A fault-tolerant info'structure for energy applications

Geert Deconinck
K.U.Leuven – ESAT – ELECTA

15 Nov. 2006

Overview

- introduction
- separating functionality & fault tolerance
- overlay network
- application example
- conclusion

Faculty of engineering (Faculteit Ingenieurswetenschappen)

L.Froyen *dean*

dept. head architecture
dept. head indust. chem.
dept. head mechanical eng.
dept. head civil eng.
dept. head material science
dept. head comp. science
M. Steyaert *dept. head* electrical eng.

Electrical engineering department (Elektrotechniek - ESAT)

M. Steyaert *dept. head*

P. Suetens PSI
R. Belmans ELECTA
J. Vandewalle SCD
W. Sansen MICAS
A. Vande Capelle TELEMIC
R. Mertens INSYS

ELECTA
electrical energy & computer architectures

- Ronnie Belmans, head
- administrative support
- technical support

J. Driesen
Power electronics
Power quality
Distributed generation

R. Belmans
Electrical drives
Electrical energy systems
Market

K. Van Reusel
Electroheat

G. Deconinck
Dependable system architectures
Industrial automation

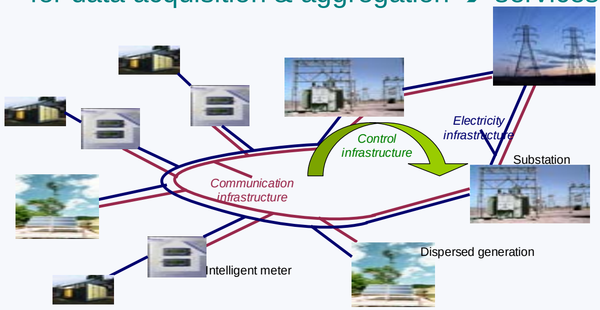
D. Van Dommelen
Generation, transport, distribution

ELECTA statistics (Nov. 2006)
www.esat.kuleuven.be/electa

- people
  - 4 professors
  - 2 post-docs
  - 34 researchers
    - 5+ nationalities
  - 8 techn./admin
- output for 2005 (2006)
  - 9 (8) PhDs
  - 23 (8) journal papers
  - 57 (60) conf papers

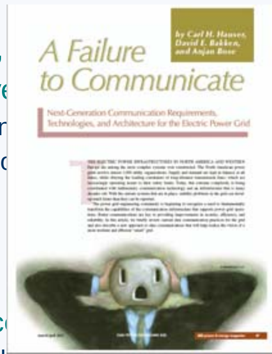
Scope:
distributed (renewable) energy apps

- decentralised & distributed control
  - for optimising power quality, losses, costs …
  - for data acquisition & aggregation ➔ services



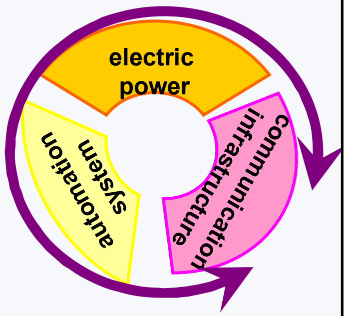
Info'structure in electricity applications

- information infrastructure
  - HW, SW, comm, applications,
  - no significant changes in 40 ye
    - more performant communicatior
    - still centralised, human in the loo
- current trends
  - off-the-shelf components
    - heterogeneous
  - more decentralised intelligenc
    - data vs. information vs. knowledge



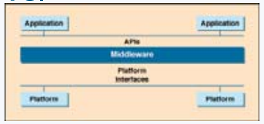
Embedded automation

- opportunities for autonomous, decentralised
  - more flexibility
- vulnerabilities
  - fault propagation
  - interdependencies



Requires methodology & architecture

- methodology
  - risk analysis
  - dependability specification
  - modelling & simulation
- architecture at middleware level
  - between OS and application
  - on heterogeneous platform
  - distributed, decentralised
  - for *dependable control* of power grid
    - survivability, self-healing, fault-tolerance, …
    - redundancy & diversity



Research questions…

- communication architecture
  - point-to-point, multicasting, broadcasting
  - time-triggered, event-triggered
  - push, pull
- interoperability
  - communication protocols
    - IEC61850, TASE.2 (ICCP), DNP3, IEC 60870, OPC…
  - vendor-independence, open, …

Research questions (cont.)

- dynamic aspects
  - different time scales
    - power electronics: sub-cycle
    - contingency: cycles .. seconds
    - economic optimisation: minutes .. hours
- dependability aspects
  - fault and failure models?
  - reliable communication
    - on top of unreliable infrastructure
  - different quality-of-service requirements

Overview

- introduction
- ➨ separating functionality & fault tolerance
- overlay network
- application example
- conclusion

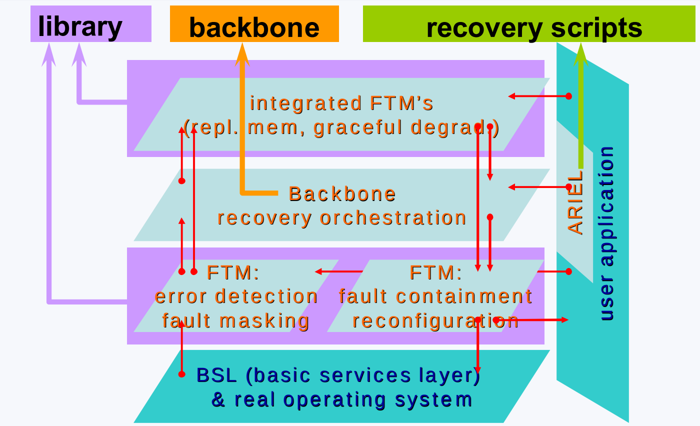
Summarizing dependability requirements of info'structure

- generic design
  - take advantage of same architecture
  - modular, composable
- fault tolerance
  - reconfiguration, self-testing & recovery abilities
  - separation of fault management from application
  - scalable fault management
    - using a hierarchical structure, i.e. an intra/inter-site hierarchy

TIRAN/DepAuDE framework approach

- middleware (implemented in C)
  - basic services layer (BSL)
    - communication: group, local, remote, tunnelled, …
    - task / node management: start, stop, inform, …
  - library of FTM (fault tolerance mechanisms)
    - detection, monitoring, masking, recovery, …
  - backbone (distributed application + database)
    - collects information (from appl, FTM, BSL)
    - orchestrates recovery (coordination of FTM)
- ARIEL language
  - for configuration of middleware
  - for expression of recovery scripts

Middleware architecture



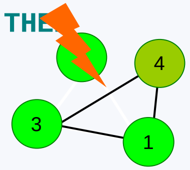
ARIEL as recovery language

- check if entity …
  - faulty?, isolated?, stopped?, running?, ...
  - ERRN(e), ERRT(e), PHASE(e)
  - expressions: ==, !=, ||, &&, !, …
- then actions …
  - stop! isolate! start! reboot! enable! send! ...
- metacharacters
  - task@: tasks for which guard is true
    - IF [FAULTY group1] THEN STOP@1
  - task~: tasks for which guard is false

Modifying recovery strategies

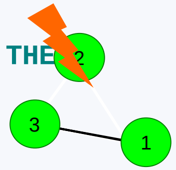
3-and-a-spare system:

```
IF [ -FAULTY group1 ] THE
   STOP task@1
   WARN task~1
   START task4
FI
```



graceful degradation:

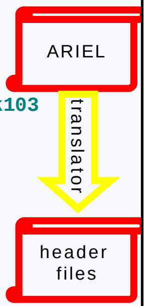
```
IF [ -FAULTY group1 ] THE
   STOP task@1
   WARN task~1
FI
```



ARIEL as configuration tool

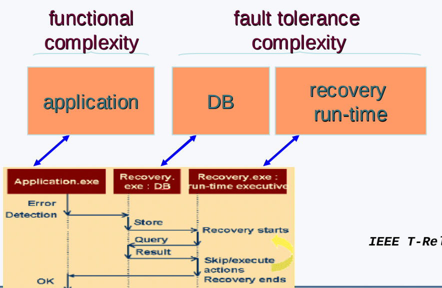
- configuration of basic tools
  - error detection, fault masking, isolation, recovery
  - templates for voting, recovery blocks, watchdogs, …
- e.g.: replication (3-modular redundancy)

```
REPLICATED task10 IS task101, task102, task103
  MULTICAST IS ATOMIC
  METHOD IS MODULAR REDUNDANCY
   VOTING ALGORITHM IS MAJORITY
   METRIC "int_cmp"
  END METHOD
  ON SUCCESS task20
  ON ERROR task30
END REPLICATED
```



Advantage: uncouple recovery from application

- modify FT strategy w/o changing application
  - 3-and-a-spare vs. degrading voting farm vs. …
- complexity ⇓, maintainability ⇑
- allows local & distributed recovery strategies



IEEE T-Reliability 51(2):158-165

Assumptions

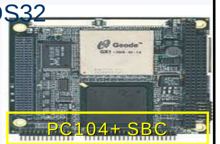
- fault and failure assumptions
  - single physical faults
  - fail-silent failures
  - fault containment region: task or node
    - depending on HW & RTOS
- synchronous system model
  - known & bounded
    - processing delay
    - communication delay
    - clock differences / clock drift
  - corresponds to instantiation on dedicated system
    - no dynamic task creation

Assumption coverage

- assumption coverage < 1
  - depending on instantiation on specific HW/RTOS
- optional mechanisms increase coverage
  - crash failure semantics
  - group-based communication, atomic multicast, …
  - e.g. perfect communication, at OSI level 5
    - i.e.
      - no lost messages, no duplicates, keeping message order
    - but
      - CRC + level 2 retransmission (Ethernet)
      - BSL: ACK-management

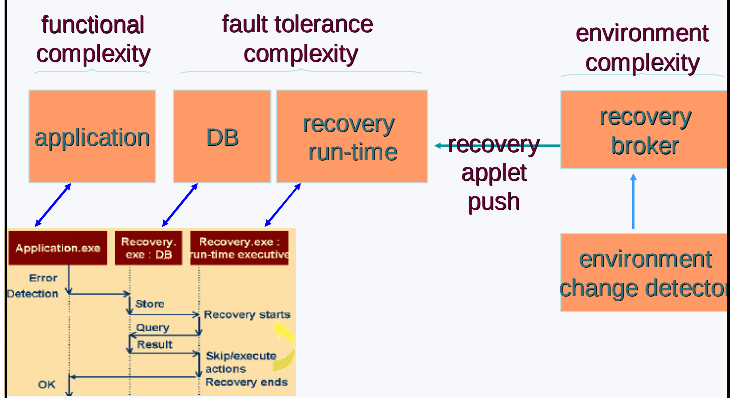
Framework implementation

- on embedded hardware / TCP/IP stack
  - RT network with SBC & PC
    - RTOS: QNX, WinCE, VxWorks, RMOS32
    - GPP OS: Linux, WinNT



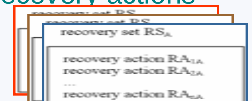
- applications
  - DepAuDE applications
    - heterogeneous substation control system
    - airfield lighting automation system
  - info'structure for distr. PQ meas. & control
    - dynamic topology: remove/reintegrate nodes
    - robust communication: protocols & inforedund.

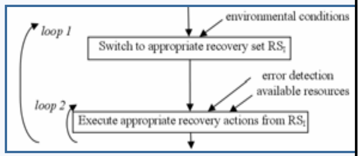
Extension to dynamic environment



Recovery sets & actions

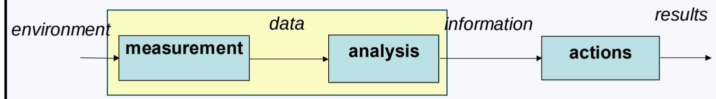
- recovery sets
  - recovery actions

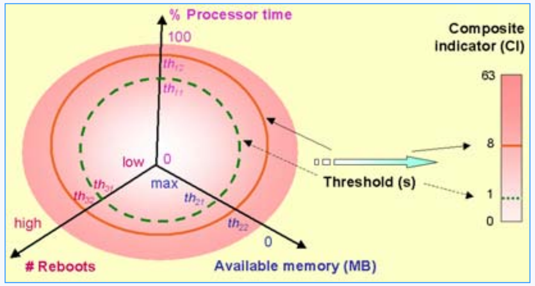




- based on resource / environment monitoring
  - selecting recovery actions
    - o resource awareness
  - switching recovery sets/strategies
    - o environment awareness
    - o adaptation of QoS to environment conditions
      - graceful degradation

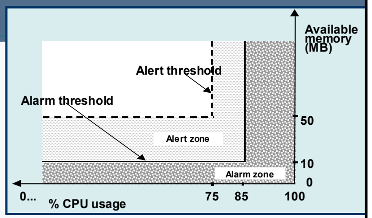
Measurement-based adaptation





CI implementation

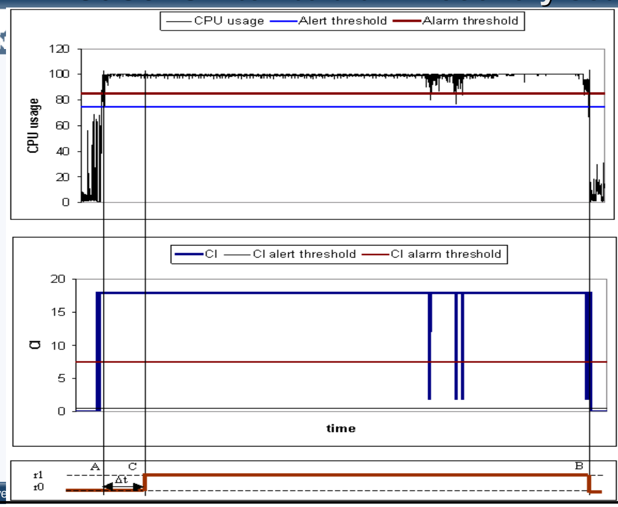
- parameters sampled
  - CPU usage
  - available memory
  - # reboots (up time)
- thresholds from literature



- OS-dependent
  - Win NT/2000/XP, QNX Neutrino RTOS, GNU/Linux
- example CI generation
  - c=3 counters, t=2 thresholds
    - o CI=0 - normal state
    - o CI>0 & CI <=7 (<=2^c-1) - transient alert state
    - o CI>7 (>2^c-1) - transient alarm state
  - ADC -Alarm Duration Counter if > ΔT triggers
    - alarm state» alarm recovery set (switch of recovery set)

Alarm threshold passed			Alert threshold passed		
Memory	CPU	reboot	Memory	CPU	Reboot
1	0	0	1	1	0

Case: switch to alarm recovery set



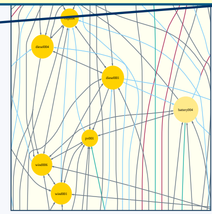
Overview

- introduction
- separating functionality & fault tolerance
- ➡ overlay network
- application example
- conclusion

Energy info'structure requirements

- energy applications: unbounded systems
  - n
  - s
- set
  - groups similar entities
  - XML description for entities
    - o attribute based addressing
  - requires resource discovery
  - adapts over time

```
<generator>
        <windturbine/>
        <ratedPowerkW>+10</ratedPowerkW>
        <generationPercent>*</generationPercent>
</generator>
```



Semantic Overlay Networks

- agora: a semantic overlay network
  - small-world overlay network
  - self-organising
  - ocality

Small-world networks

- small diameter (best of random graphs)

Group locality	**Time locality**
nodes work in groups	nodes request same resource frequently
⇩ cluster by functionality	⇩ pre-link


```
<entityDescription>
  <description>
    <deviceSegment>Arenberg-ESAT</deviceSegment>
    <electricalDeviceType>
      <generator>
```

```
<description>
  &lt;segment&gt;Arenberg-ESAT&lt;/segment&gt;
  <electricalDeviceType>
    <transformer/>
  </electricalDeviceType>
</description>
```

0.6

```
<description>
  &lt;segment&gt;Arenberg-ESAT&lt;/segment&gt;
  <electricalDeviceType>
    <generator>
      <photovoltaic/>
    </generator>
  </electricalDeviceType>
  <ElectricalDevice>
    <powerInW>100</powerInW>
  </ElectricalDevice>
</description>
```

- semantic di
  - metric refle

δ(u,v) = 1 - siz

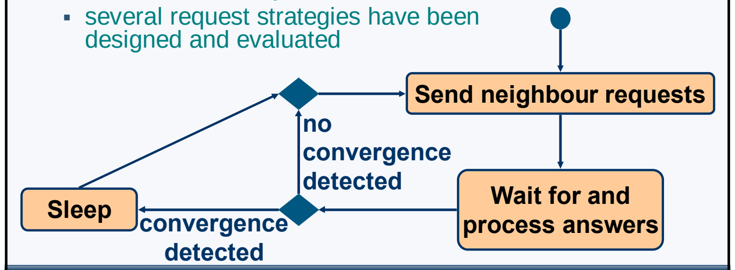
```
    <description>
      <deviceSegment>Arenberg-ESAT</deviceSegment>
      <electricalDeviceType>
        <transformer/>
      </electricalDeviceType>
    </description>
  </interests>
</entityDescription
```

Links from Agora nodes

- companion links
  - to the semantically closest nodes
  - provide group locality support
- pupil links
  - to nodes semantically closest to expressions of interest
  - provide time locality support
- far links
  - link to the semantically furthest node
  - all nodes have 50% probability for far link
  - prevent network partitioning and reduces diameter
  - effectuate small world topology

Node convergence

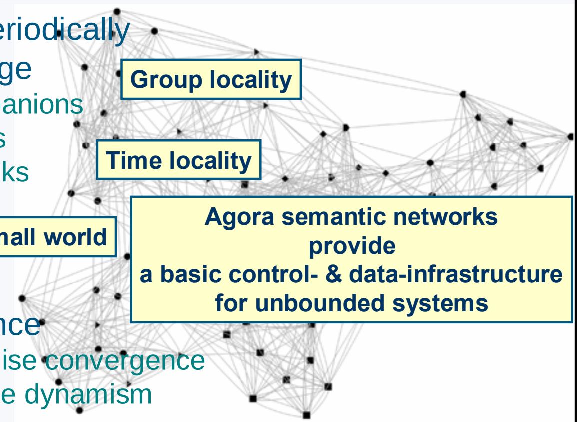
- nodes periodically converge
  - = send requests for better neighbours
  - handle dynamic environment
- requests are forwarded in the overlay
  - to the semantically closest area
  - several request strategies have been designed and evaluated



To summarise

nodes periodically

- converge
  - companions
  - pupils
  - far links
- announce
  - optimise convergence
  - handle dynamism



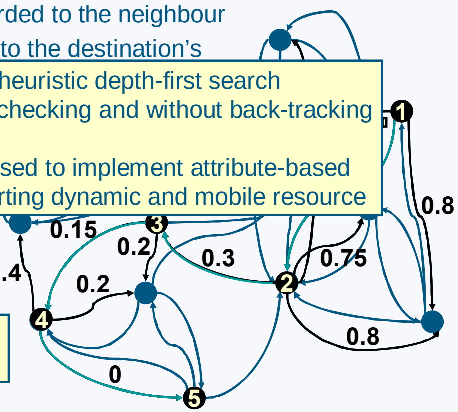
Agora semantic networks provide a basic control- & data-infrastructure for unbounded systems

Semantic routing

messages are forwarded to the neighbour semantically closest to the destination's descriptio

= heuristic depth-first search with cycle-checking and without back-tracking

can be used to implement attribute-based RD supporting dynamic and mobile resource



Semantic distance used as heuristic

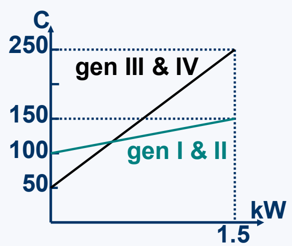
Overview

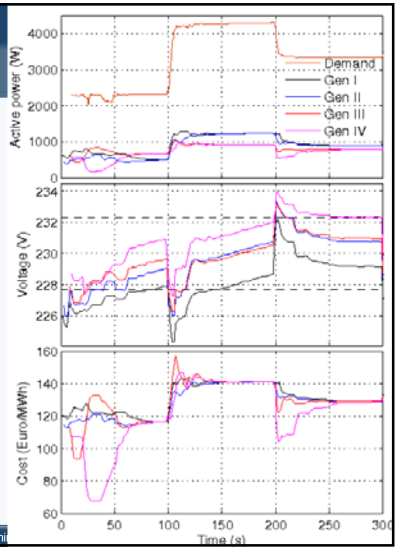
- introduction
- separating functionality & fault tolerance
- overlay network
- ➔ application example
- conclusion

Evaluation: AEN autonomous electricity networks

Experiments

- 4 converters operating as generators
- island operation





Overview

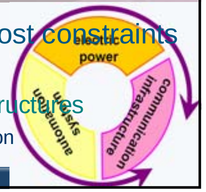
- introduction
- separating functionality & fault tolerance
- overlay network
- application example
- ➔ conclusion

Fault tolerance: Embedded controllers require robust system architecture

- to deal with
  - internal problems
    - physical faults in controllers or communication
  - external dynamic environment
    - changing interconnection topology
    - bandwidth reduction
- to exploit redundancy & diversity
  - in interconnections and resources
- for quantitative & qualitative assessment
  - explicit fault and failure model
  - dependability, time & cost constraints

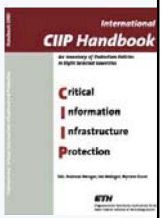
Design methodology for dependable embedded systems

- explicit fault and failure models of ICT infrastructure
- dependability modeling
  - reliability, safety, …
  - fault prevention, ~tolerance, ~removal, ~prediction
- instantiation of modular middleware architecture
  - dependable communication & computation
  - in dynamic environments
- assessment of dependability, time & cost constraints
  - conceptual evaluation and fault injection
  - interdependencies on other critical infrastructures
    - energy, telecom, information & fault propagation



Outlook
ICT: key for critical infrastructures

- survivability of infrastructures
  - self-healing, autonomous system architecture
    - diversity for reduced vulnerability
  - middleware for heterogeneous, distributed systems
    - adaptive, reconfigurable in dynamic environment
    - interoperability
- assessment
  - infrastructure vulnerability / interdependency
    - fault propagation
  - performability evaluation of real applications
    - service point-of-view: end-to-end properties
    - focusing on interconnection of subsystems



Conclusion

- context
  - new threats and vulnerabilities emerge from tight coupling of power - info infrastructures and from evolving control systems
- vision
  - resilient power control *in spite of* threats to their information infrastructures
- projects
  - crutial.cesiricerca.it
  - grid.jrc.it
  - www.kuleuven.be/esat/electa