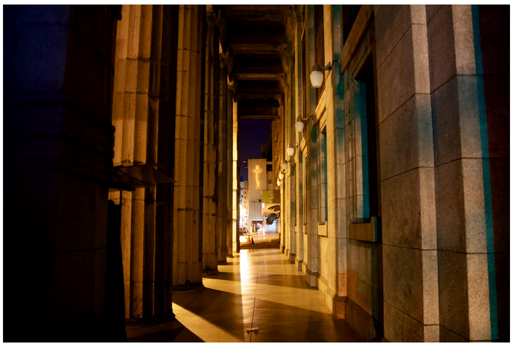
Requirements Engineering of Reliable IoT Systems : The First Step



Phillip A. Laplante
Pennsylvania State University
plaplante@psu.edu

Abstract - The Internet of Things refers to a system of interacting processing elements, sensors, devices and humans, connected wirelessly and to the Internet. Much has been written about designing, building and testing these applications, yet very little has been written with regard to requirements engineering. Because of the likelihood of continuous evolution and the potential for planned and eventual interaction with critical systems, however, a specialized approach to requirements engineering for IoT systems is essential. In this article we introduce the first principles of such an approach.

Keywords - internet of things, requirements engineering, soft systems methodology, systems engineering

1. Introduction

Anticipation surrounds the many exciting applications of the Internet of Things (IoT), also known as cyber-physical systems, but the realities of implementation are daunting. Ever changing requirements, end-to-end communications difficulties, low power concerns and anticipation of new uses present incredible challenges to systems designers. Of further concern is the unanticipated interaction of non-critical systems with safety critical systems where failure could directly lead to death or injury.

Requirements engineering is concerned with discovering the goals for, functions and constraints on systems [1]. These objectives are especially difficult in an IoT system with its emergent behavior, unforeseen interaction and potential contact with safety critical systems. Requirements engineering involves a set of life cycle processes, namely, elicitation/discovery, analysis and reconciliation, representation/modeling, verification and validation and requirements management. Agile software development provides an alternative model to waterfall style requirements engineering, but the agile approach is generally only suitable for pure software systems and certainly not for cyber-physical systems. Therefore the paradigm and techniques for cyber-physical must be considered as a special case of requirements engineering.

A conventional requirements engineering approach follows the sequence: mission identification; identification of stakeholders and their goals; identification of relevant requirements standards and applicable laws; requirements elicitation and derivation (including hazard identification); requirements analysis and reconciliation; requirements representation (creating a requirements document); requirements testing; and monitoring and maintenance of requirements. While this conventional approach is suitable for IoT systems, we need to treat the activities in each of the phases differently. In this article we focus only on the aspect of stakeholder and goal identification phase.

2. Systems Thinking and Requirements Engineering for IoT

There are several traditional approaches to stakeholder and goal identification. These usually start with the generating a mission statement, then using collective experience to identify stakeholder groups and deriving goals by engaging these groups. But in IoT systems the stakeholder groups and their goals are not so easily identified, given the evolving nature of the system. It seems that a particular systems thinking approach based on systems thinking is more appropriate. One such approach, Soft Systems Methodology (SSM), focuses on managing complexity, “thinking” systems and adaption and it is well suited for IoT applications. SSM consists of seven steps:

1. entering the problem situation,
2. expressing the problem situation,
3. formulating root definitions of relevant systems,
4. building conceptual models of human activity systems,
5. comparing the models with the real world,
6. defining feasible and desirable changes,
7. taking some action to improve the real world situation [2].

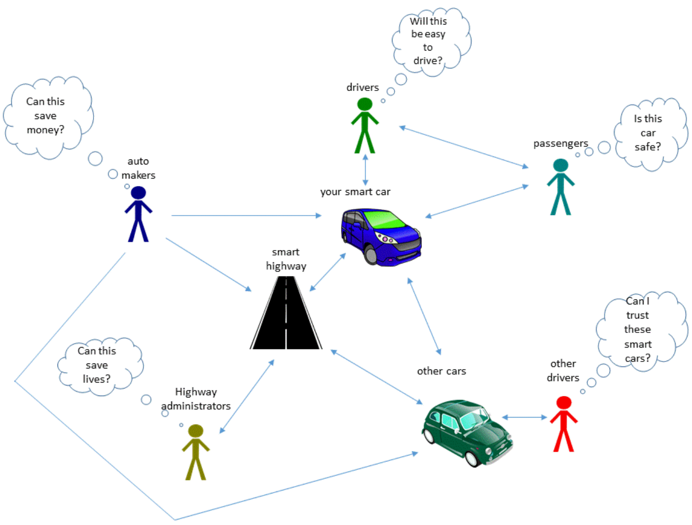


Figure 1: Rich Picture for smart car associated IoT

Steps 1 through 4 involve what we would consider the traditional identification of mission, stakeholders and their goals. Steps 5 through 7 are related to continuous improvement of an existing system and are not relevant to our IoT paradigm unless dealing with a situation in which the external conditions for the system can be adapted pre-deployment.

During steps 1 through 4, requirements engineers create conceptual models in various representations from back-of-the-napkin drawings and natural language, to SysML models and other formal representations until a desirable (or acceptable) model is created. In traditional, plan driven requirements engineering, these activities are usually conducted in a linear, waterfall fashion. In SSM these steps are iterative. The centerpiece of SSM is the "rich picture," which is a cartoon-like drawing based on informal rules.

Rich pictures depict various human actors in the system along with their goals wants and needs. Rich pictures, which resemble annotated Use Case diagrams or concept maps. The rich picture model can then be used as an entry point to the next step in the requirements process: elicitation. For example, consider a partial rich picture for a smart car IoT ecosystem (Figure 1), which represents a first iteration in steps 1 through 4 of SSM (we assume the mission statement was already defined).

The figure depicts one or more smart (or non-smart) cars and a smart highway system. Various interactions occur between these actors, stakeholders (such as drivers and passengers), and are shown as directed edges. At least one concern of each stakeholder is shown, though many more exist and are recorded in ancillary documentation. For example, drivers of smart cars are concerned with how easy it will be to interact with the smart car, but they are also concerned about safety, privacy, cost and more. Government highway administrators are concerned with safety first, but they are also interested in cost containment. Automakers want their smart cars to be appealing to consumers and they want the cars to be safe too. One may discover other missing information, functionality, even stakeholders. For example, what about smart tolls, advertising signs, construction equipment and so on. The possibilities for interaction, both planned and accidental, potentially even with critical infrastructure systems, are vast. A rich picture helps in identifying these gaps, omissions, misunderstandings and errors early in the system development life cycle.

3. Final Observations

All the activities of requirements engineering are critical, but in this short article we addressed only identification stakeholders and their goals using rich pictures. There is much more to the process, but the essence of has been introduced.

References

[1] Phillip A. Laplante, *Requirements Engineering for Software and Systems, Second Edition*, Taylor & Francis, 2013.

[2] Peter Checkland and Jim Scholes, *Soft Systems Methodology in Action*, John Wiley & Sons, 1999.

Phillip A. Laplante is Professor of Software Engineering at The Pennsylvania State University. He received his B.S., M.Eng., and Ph.D. from Stevens Institute of Technology and an MBA from the University of Colorado. He is a Fellow of the IEEE and SPIE and has won international awards for his teaching, research and service. Since 2010 he has led the effort to develop a national licensing exam for software engineers. He has worked in avionics, CAD, and software testing systems and he has published 27 books and more than 200 scholarly papers. He is a licensed professional engineer in the Commonwealth of Pennsylvania and a Certified Software Development Professional. He is also a frequent technology advisor to senior executives, investors, entrepreneurs and attorneys. His research interests are in software testing, requirements engineering and software quality and management. Prior to his appointment at Penn State he was a software development professional, technology executive, college president and entrepreneur. More information can be found at www.perso