Dependable Cyber-Physical Systems Networking: An Approach for Real-Time, Software Intensive Systems

Miroslav Švéda*. Ondřej Ryšavý*

**Faculty of Information Technology, Brno University of Technology, Bozetechova 2, CZ-612 66 Brno, Czech Republic;*
(Tel: +420-54114-1288; e-mail: {sveda, rysavy}@fit.vutbr.cz).

Abstract: This paper describes principles of an cyber-physical system networking design that props safety and security of the consequence applications. After reviewing basic features of cyber-physical systems, the main attention is focused on concepts of IP networking fitting cyber-physical systems applications and on the concepts of proposed design and development environment.

Keywords: embedded systems, cyber-physical systems, dependability, safety, security, computer communication networks, communication protocols.

1. INTRODUCTION

The design of well thought-out computer-based systems should consider namely functionality and dependability measures (viz. Sveda, Trchalik and Ocenasek, 2009). Functionality means services delivery in the form and time fitting requirements specification, where the service specification is an agreed description of the expected service. Functionality properties should be realized efficiently and cost-effectively, so reachable performance and simplicity of implementation belong to the checked properties. Dependability is that property of a system that allows reliance to be justifiably placed on the service it delivers. Security is concerned with the risks originating from the environment and potentially impacting the system, whereas safety deals with the risks arising from the system and potentially impacting the environment, see Fig. 1**.** As e.g. Akela, Tang, McMillin (2010) pointed out, the development of computer-based systems, where safety or security are important aspects, follows much the same approach for assessing risks involved with the systems.

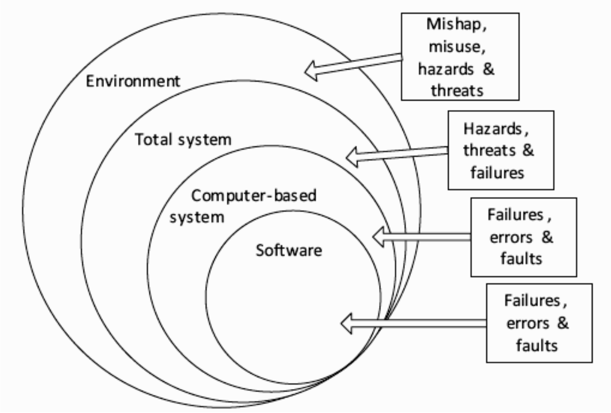
Environment
Total system
Computer-based system
Software
Mishap, misuse, hazards & threats
Hazards, threats & failures
Failures, errors & faults
Failures, errors & faults

Fig. 1. A layered view.

Computer-based systems alone do not pose any risk. It is when they are put in a total system context that they have the potential of contributing to hazards or threats. This applies to both security and safety, and has to be the basis for any risk assessment. Risks happen classed according to standards in the following way:

- Harm – is the "physical injury or damage to the health of people or damage to property or the environment" (IEC, 2008).
- Hazard – is a "potential source of harm" (IEC, 2008).
- Threat – is the "potential cause of an incident which may result in harm to a system or organization" (ISO, 2005).
- Failure – is a "termination of the ability of a functional unit to provide a required function or operation of a functional unit in any way other than as required" (IEC, 2008).
- Error – is the "discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition" (IEC, 2008).
- Fault – is an "abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function" (IEC, 2008).

In the safety field the benefits of a system and its features have to be balanced against the possible accidental harm it might impose, while the security field needs to consider such benefits against possible malicious harm as mentioned by Raspotin and Opdahl (2013), see Fig. 2.

The boundary between the total system and environment can often be unclear, just as how comprehensive the environment has to be defined in the development process.

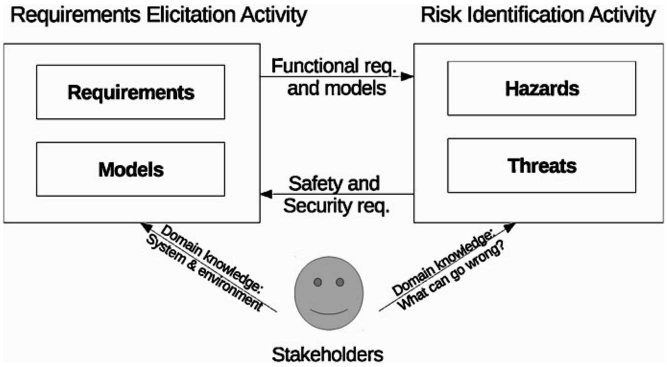
Requirements Elicitation Activity
Risk Identification Activity
Requirements
Models
Functional req. and models
Safety and Security req.
Hazards
Threats
Domain knowledge: System & environment
Domain knowledge: What can go wrong?
Stakeholders

Fig. 2. Requirements elicitation and risk identification activities.

The integration of physical systems and processes with networked computing has led to the emergence of a new generation of engineered systems: Cyber-Physical Systems (CPS), see Lee (2010). Those systems use computations and communication deeply embedded in, and interacting with, physical processes to add new capabilities to physical systems. This paper presents a safety and security-based approach to networked CPS design that offers reusable design patterns for applications dedicated to various domains.

After introduction that presents basic requirements dealing with computer-based systems, embedded systems and related applications, the remainder of the paper is structured as follows. The next section describes the overall objectives of the related research project. While the sub-section 2.1 introduces the architecture, the subsection 2.2 focuses on appropriate networking stemming from Internet-compatible protocols. Section 3 then provides an overview of related work, in particular multi-domain CPS and how comparable concepts can be used to intelligently manage the design and development environment infrastructure. Section 4 outlines the architectural concept of the intended case study and discusses innovations the project addresses either as completed or considered for the following research. The paper concludes with a summary and description of future work possibilities.

2. OBJECTIVES OF THE PROJECT

This paper reviews partial results of the long-term project focused on embedded or respective cyber-physical systems and on their architecture, applications and associated development environments. Preceding achievements were presented subsequently by the papers (Sveda and Vrba, 2005), (Sveda and Vrba, 2010), (Sveda and Vrba, 2011) and (Rysavy, Sveda and Vrba, 2012). The current phase of this project aims namely at networking concepts for CPS application designs and on the conception of projected design and development environment.

2.1 CPS Networking Overview

CPS networking can stem from hierarchically interconnected networks, mostly Internet, local area wired and wireless networks, and wireless sensor networks. Internet access to individual components of distributed embedded systems can be based on both wired and wireless LAN technologies, predominantly on IEEE 802.3 and related Ethernet standards, and on IEEE 802.11 WiFi and associated wireless LAN protocols. Particular embedded systems and their components can be attached directly to Ethernet with TCP/IP protocol stack, but also indirectly or exclusively through various wired Fieldbuses or wireless technologies such as IEEE 802.11b and IEEE 802.15.4 with related ZigBee. Sensor networks bring an important pattern with single base station connected to a wired network on one side and wirelessly to smart sensors on the other side. When sensors are clustered, the base station communicates to cluster heads and through them to individual sensors. Next patterns emerge with mobile nodes and ad-hoc networking.

2.2 TCP/UDP/IP Networking

This paper focuses on Internet-compatible protocols, i.e. protocol profiles stemming from IP or IP-mobile enabling direct interconnection of CPS nodes or components to Internet. From that viewpoint all network nodes can also be considered as IP routers, which may well provide also gateway functions to non-IP subnets, see Fig. 3.

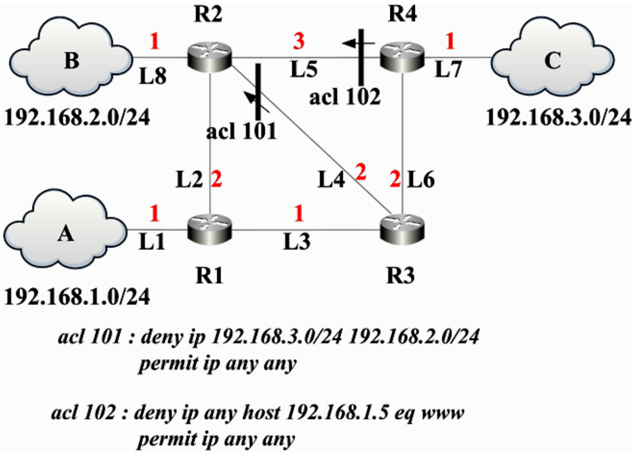

Fig. 3. IP network example interconnecting A, B and C subnets.

The figure above depicts a network model, which is a 3-tuple $N = \langle R_N, L_N, F_N \rangle$, where

- R_N is a finite set of network devices,
- $L_N \subseteq R_N \times R_N$ is a finite set of links between routers, such that for every physical link between R_1, R_2 there is a pair of channels $l_{12} = \langle R_1, R_2 \rangle$, $l_{21} = \langle R_2, R_1 \rangle$, and
- $F_N = \{f : P \rightarrow \{\text{true, false}\}\}$ is a finite set of filtering predicates and P is a set of all possible packets.

A filtering predicate $f(p) \in F_N$ is able to determine whether a packet p is allowed to be send. This function is defined so that it uniformly represents the interpretation of Access

Control List (ACL) and routing table information adequate to the link. A simple example is a filter f (p)

$$f(p) = \neg(p.proto = Tcp \wedge p.dstPort = 80)$$

that turns down all web traffic, i.e. TCP packets with destination port 80. Both ACL and routing information of a network node can be translated to a filtering predicate.

3. RELATED WORK

Many of the embedded systems-related studies and efforts in the past have focused on the challenges the physical environment brings to the scientific foundations of networking and information technology (Lee, 2010). However, the full scope of the change enabled by introducing CPS as a new branch of science and technology provides much more than restructuring inside this domain. The new approach can turn entire industrial sectors into producers of CPS. Actually, CPS is about merging computing and networking with physical systems to create new capabilities and improve product quality.

Cyber-physical systems denote a new modeling paradigm that promotes a holistic view on real-world – and therefore complex – systems. These systems have been studied before from various particular perspectives using paradigms like ubiquitous and distributed computing or embedded and hybrid systems. The above mentioned facts require also another approach to the design of such systems respecting from the beginning of design process the application domain that influences quality-of-service requirements such as real-time behavior, safety and security, but also precision, reliability and other non-functional properties affecting attributes specified usually by official standards (Donzelli and Basili, 2006).

In a CPS application, the function of a computation is defined by its effect on the physical world, which is in this case not only a system environment, but evidently also a component of the designed application system (Akela and Tang and McMillin, 2010). Therefore, proper design environments should be used to improve or at least to enable efficiency of the design process. In cyber-physical systems the passage of time becomes a central feature — in fact, it is this key constraint that distinguishes these systems from distributed computing in general. Time is central to predicting, measuring, and controlling properties of the physical world: given a (deterministic) physical model, the initial state, the inputs, and the amount of time elapsed, one can compute the current state of the plant. This principle provides the foundations of control theory. However, for current mainstream programming paradigms, given the source code, the program's initial state, and the amount of time elapsed, we cannot reliably predict future program state. When that program is integrated into a system with physical dynamics, this makes principled design of the entire system difficult. Instead, engineers are stuck with a prototype-and-test style of design, which leads to brittle systems that do not easily evolve to handle small changes in operating conditions and hardware platforms. Moreover, the disparity between the dynamics of the physical plant and the program seeking to control it potentially leads to errors, some of which can be catastrophic (Raspotin and Opdahl, 2013).

4. DESIGN AND DEVELOPMENT ENVIRONMENT

Development systems, see e.g. (Eidson et al., 2009) or (Raspotin and Opdahl, 2013), should support important concepts and methods by their tools for entire design and development life cycle of applications belonging to considered application domains. The final toolset related to the discussed design framework will necessarily include also original methods and tools (Lee, 2010).

At the beginning, the development means target predominantly front-end parts of specification and design, namely formal specification, verification and rapid prototyping. Moreover, a special support is dedicated to prop up IP networking techniques. First results accomplished in this direction were published (Sveda et al., 2010).

Conventional verification techniques to be used in the development environment have high memory requirements and are very computationally intensive. Therefore, they are unsuitable for real-world CPS systems that exhibit complex behaviors and cannot be efficiently handled unless we use scalable methods and techniques, which exploit fully the capabilities of new hardware architectures and software platforms (Lee, 2009). High-performance verification techniques focus on increasing the amount of available computational power. These are, for example, techniques to fight memory limits with efficient utilization of external techniques that introduce cluster-based algorithms to employ aggregate power of network-interconnected computers, or techniques to speed-up the verification on multi-core processors.

Researching CPS models consist of capturing characteristics of CPS. We study existing and propose new models for common architectural and behavioral artifacts and communication patterns of the CPS domain.

To be more explicit, at the beginning we are going to define models of applications using Ptolemy II framework (see http://ptolemy.berkeley.edu/ptolemyII) extended by existing formal tools, and we will study the possibility to integrate the formal verification methods for these hybrid models. It would require examining carefully the semantics bound in different models and define precise transformations to extract verifiable models from design models.

Domain specific modeling languages (DSML), contrary to the universal modeling languages, are specifically customized to the area of problems being solved (Halfar and Rab and Rysavy and Sveda, 2012). Using DSML approach, the modeling of a system is itself preceded by the phase of meta-modeling of the application domain. We plan to propose a DSML for the reliable real-time embedded devices in smart sensor and control networks domain and provide formal semantics for this language that should enable applications of formal methods for transformation and verification of CPS properties.

We will research possibilities to apply existing formal methods to the models generated from the specifications written in CPS-DSML. The models describe the system being developed at different levels and views. Automated tools should support inter-model validation. Thus our primary concern is to demonstrate how tools based on formal methods can proof the inter-model consistency and property preservation. For instance, model of software components, which behavior is driven by discrete means of computing should be in consistency with lower level model of hardware processing units and also with same level model of abstract environment behavior. The difficulty and novelty lies in consideration that different models obey different means of computing.

Designed development environment prototype will include tools and methods that can be used to approach demonstration and experimenting with the selected application area. We assume that various methods will be experimentally implemented as software tools to show the capability of the approach on non-trivial use cases. New design patterns and components will be created and verified in frame of case studies. These case studies will serve to gather experience in development of CPS. The work should conclude by critical evaluation of the proposed approach, showing the strength aspects of considered method and revealing drawbacks that deserve further research.

5. CONCLUSIONS

The paper deals with principles of a launching research focusing on CPS design environment with regard to networked CPS applications. In this paper, we demonstrate also porting the problem of security analysis of TCP/IP based computer networks to CPS domain.

There are also various possible extensions to the method. The further work should be focused on refining classification of properties and on proposing an adequate extension of the design and development environment including specification language and the verification procedure. For performing practical experiments it is necessary to implement reliable and effective tools that would improve and extend the currently available trial tools, which need to be sometimes manually supported.

ACKNOWLEDGMENT

The authors acknowledge contributions to this work by their colleagues Petr Matousek, Jaroslav Rab, Radimir Vrba and Patrik Halfar. This work has been carried out with a financial support from the Czech Republic state budget through the IT4Innovations Centre of Excellence, EU, CZ 1.05/1.1.00/02.0070CEZ and through the MMT project no. MSM0021630528: Security-Oriented Research in Information Technology, by the Technological Agency of the Czech Republic through the grant no. TA01010632: SCADA system for control and monitoring RT processes Also, the author was partially supported by the Ministry of Industry and Trade through the grant no. FR-TI1/037 of Ministry of Industry and Trade: Automatic Attack Processing.

REFERENCES

Akela, R., Tang, H., and McMillin, B.M. (2010). Analysis of flow security in cyber-physical systems. *International Journal of Critical Infrastructure Protection,* 3(3-4), pp.157-173.

Donzelli, P., Basili, V. (2006). A practical framework for eliciting and modeling system dependability requirements: Experience from the NASA high dependability computing project. *The Journal of Systems and Software,* 79(1), pp.107-119.

Eidson, J.C., Lee, E.A., Matic, S., Seshia, S.A., Zou, J. (2009). *Time-centric Models For Designing Embedded Cyber-physical Systems*, EECS Department, University of California, Berkeley, Technical Report No. UCB/EECS-2009-135.

Halfar, P., Rab, J., Rysavy, O., Sveda, M. (2012) A formal authorization framework for networked SCADA systems, In *Proceedings IEEE ECBS*. Novy Sad, RS, IEEE CS, 2012, pp.298-302.

Lee, E.A. (2009). Computing Needs Time, *Communications of the ACM*, 52(5), pp.70-79.

Lee, E.A. (2010). CPS Foundations, In *Proceedings DAC'10*, Anaheim, California, US, ACM, pp.737-742.

Raspotin, C., Opdahl, A. (2013). Comparing risk identification techniques for safety and security requirements. *The Journal of Systems and Software*, 86 (5), pp.1124-1151.

Rysavy, O., Sveda, M., Vrba, R. (2012). A Framework for Cyber-Physical Systems Design - A Concept Study, In *Proceedings ICONS* 2012, Saint Gilles, Reunion Island, US, IARIA, pp.79-82

Sveda, M., Vrba, R. (2001). Executable specifications for distributed embedded systems. *IEEE Computer*, 34(1), pp.138-140.

Sveda, M., Trchalik, R., Ocenasek, P. (2009). Design of networked embedded systems: An approach for safety and security, In *Preprints of IFAC Workshop on Programmable Devices and Embedded Systems PDeS* 2009, Ostrava, CZ, IFAC, pp.131-136.

Sveda, M, Vrba, R. (2010). An Embedded Application Regarded as Cyber-Physical System, In *Proceedings of the Fifth International Conference on Systems ICONS* 2010, Les Menuires, FR, IEEE CS, pp.170-174.

Sveda, M., Rysavy, O., Matousek, P., Rab, J. Čejka, R. (2010). Security Analysis of TCP/IP Networks -- An Approach to Automatic Analysis of Network Security Properties, *Proceedings of the International Conference on Data Communication Networking ICETE-DCNET*, Athens, GR, INSTICC, pp.5-11.

Sveda, M, Vrba, R. (2011). A Cyber-Physical System Design Approach, In *Proceedings of the Sixth International Conference on Systems - ICONS* 2011, St. Maarten, AN, IARIA, 2011, pp.12-18.

Uzunov, A.V., Fernandez, E.B., Falkner, K. (2012). Engineering security into distributed systems: A survey of methodologies. *Journal of Universal Computer Science,* 18(20), pp.2920-3006.