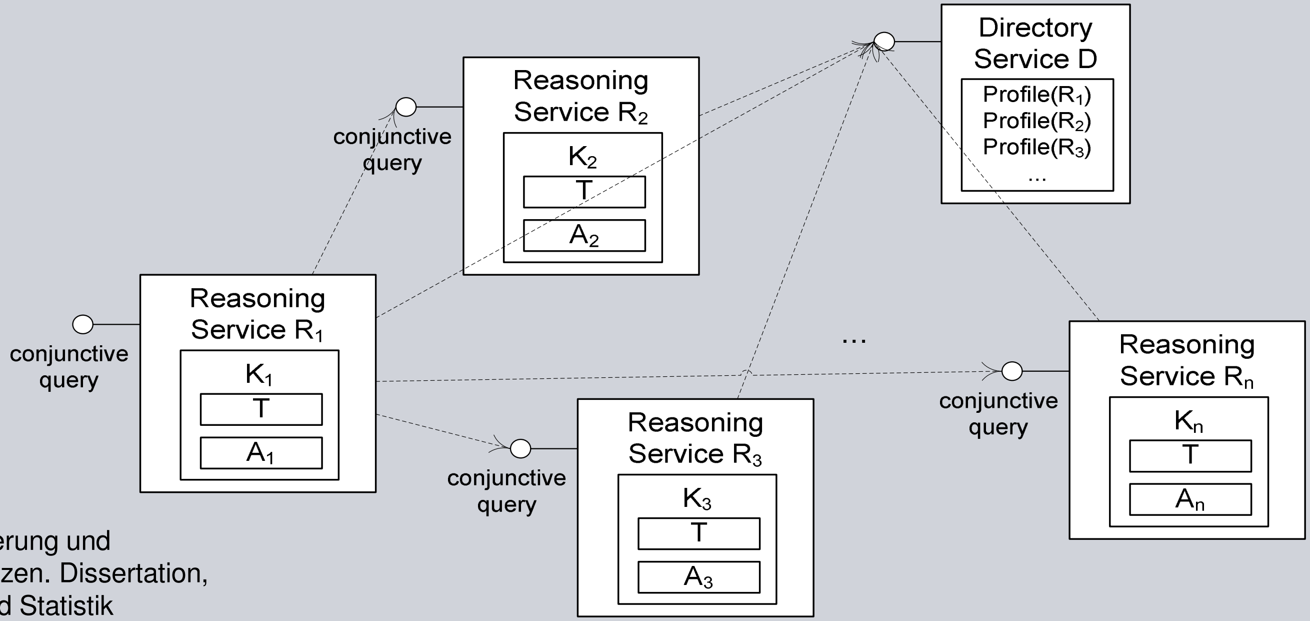
Action

11MBM20CT012_XQ01 minus Shell Temp.
11MBM20CT011_XQ01 minus Shell Temp.

Distributed Situation Recognition

Reasoning over Distributed Knowledge Bases

- Avoid collection of all sensor data in one central knowledge base
- Instead of collecting data distribute queries to remote reasoners and integrate results



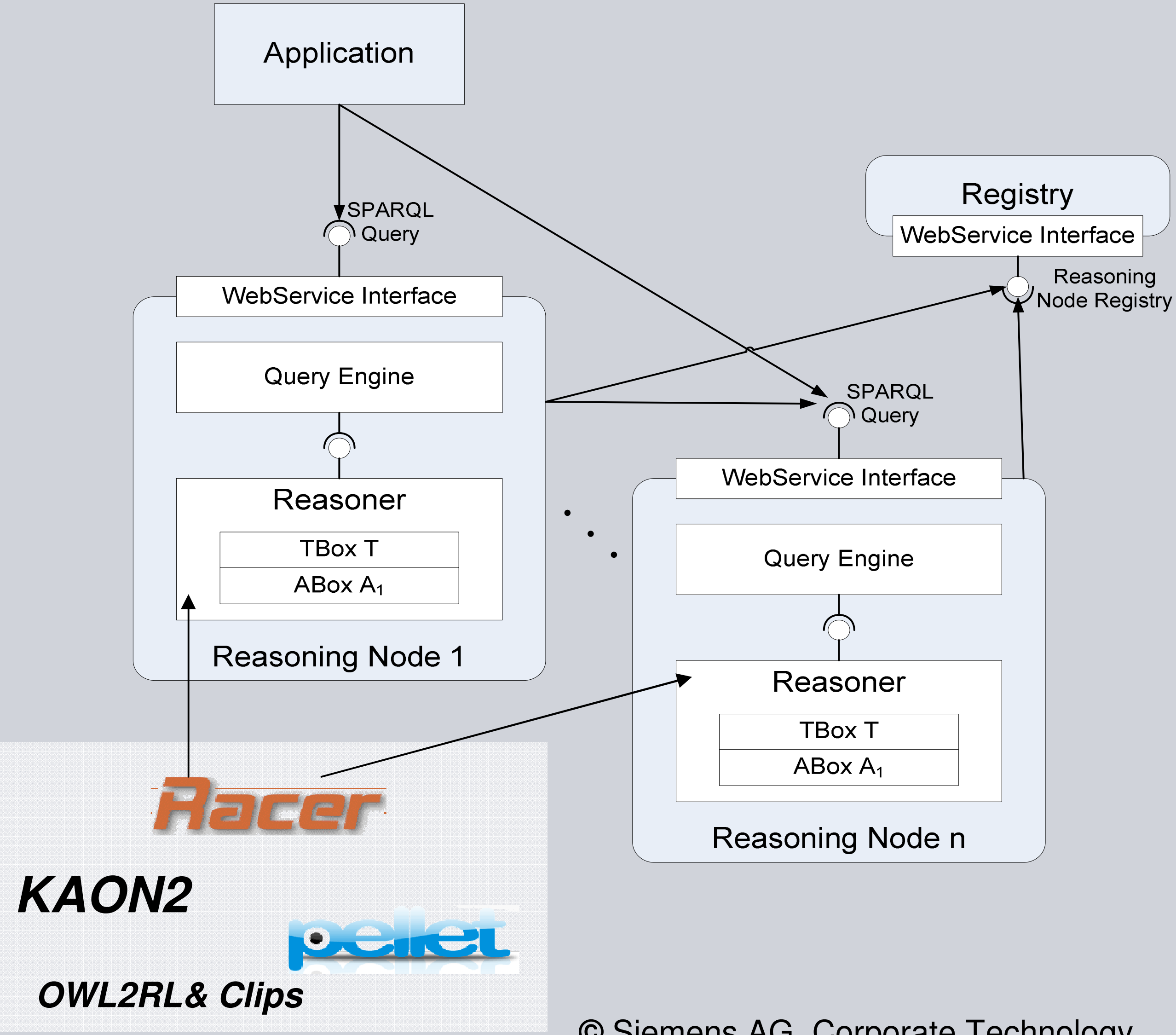
Source: Fuchs, Florian (2008): Semantische Modellierung und Reasoning für Kontextinformationen in Infrastrukturnetzen. Dissertation, LMU München: Fakultät für Mathematik, Informatik und Statistik

SIEMENS

Distributed Situation Recognition: Implementation

Architecture:

- Java Axis2 WebServices
- ReasoningNode as black box
- Intelligence in Query Engine: generates sub queries
- Deployment in servlet engine (e.g. Tomcat)
- Different setups possible (e.g. hierarchical, mesh-networks, etc.)



Restrictions w.r.t. OWL DL:

- Shared roles must not be universally quantified (forall r.C' not allowed)
- Shared roles must not be restricted in cardinality (≤n r not allowed)

Situation-aware Applications: Challenges for Semantic Technologies

Situation Understanding Algorithms for Event Streams

- Efficient situation understanding for stream data necessary
- Situations depend on temporal as well as causal relations between events

Distributed Situation Understanding

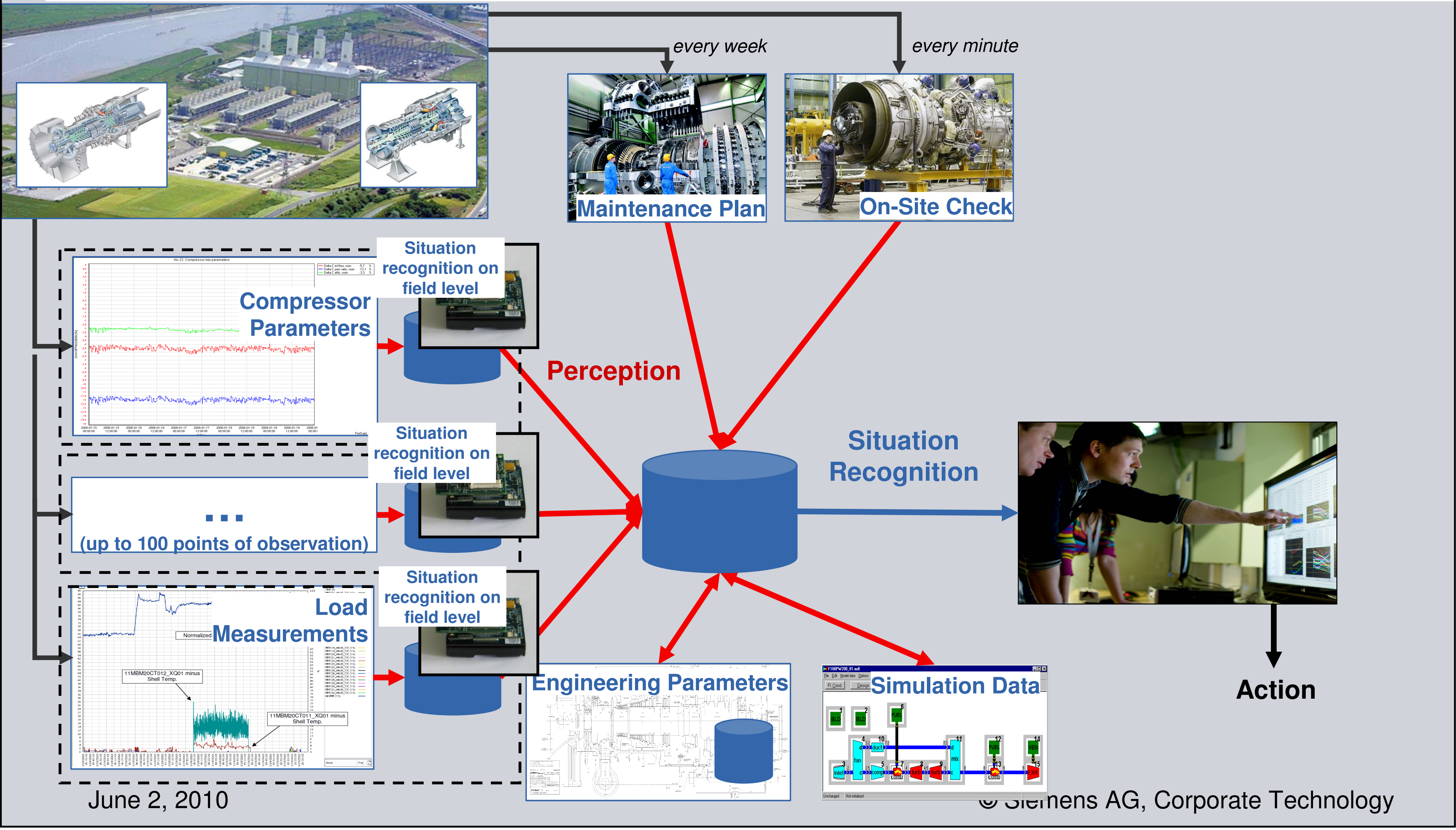
- Events come from various sources and locations
- Processing and communication in real time

Intelligent Sensors with Situation Understanding “on board”

- Situation understanding algorithms are executed embedded on device/sensor
- Tractability on resource restricted hardware

Towards “on-board diagnostics”

Example: Proactive Maintenance for Complex Industrial Processes



every week

every minute

Maintenance Plan

On-Site Check

Situation recognition on field level

Compressor Parameters

Situation recognition on field level

...

(up to 100 points of observation)

Situation recognition on field level

Load Measurements

Perception

Situation Recognition

Action

Engineering Parameters

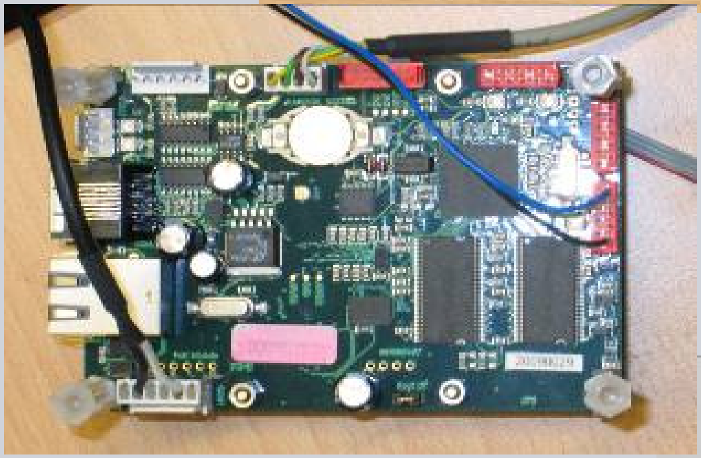
Simulation Data

Normalized
11MBM20CT012_XQ01 minus Shell Temp.
11MBM20CT011_XQ01 minus Shell Temp.
inlet
fan
duct
comp
comb
mix
FUEL
EXH

Possible Hardware

Gumstix Verdex Pro XM4 COM

- Processor: Marvell™ PXA270,
- Speed:ARM XScale 400MHz
- Memory: 64MB RAM, 16MB Flash
- Linux 2.6



AUG Development Board

- ATMEL AT91SAM9261 (ARM9)
- Speed: 200MHz
- Memory: 64 MByte SDRAM, 256 MByte NAND Flash

xBow Imote2

- PXA271 processor at 13–416MHz
- 256kB SRAM, 32MB FLASH, 32MB SDRAM
- Integrated 802.15.4 Radio

Reasoners under investigation….

FaCT++ ⚡

- Programming language C++
- Stand-alone → I/O error caused by ARM
- OWL API → Alpha version for FaCT++ dosn‘t work

Pellet ⚡

- Programming language Java
  →no working Java environment
- GNU classpath – not fully compatible
- SUN Embedded for Java – no required lib

CEL ⚡

- Programming language Lisp
- Only Allegro CL → closed source, no ARM-version (cross-compiling necessary)

Racer ⚡

- Programming language Lisp
- closed source, no ARM-version (cross-compiling necessary)

Clips rule engine as OWL2RL reasoner

- Small, light, and rather fast
- Uses Rete algorithm, forward chaining
- Implemented in C, C++ wrapper available
- Available for various platforms (e.g. Windows, Linux, Unix)
- License: Open Source

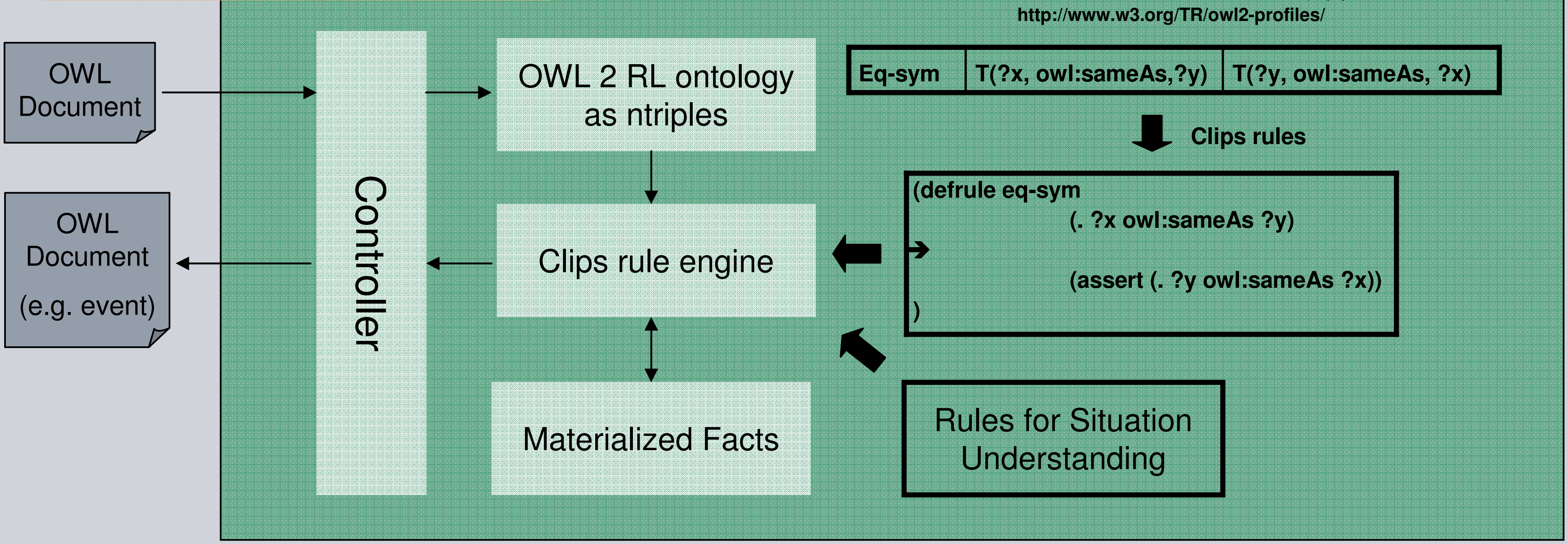
SIEMENS

Embedded situation recognition

- Simplified architecture of embedded reasoning on Gumstix hardware

OWL 2 RL/RDF rules axiomatization (approx. 80 rules)

http://www.w3.org/TR/owl2-profiles/


OWL Document
OWL Document (e.g. event)
Controller
OWL 2 RL ontology as ntriples
Clips rule engine
Materialized Facts
Clips rules
Rules for Situation Understanding

Eq-sym	T(?x, owl:sameAs,?y)	T(?y, owl:sameAs, ?x)

```
(defrule eq-sym
                    (. ?x owl:sameAs ?y)
→
                    (assert (. ?y owl:sameAs ?x))
)
```

Current work: improve approach w.r.t. memory usage

Conclusions

- Many applications of situation-aware applications in energy, industry and healthcare sector

- Situation-aware applications require
  - Identification of_relevant events based on multi-model sensor inputs
  - Recognition of complex situations
  - Triggering of appropriate actions

- Challenges for situation recognition
  - Efficient data stream processing with complex event patterns
  - Realization of distributed situation recognition algorithms
  - On-board diagnostics with situation recognition algorithm embedded on device

SIEMENS

Thank you for your attention!

Dr. Steffen Lamparter
Corporate Technology
Siemens AG

Otto-Hahn-Ring 6
81739 München

Phone: 089 - 636 40383
Fax: 089 - 636 41423

E-mail:
steffen.lamparter@siemens.com