57th IFIP WG 10.4 Meeting

Ishigaki Island, Japan — January 21-25, 2010

Workshop on Dependable Operating Systems

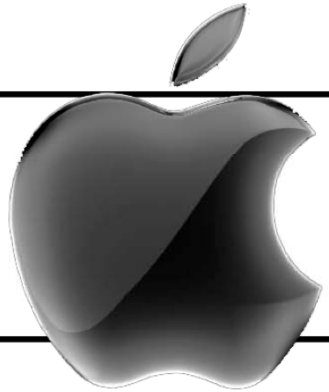
Wrap-Up:
A Meeting among
Open Systems & Operating Systems

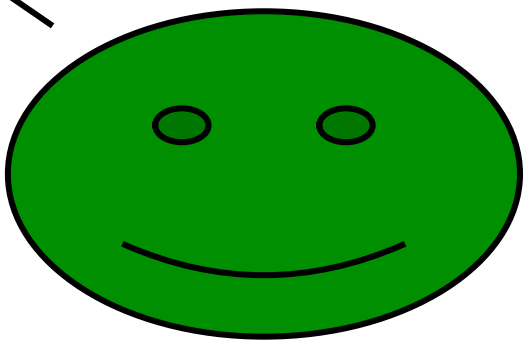
Université de Toulouse

Jean Arlat

[jean.arlat@laas.fr]

LAAS-CNRS

An  a Day Keeps the Docteur Away…



An a Day Keeps the Docteur Away…

YAPS!*

* Yet Another Panic Screen!

NB. Just meant to add to Xavier's Introduction to the Workshop ;-)

Dependable Open Systems: DEOS project

- **Dependability wrt to incompleteness, changes/uncertainties: Open Systems <—> Resilience (EU FP6 ReSIST NoE: www.resist-noe.org)**

- **Accountability: showing evidence of "best effort" wrt to risk management**

  —> "process-orientation" / "product-orientation"?

- **New standards**
  - **Adaptability, usability, manageability, not so well addressed…**
  - **System profiling under incompleteness and uncertainty**
    Assurance Case (AC): stake holders analysis, consensus building, decision making

  —> Are AC "open" enough? — "Assurance" vs. "Insurance"?

- **Evidence-based Computing**
  - **Architectural framework for Open Systems dependability**
  - **Based on monitoring (insertion of probes) — D-Box ≈ flight recorder**

  —> Worries about any "probe effect"?

- **Tooling Framework**
  - **Model checking (ABSL) and Type Checking (insertion of dynamic checks)**
  - **Qualitative metrics and D-cases — first,Operatings Systems, then Open Systems**

Hypervisors and Virtualization

- Virtual CPUs to support execution of replicated entities on single machine
- SeL4 μkernel: the core element (TCB) for high confidence requirements
- SPUMONE (Composition kernel): to accomodate HW layer evolution (multicore): application/OS reuse and core reconfiguration management
  Guest OS Kernels: user/kernel modes?

  —> Generic platform: cyber-physical, cloud computing, ambient-inteligence ?

Architecting Operating Systems

- **ArSec: COTS OSs, mature, powerful, flexible, but vulnerable**
  —> Call for data flow control, diversification and advances in virtualization to the rescue!
  Still needs some "Trusted Computing Base" to rely upon …
  (see next slide)

- **P-Bus: Extension of OS services wrt Dependability features**
  Specification of properties (implemented into source code)
  —> on-line model/type checking — limited overhead,
  the device driver issue, again… ≈ "Wrapping" technique

- **GENESYS: An architecture supporting "Distribution" of OS services at HW level over MPSOCs ≠ monolithic OS**
  —> better isolation and management, including error handling

Proving/Verifying Operating Systems

- **SeL4 μKernel — Formally-verified OS Kernel**
  - Confinement (Safety properties?) -> Access Control
    —> Spec — Design* — C-code (restricted) * Haskell prototype
  - Assumptions: Compiler/Linker, Assembly code, HW, Boot code, …
  - Cost-effective, Drivers… **see next slide**, Multicore **(GENESYS?)**

- **RT OS verification — OSEK/VDX**
  [suitable for AUTOSAR (AUTomotive Open System Architecture) framework]
  - Focus on Scheduler
  - Promela description and SPIN checking
  - Powerful computer cluster needed, State explosion
  - Application to automotive industry:
    complementarity of formal methods and testing

The Drivers

- **Most part of Kernels, Most failures, Most evolutions, …**
- **Impact evidenced by several studies:**
  failure data analyses and fault injection experiments
  (see subsequent slide)
- **What can be done about it?**
  - **Restrict driver's operation to user mode?**
  - **SW Language-based approach: Coccinelle —> Automatize changes**
  - **Better specify the interface between kernel and drivers**
    —> Towards a "DPI" ≈ API for drivers
    CDI (common driver interface), DDI (device driver interface),
    DKI (driver kernel interface), …
  - **"Shadow" driver (Swift et al. OSDI 2004)**
  - **See also the "wrapping" concept: (Rodríguez et al. EDCC-2002)**
    ≈ on-line checking by P-Bus

Assessment, Metrics

- COTS OSs…
- Fault injection: a pragmatic approach to get useful insights
- Objective characterization of behaviors in presence of faults: impact of hardware platform, discriminate OSs, drivers,
- Fault injection
  - What? parameter, bit-flip, … ≈ fault vs. error injection
  - Where? data representation ≈ code pre-analysis (Chalmers)
  - When? activity ≈ stress-based fault injection (UIUC)

  -> "Fault collapsing" techniques in Testing (equivalence classes)
- Metrics -- qualitative, quantitative — D-Case: Evidence
  divide, conquer (combine?), visualize

  —> • What, Where, When to Observe?
  Kernel reaction/status + Workload behavior?

  —> • FP5 EU Project DBench (www.laas.fr/DBench)
  • WG 10.4 SIG DeB (www.dependability.org/wg10.4/SIGDeB)

Are OPEN SYSTEMS part of OPERATING SYSTEMS?

Are OPEN SYSTEMS part of OPERATING SYSTEMS?

Yes, … to a large extent:

OPEratiN**g SYSTEMS**