

Evolutions in Open Source Control Software
Embedded Systemen 2009
Peter Soetens,
The SourceWorks

Something to think about...

Robots move things around,
Robot software moves data around.

A Question

Which of these machine controllers has been built using open source software ?



Evolutions in controllers

Concept of

- **Move data to hardware**
- Move data around
- Building these systems

Open Source in

- ▪ **Controller RTOS**
- ▪ Controller platform
- ▪ Tooling

Evolutions of the RTOS

- Linux moves to real-time scheduling
 - 20% of our RT application is RT bounded
- Linux supports an immense range of hardware
 - *'All device drivers belong in the Linux kernel'* (GKH)
- Controller platforms become networked nodes
 - Not only RT, but whole system reliability
- Portability
 - The less we notice HW, the happier we are

Our Software picks of the day

- Determinism:
 - Xenomai, the Hard-RT OS fully integrated with Linux
 - PREEMPT-RT, a Linux kernel on steroids
- Portability:
 - Linux in general runs on any 32/64 bit system
- Ease:
 - LTT/LTTng visualize what our RT application is doing.

Our Hardware picks of the day

- There are various open-source friendly busses out there
 - Good old CAN – reaching its limits but universal
 - SERCOS III – open, Ethernet based, more than motion
 - EtherCAT (?) – if you can get a compatible license that is

Evolutions in controllers

Concept of

- Move data to hardware
- **Move data around**
- Building these systems



Open Source in

- Controller RTOS
- **Controller platform**
- Tooling

Evolutions in Moving data around

- How can Open Source help building our control architectures ?
 - The hardest part is integrating # systems
 - Controllers get distributed
 - Or are accessible over a network
- We can use some help !

Component deployment in a SOA

Difficult words, easy concept

- Component Based Design =
 - Partitioning the application in distributable, *data* exchanging software components
 - Specifying what a component requires and provides
 - Resources
 - Services
 - IO ports
 - ...

Component deployment in a SOA

Difficult words, easy concept

- Service Oriented Architecture =
 - Be informed of appearing and disappearing *services*
 - Networked environments
 - Restarts of components
 - Upgrades
 - Reconfigurations
 - Complex dependency graphs

Which adds up to...

- Innovative controller design
 - Reliability in an unreliable environment
 - Automatic (re-)connection of components
 - Seamless distribution
- The Open Source projects
 - OSGi (Java, standard)
 - Open Robot Control Software – Orocos (C++, free)

Evolutions in controllers

Concept of

- Move data to hardware
- Move data around
- **Building these systems**

Open Source in

-
- Controller RTOS
-
- Controller platform
-
- **Tooling**

Open Source Tooling

- Some help from our friends
 - Eclipse platform !
 - Virtualisation of target with VirtualBox !
 - GCC + Gnu Debugger 7 + Eclipse 3.5 + ccache + gold + ... !
 - Valgrind memory checker x86/ppc (32/64bit) !
 - DENX toolchain solid creation of your target OS !

Exciting!

Don't forget...

Open Source is a technology and making a team familiar with a technology has a cost, regardless of whether it is open sourced or not.

(the end)