Dependability in Cyber-Physical Systems Network Applications

MIROSLAV SVEDA
Faculty of Information Technology
Brno University of Technology
Bozetechova 2, 61266 Brno
CZECH REPUBLIC
sveda@fit.vutbr.cz http://www.fit.vutbr.cz/~sveda/

Abstract: - This paper deals with the role and implementation of the notion "dependability" in distributed cyber-physical systems. It aims at principles of 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 not only on concepts of IP networking fitting typical cyber-physical systems applications, but also on proposed design and development environment. Stemming from a brief state-of-the-art review of dependability as considered in the domain, the paper focuses on networking for cyber-physical systems applications. The aim of the article is to select the fitting methods that enable to support the related specification and design approach for a broad-enough class of distributed cyber-physical systems applications.

Key-Words: - Cyber-Physical System; Dependability; Networking; Specification and Design.

1 Introduction

The design of well thought-out computer-based systems should consider namely functionality and dependability measure, see e.g. [12]. Functionality means services delivery in the form and time fitting requirements specification, where the specification is an agreed description of the expected service. Functionality properties should be realized efficiently and cost-effectively, therefore, reachable performance and simplicity of implementation belong to the checked properties.

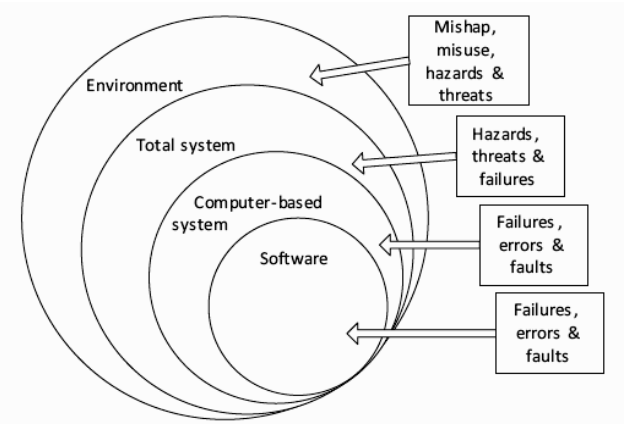

Fig. 1. A layered view

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. [17]. As e.g. Akela, Tang and McMillin [1] 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.

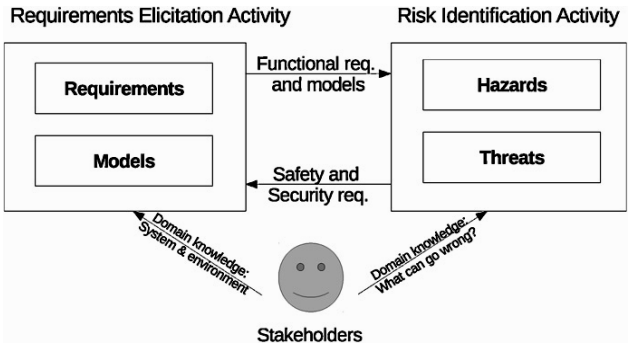

Fig. 2. Requirements elicitation and risk detection

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, see e.g. [17]:

- 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 e.g. by Raspotin and Opdahl in [8], 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. 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 [7]. 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.

The current paper reviews also partial results of the long-term project focused on embedded or cyber-physical systems and on their architecture, applications and associated development environment. Preceding achievements were presented subsequently by the papers [13], [16], [9] and [17]. The current phase of this project aims namely at networking concepts for CPS application designs and projected design and development environment.

Structure of the paper is as follows. The next section brings basics of CPS networking stemming from embedded system networks concepts, which originate from industrial applications. Section 3 presents a case study that refines and exemplifies a typical industrial application of that domain including a reusable application structure. Section 4 introduces a formal model of CPS network components reachability that enables to verify safety and security properties of system configurations. Finally, section 5 concludes this paper.

2 CPS Networking Basics

Contemporary industrial distributed computer-based systems encompass, at their lowest level, various wired or wireless digital actuator/sensor to controller connections. Those connections usually constitute the bottom segments of hierarchical communication systems that typically include higher-level fieldbus or Intranet backbones. Hence, the systems must comprise suitable interconnections of incident higher and lower fieldbus segments, which mediate top-down commands and bottom-up responses. While interconnecting devices for such wide-spread fieldbuses as CAN, Profibus, or WorldFIP are currently commercially available, some real-world applications can demand also to develop various couplers either dedicated to special-purpose protocols or fitting particular operational requirements, see [11].

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.

3 Case Study

This section describes a case study that demonstrates utilization of the originally developed framework. The deployed case deals with pressure and

temperature measurement and safety and security management along gas pipes. The related implementation stems from the IEEE 1451.1 model with Internet and the IEEE 1451.5 wireless communication based on ZigBee running over the IEEE 802.15.4, see [15] and Fig. 3.

The interconnection of TCP/IP and ZigBee provides an interface between ZigBee and IP devices. Each wireless ZigBee-based sensor group is supported by its controller providing Internet-based clients with secure and efficient access to application-related services over the associated part of gas pipes. In this case, clients communicate to controllers using a messaging protocol based on client-server and subscribe-publish patterns employing 1451.1 Network Block functions. A typical configuration includes a set of sensors generating pressure and temperature values for the related controller that computes profiles and checks limits for users of those or derived values. When a limit is reached, the safety procedure closes valves in charge depending on safety service specifications.

Security configurations follow the tiered architecture IP - ZigBee mentioned above. To keep the system maintenance simple, all wireless communication uses standard ZigBee hop-by-hop encryption based on single network-wide key, because separate pressure and/or temperature values, which can be even-dropped, appear useless without the overall context that can be hardly reconstructed from discrete quantities. Security in frame of Intranet subnets stems from current virtual private network concepts. The discussed application utilizes ciphered channels based on tunneling between a client and a group of safety valve controllers. The tunnels are created with the support of associated authentications of each client.

The application architecture, see Fig. 3., comprises several groups of wireless pressure and temperature sensors with safety valve controllers as local base stations connected to wired intranets that dedicated clients can access effectively through Internet. The WWW server supports each sensor group by an active web page with Java applets that, after downloading, provide clients on Internet with transparent and efficient access to pressure and temperature measurement services through controllers. Those controllers offer clients not only to securely access measurement services over systems of gas pipes, but also communicate to each other and cooperate so that the system can resolve safety and security-critical situations by shutting off some of the valves.

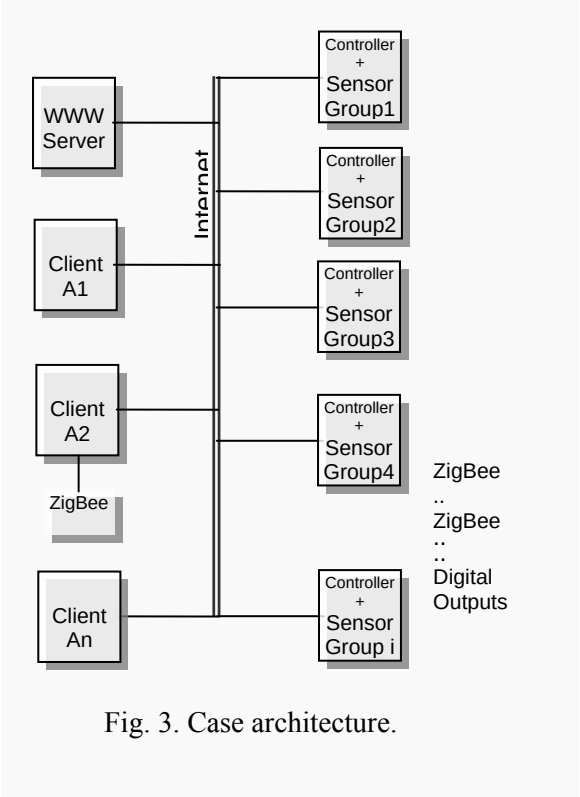

Fig. 3. Case architecture.

Each controller communicates wirelessly with its sensors through IEEE 1451.5 interfaces by 1451.5-ZigBee, which means ZigBee over IEEE 802.15.4. It fits application requirements, namely those dealing with power consumption, response timing, and management. The subscriber-publisher style of communication, which in this application covers primarily distribution of measured data, but also distribution of group configuration commands, employs IP multicasting. All regular clients wishing to receive messages from a controller, which is joined with an IP multicast address of class D, register themselves to this group using protocol IGMP. After that, when the controller generates a message by Block function publish, this message is delivered to all members of this class D group, without unnecessary replications.

The IEEE 1451.1 network model provides an application interaction mechanism supporting both client-server and publish-subscribe paradigms for event and message generation and distribution. Controllers play the role of clients or subscribers for the wireless part of the system network, and the role of servers or publishers for the wired part. Moreover, they compute temperature and pressure profiles, check the limit values and handle the safety valves.

4 Design and Development Tools

Development systems, see e.g. [3] or [8], should support important concepts and methods by their

ISBN: 978-960-474-374-2

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 as stressed by Lee in [7].

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 recently published [14].

Conventional verification techniques to be used in the development environment have enormous memory requirements and are 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 exploiting fully the capabilities of new hardware architectures and software platforms [6]. 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 means 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, we define models of applications using Ptolemy II framework (see http://ptolemy.berkeley.edu/ptolemyII) extended by existing formal tools, and we are studying possibilities to integrate the formal verification methods for those complex models. It would require examining carefully the semantics bound in different models and define precise transformations to extract verifiable models from design models [7].

Domain specific modeling languages (DSML), contrary to the universal modeling languages, are specifically customized to the area of problems being solved [4]. 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 fitting 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 a 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 prove 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 a 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 experimentation 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 Conclusion

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. 4.

The Fig. 4 depicts a network model, in which N is a 3-tuple: $N = \langle RN, LN, FN \rangle$, 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
- $F_N = \{f : P \rightarrow \{true, false\}\}$ is a finite set of filtering predicates and P is a set of all possible packets.

ISBN: 978-960-474-374-2

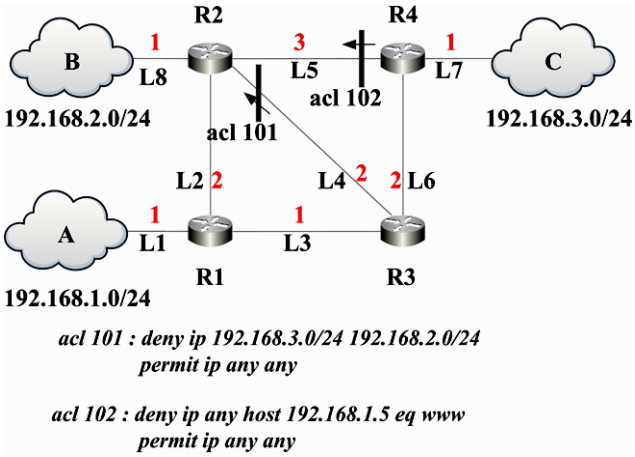

Fig. 4. IP example interconnecting A, B, C subnets.

The Fig. 4 depicts a network model, in which N is a 3-tuple: N = ‹RN, LN, FN›, 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 $l_{12} = ‹ R_1, R_2›$, $l_{21} = ‹ R_2, R_1›$, and
- $F_N = \{f : P \rightarrow \{true, false\}\}$ 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.

In our work we explore how security and safety properties can be verified under every network configuration using model checking [14]. The model checking is a technique that explores all reachable states and verifies if the properties are satisfied over each possible path to those states. Model checking requires specification of a model and properties to be verified. In our case, the model of network consists of hosts, links, routing information and ACLs. The network security properties are expressed in the form of modal logics formulas as constraints over states and execution paths. If a property is not satisfied, the model checker generates a counterexample that reveals a state of the network that violates the property. If the property is proved, it means, that the property is valid in every state of the systems.

This paper stems from the author's research projects with partial results published in [12], [13], [16], [4], [17] and [18]. The current paper addresses the role and implementation of the notion "dependability" in distributed cyber-physical systems.

ACKNOWLEDGMENT

This project has been carried out with a financial support from the Czech Republic state budget by the IT4Innovations Centre of Excellence, EU, CZ 1.05/1.1.00/02.0070CEZ, by the Brno University of Technology grant FIT-S-14-2299 and by the Faculty of Information Technology of that University.

The author acknowledges contributions to the presented work by his colleagues Ondrej Rysavy, Petr Matousek, Jaroslav Rab, Pavel Ocenasek, Roman Trchlik, Vladimir Vesely, Matej Gregr, Libor Polcak from the Faculty of Information Technology, and Radimir Vrba from the Faculty of Electrical Engineering and Communication.

References:

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] Krogh, B.H., Lee, E., Lee, I., Mok, A., Rajkumar, R., Sha, L.R., Vincentelli, A.S., Shin, K., Stankovic, J., Sztipanovits, J., Wolf, W., Zhao, W. (2008). Cyber-Physical Systems, Executive Summary, CPS Steering Group, Washington D.C., March 6, 2008.

[http://www.nsf.gov/pubs/2008/nsf08611/nsf08611.htm]

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

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

[8] 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.

[9] 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

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

[11] Sveda, M. (2004). A Design Framework for Internet-Based Embedded Distributed Systems, In: Proceedings of the International IEEE Conference and Workshop ECBS'2004, Los Alamitos, California, US, IEEE CS, 2004, pp.113-120

[12] 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.

[13] 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.

[14] 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.

[15] Sveda, M. (2010). Network Convergency and Modeling -- Design Experience with Routing SW for Intranets and Fieldbusses. In: Proceedings of the Fifth International Conference on Software and Data Technologies, ICSOFT 2010. Athens: Institute for Systems and Technologies of Information, Control and Communication, 2010, pp. 173-178.

[16] 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.

[17] Sveda, M., Rysavy, O. (2013). Dependable Cyber-Physical Systems Networking: An Approach for Real-Time, Software Intensive Systems. In: Programmable Devices and Embedded Systems. Laxenburg, IFAC, pp.1-4.

[18] Sveda, M, Vrba, R. (2013). Cyber-Physical Systems Networking with TCP/IP -- A Security Application Approach. In: IEEE Proceedings AFRICON 2013. New York: Institute of Electrical and Electronics Engineers, 2013, pp.101-106.

[19] 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.