eman ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

Escuela Universitaria de Ingeniería Vitoria-Gasteiz Ingeniaritzako Unibertsitate Eskola Vitoria-Gasteiz

Improving Design, Operation and Security in CPS Using ROS Middleware and Lightweight Virtualization

Pablo González-Nalda,
Ismael Etxeberria-Agiriano, Isidro Calvo,
Mari Carmen Otero

University College of Engineering, Vitoria-Gasteiz (UPV/EHU, Spain)

Virtual Concept Industrie 4.0,
November 26^{th}, 2015

This presentation

Contenidos de la presentación

Abstract

Cyber-Physical Systems (CPS)

- Multidisciplinary
  - Embedded Computation
  - Control Theory
  - Communication Networks
- Complexity
- Generic Architecture for CPS
  - Free open Software
  - Common components from ICT
  - Inexpensive or free
  - For Industrial Automation and Mobile Robotics
  - Security concerns
  - Modularity and Operations: ROS+Docker
  - Proof of Concept

Cyber-Physical Systems (CPS)

Systems that connect computers and the physical environment

These systems combine

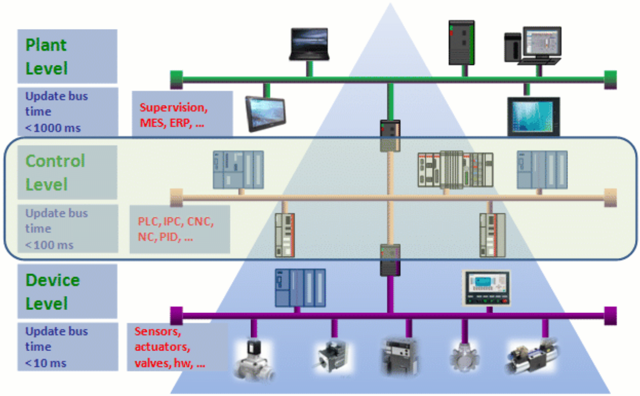
- Embedded Computation and Real-Time
- Control Theory
- Communication Networks

Even more frequent in domotics and industry

CPS systems are difficult to study because of the heterogeneity

Based on complex theory and technologies:

- Real Time systems
- Electronics
- Concurrent and distributed computation



Automation levels

Constraints:

- Cost of equipment and technical support
- Robustness
- Ease of installation, deployment, management, growth
- Security Changes

Change from expensive proprietary protocols to more affordable Ethernet-based networks

Exceptions:

- Legacy systems
- Hard real-time systems
- Electrically noisy plants

Industrial Automation and ICT components

Information and Communication Technology (ICT) components lower the cost.

Problems:

- No isolated from Internet (office, plant)
- Cyber-security threats (access, DoS)
- Introduce new vulnerabilities in old systems

Free Open Source Software: GNU/Linux

The best choice frpm ICT: Free Software. Why?

Freedom in the context of software allows anybody to read and understand, cooperatively change and improve programs.

GNU/Linux: Family of Free operating systems.

At home:

- Smartwatches and phones (Android, Ubuntu)
- Smart TVs and Entertainment Systems
- Desktop PCs, home automation

Professional environments:

- Embedded systems and Single Board Computers
- Intranet and Internet servers
- High Performance Computing
- Robots

Drawbacks of free systems

Drawbacks of free systems:

- Complexity of mixed systems and adaptation
- Reticence to replace proven proprietary field bus technologies

But:

- Legacy systems must face the increase of complexity in automation processes
- Deal with security holes in old designs with new technologies

Containers

Lightweight virtualization or *containers* (like Docker containers)

- Isolate and protect the program from undesired interactions
- Wrap the application to be shared and managed
- Facilitates the standardization, sharing, development, deployment and security of applications

Contenidos

1 Introduction

2 Proposed Architecture

3 Prototype

4 Conclusions

Generic architecture

Generic architecture based on a network of ROS nodes encapsulated by Docker containers.

Docker completely isolates ROS nodes

Objectives:

- Improve security, as the attack surface is reduced and
- Ease system administration

ROS

Contenidos

What is ROS?

- It is a middleware
- It manages an event-driven network of nodes
- Nodes publish and subscribe topics
- It abstracts from hardware and OS
- Distributed object oriented network of automation systems

Hardware

Hardware: an Industrial Computer (a PC-104 with Linux) or prototypes with Raspberry Pi and Arduino-based hardware

Raspberry Pi

- Free-hardware low-cost *Single Board Computer* (SBC)
- Standard GNU/Linux Operating System
- Widely used in Education and Do-It-Yourself (DIY) systems
- A lot of information in Internet

Arduino

- Free affordable electronic system to read and write analog, digital and PWM signals
- Used to learn electronics and DIY
- Arduino-based *Programmable Logic Controllers* (PLCs) for Industrial Automation

From ICT to Industrial Automation:

- ROS has many extensions, plug-in modules
- For controlling industrial robots
- Also for EtherCAT and other proprietary protocols and hardware
- ROS module to control Arduino systems, so the coupling of a Raspberry Pi nanocomputer and an Arduino based slave-PLC
- The Arduino PLC can even be used as an RS-485 device to communicate with other systems connected to the field bus network
- A bridge between the Ethernet command network and the field bus

The problem of *intelligent* Automated Guided Vehicles (AGV) for in-plant material handling

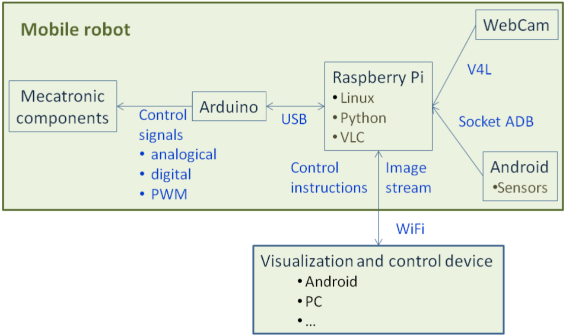
Automated Guided Vehicle.

Public Domain https://commons.wikimedia.org/wiki/File:IntelliCart1.jpg

Issues:

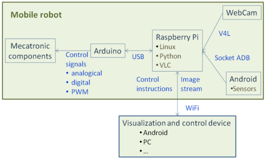
- Collision-avoidance and obstacle detection: hard problem
- Wireless communication to the plant high level logic for planning and orchestration
- Standard GNU/Linux OSs for controlling embedded real-time robotics

Then the architecture adapted to the problem of a CPS automated vehicle

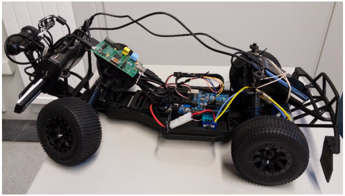


Proposed Architecture of the CPS.

Structure



- Arduino board receives commands from Raspberry Pi SBC
- These commands are sent to the AGV mechatronics
- The cameras and sensors feed the Raspberry Pi
- Raspberry Pi SBC integrates artificial intelligence for autonomous behaviour
- SBC reports its status to the plant control
- RasPi receives high-level commands, within ROS network



Implementation of the proof of concept in progress.

This scheme is a proof of concept prototype to demonstrate the feasibility of the architecture.

It also allows to get valuable information about the design of low cost and low power autonomous robots.

Conclusions and future work

The goals fulfilled by applying these technologies can be summarized as follows:

- Object oriented, data-driven and event-driven design
- Modularity and use of software agents
- Better communication among different network levels
- Communications security through mature firewall and virtual network technologies
- Improved security and ease of operations with lightweight virtualization
- Free software benefits: Collaboration allows sharing developments and security improvement
- Inexpensive free hardware such as Raspberry Pi and Arduino PLCs
- Ethernet standards

Modular architecture:

- Decoupling the complexity of constructing complex CPS
- Simpler subsystems developed in concurrence
- Subsystems independently by specialized teams

Also, since the architecture uses very common devices, there exist a broad community of users and wide availability of related documentation.

Future work

Future work

- Image analysis
- Neural Networks and Evolutionary Algorithms
- Analyse the impact of security measures such as encryption of communications
- Real-time constraints

Acknowledgements

Projects EHU13/42 and CPS4PSS ETORTEK14/10

Thanks a lot…
Questions?

eman ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

Escuela Universitaria de Ingeniería Vitoria-Gasteiz
Ingeniaritzako Unibertsitate Eskola Vitoria-Gasteiz

Improving Design, Operation and Security in CPS Using ROS Middleware and Lightweight Virtualization

Pablo González-Nalda,
Ismael Etxeberria-Agiriano, Isidro Calvo,
Mari Carmen Otero

University College of Engineering, Vitoria-Gasteiz (UPV/EHU, Spain)

Virtual Concept Industrie 4.0,
November 26^{th}, 2015

This presentation