Using Models to Address Challenges in Specifying Requirements for Medical Cyber-Physical Systems[*]

Anitha Murugesan
Department of Computer
Science and Engineering
University of Minnesota
200 Union Street,
Minneapolis, Minnesota 55455
anitha@cs.umn.edu

Sanjai Rayadurgam
Department of Computer
Science and Engineering
University of Minnesota
200 Union Street,
Minneapolis, Minnesota 55455
rsanjai@cs.umn.edu

Mats Heimdahl
Department of Computer
Science and Engineering
University of Minnesota
200 Union Street,
Minneapolis, Minnesota 55455
heimdahl@cs.umn.edu

ABSTRACT

Gathering and analyzing Cyber-Physical System (CPS) requirements pose some challenges to the requirements engineering community warranting a fresh perspective on requirement engineering methods; a perspective that is sensitive to the interplay between the cyber and physical aspects of the system. In this paper we share our experiences and lessons learned in the process of formulating requirements for a generic version of an infusion pump, a commonly used piece of medical equipment. Specifically, determining the precise scope of the system and finding its significant attributes in the continuous physical domain in which it operates were surprisingly difficult. To address these challenges, we pursued a model-based approach, which we believe is broadly applicable to CPS requirements elicitation and specification.

Keywords

Requirements; Modeling; Cyber-physical systems

1. INTRODUCTION

Infusion pumps are used to accurately infuse liquids into a patient's bloodstream. Unfortunately, infusion pumps have been involved in many incidents that have resulted in harm to the patient [5]. The US Food and Drug Administration (FDA), through its Infusion Pump Improvement Initiative [3] has sought to pro-actively promote the safe use of these devices by establishing additional requirements for infusion pump manufacturers. In this context, the research community—in collaboration with the FDA—is exploring various methods to improve the safety of infusion pump systems. Development of rigorous assurance cases [7] has been identified as one of the areas that will be explored in the effort to improve safety [4]. Assurance cases are structured arguments, supported by evidence, that provide a compelling, comprehensible and valid case that a system satisfies the given claim in a given environment. The medical device manufactures may submit assurance cases to the FDA as part of the process to gain approval to sell a medical device on the U.S. market. Nevertheless, development and review of these assurance cases have been a challenging task due to poor development practices, poor evidence collection, and poor structure of the case itself [17].

Our aim is to contribute to this FDA initiative by providing an archetype of system development artifacts for a Generic Patient Controlled Analgesia Infusion Pump (GPCA) system; a collection of artifacts that could serve as a generic reference standard used by researchers, practitioners, and certification authorities while developing and reviewing assurance cases. To provide a well argued assurance case, it is absolutely crucial to have a suitable set of system requirements from which to start; to argue that a hazard is not present in a system we must first demonstrate the system requirements do not allow the hazard to occur. Absent such requirements, the assurance case cannot be made. Unfortunately, the publicly available requirements documents for the GPCA system did not meet our standards of completeness, consistency, and rigor needed to serve as a basis for an assurance case. Therefore, before we could start our work on exemplar assurance cases, we needed to focus our efforts on developing a suitable requirements document for the GPCA system.

Initially, given the available information about the GPCA, we anticipated that this effort would be straight forward and completed in short order. During the process, however, we faced unanticipated challenges that we believe are not limited to the GPCA but also to a wide variety of systems in the CPS domain. In particular, we faced two challenges. First, determining the scope of the system under consideration turned out to be unexpectedly difficult. What phenomena in the enviroment should the requirements be expressed over? For example, should the requirements be written in terms of the flow of liquid into the patient? Through the hose? Such issues have a large impact on how an assurance case for the system is constructed and we found preciously little guidance on how to appropriately scope a system. Second, requirements for the continuous and continual physical

[*]This work has been partially supported by NSF grants CNS-0931931 and CNS-1035715.

domain in a cyber-physical system were inadequate in the existing GPCA documentation. For example, there were requirements related to the maximum size of air-bubbles flowing through the hose. Unfortunately, there were no requirements on how to treat a closely spaced sequence of smaller air-bubbles flowing through the hose—requirements on the cumulative volume of air infused over a sliding time-window are needed.

In addressing these challenges we relied on modeling. Models have proved useful during requirements elicitation [18], analysis and validation [15], as well as verification [10] of complex safety related control systems. Similarly to the existing GPCA requirements, in our attempt to reuse existing GPCA behaviorial models [1], we found that the models did not suit our needs in terms of the level of structure, level of detail, ease of understanding, and accommodation of change. The existing models were simply developed with different goals in mind.

In this paper we describe our experiences and lessons learned during the modeling and requirements effort. When developing models to address our requirements challenges, we relied on architectural models to visualize and understand the system boundaries and control models in the continuous domain to help identify the missing requirements on the physical side of our cyber-physical system. (For the modeling activity we choose Simulink [2], one of the most widely used modeling tools.) We begin by giving an overview of the GPCA system in Section 2, so that the reader can follow the examples of requirement statements used through out the paper. In Section 3 and 4 we describe in detail the challenges and modeling approaches used to address them. We then point out some of our next steps related to this effort in Section 5 and finally conclude with a brief summary of our insights in relation to the Infusion Pump Initiative Project as well as the larger CPS research community.

2. GPCA SYSTEM DESCRIPTION

Infusion pumps are medical cyber physical systems used for controlled delivery of liquid drugs into a patient's body according to a physician's *prescription*, a set of instructions that governs the plan of care for that individual. These pumps may be classified into various kinds depending on their features, construction, and usage. Patient-controlled analgesia (PCA) pumps are generally equipped with a feature that allows patients to self-administer a controlled amount of drug, typically a pain medication.

Infusion pumps typically provide multiple modes of drug delivery. In *basal* mode, the drug is delivered at a constant (and usually low) rate for an extended period of time. In a *bolus* mode, the drug is delivered at a higher rate for a short duration of time to address some immediate need or to increase the drug delivery according to some therapy regimen. There may be multiple bolus modes. In *clinician bolus* mode, the drug is delivered at an elevated rate in response to a clinician's request. For example, the clinician may prescribe an elevated rate of infusion for a period of time at the beginning of infusion therapy. Further, in a PCA system, a *patient bolus* mode may be activated to deliver additional drug in response to a patient's request for more medication, typically to alleviate acute pain.

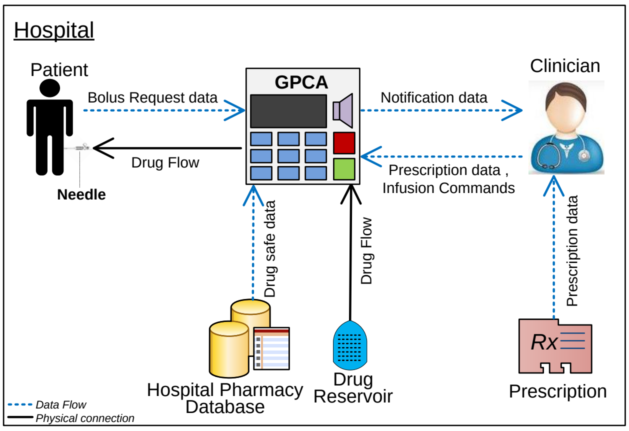

Figure 1: Environment - GPCA System Overview

Figure 1 shows an external intravenous generic Patient controlled analgesia (GPCA) device in a typical usage environment, a hospital or a clinic. We refer to the device along with its environment as *infusion system* and the device in isolation as *GPCA system*. In an infusion system, the clinician operates the GPCA device, programs the prescription information, loads the drug, connects the device with the patient, and responds to exceptional conditions that occur during the therapy. The patient receives the medication from the device through an intravenous needle. The patient can self administer prescribed amounts of additional drug by requesting a bolus, a request usually done by pressing a bolus request button accessible at the patient's bed. The hospital pharmacy database is a repository that stores manufacturer provided drug information. This typically includes ranges of values for various drug parameters that are safe for use in infusion therapy. The GPCA system has an interface to this repository for accessing this drug data that it used for verifying various infusion parameter values against the drug specific data from the repository to ensure that the programmed therapy regimen is within safe limits.

The GPCA system has three primary functions (1) deliver the drug based on the prescribed schedule and patient requests, (2) prevent hazards that may arise during its usage, and (3) monitor and notify the clinician of any exceptional conditions encountered.

3. DETERMINING SYSTEM SCOPE

Through assurance cases, one seeks to establish claims about various system attributes such as safety, dependability and security, by providing a systematic argument backed with evidence that those claims are true. A claim is always *contextual*—construction of an assurance case as well as its scrutiny are necessarily dependent on the system context in which the claim is made. The context clarifies what may be assumed of the environment and what must be assured by the system. Precise demarcation of this boundary between a system and its environment—determining scope—is a classic requirements engineering challenge that has a profound effect on problem analysis [6]. Further, insufficient analysis among and between different engineering disciplines in understanding the scope of the system and its components is known to be a significant cause of CPS failures [8, 13].

This is particularly a challenge for CPS systems that are typically expected to have high dependability and reliability in an uncertain physical world [9]; Especially, medical CPS are expected to be safe and reliable even in changing environments and unforeseen conditions [14], hence precisely scoping the environment is crucial to assure safety.

3.1 Scope Challenge

In the GPCA project, precisely scoping the system and its environment was far more challenging than we first anticipated. This is partly because the existing documents that stated requirements over the GPCA system had requirements expressed over multiple scopes and there was little guidance to determine which scope would be appropriate. For example, consider the following statements from the existing requirements sources:

> *"An air-in-line alarm shall be triggered if air bubbles larger than 5 ml are infused into the patient."*
> *"When the option to suspend the pump is selected, the current pump stroke shall be completed prior to suspending the pump."*

The above statements are clearly requirements about the GPCA system, but they are expressed over different scopes. The first statement is concerned with the the entire infusion system interacting directly with the patient (Figure 1), disregarding the internal components of the system. The second statement is about a specific component (the physical pump) within the GPCA system in response to an event. These two statements should not be parts of the same requirements document since they are expressing constraints over two very different scopes.

Providing a consistent scope for the requirements is essential to maintain intellectual control of the the development and assurance efforts. We do not want our efforts to get bogged down in implementation details, such as how to operate a particular pump, too early during development. On the other hand, we do not want the scope of the system to grow as development progresses, for example, by including the clinician as part of our GPCA system.

Previous efforts in requirements engineering—most notably the requirements reference models developed by Parnas and Madey [12] and Gunter et al. [6]—provide guidance for maintaining a consistent scope in the requirements efforts. They do this by delineating how to define requirements and environmental constraints over the quantities (phenomenon or variables) in the environment, in the system, and in the interface between the two. Unfortunately, these models do not provide much guidance regarding how to identify this interface between the outside (the environment) and the inside (the system) and what lies at the interface between the two. In other words, there is little guidance for determining the scope of the system.

If we provide requirements with a large scope, the requirements will be closer to the safety claims we need to establish in our assurance case. This will make it easier to establish the beginnings of the assurance case, since few assumptions about the surrounding system needs to be made to establish the case. For example, if we write infusion pump requirements in terms of the *prescribed dosage* of a drug we can at this scope postpone discussion of entry errors (the clinician entering the prescribed dosage into the pump is part of our system). On the other hand, from a development perspective, is is most likely preferable to write infusion pump requirements in terms of the *dosage entered at the pump interface*; the clinician is not part of the pump system and entry errors are part of its environment. Thus, the choice of scope does not only have implications for our development efforts (what is "in" our system and what is "outside" our system), but also has profound implications for the way assurance cases are constructed. To assist us in the process of scoping our system and establishing rigorous interfaces between the system (what is "in") and the environment (what is "out") we relied on architectural modeling. Although this modeling cannot help us definitely determine the scope of our system, the clarity provided through modeling certainly makes the effort more tractable.

3.2 Architectural Models

The scope of a system—the interface between the system to be developed and the environment in which it is to reside—cannot be full defined until the early stages of a solution to the engineering task has been established. The rough structure of the system (the solution to our problem) must be defined. For example, we need to establish what will be sensed and actuated, and chose candidate control strategies—we need to establish an early *architecture* of the system.

The requirements and the architecture are interrelated [11] and share a symbiotic relationship. Requirements and the solution architecture co-evolve in the process of identifying the appropriate architecture as well as discovering the requirements [16]. Requirements define the constraints on an architecture and identifying an architecture helps in discovering the precise requirements and environmental assumptions. This evolutionary process helps scope the system.

In the GPCA project, we defined the top level as illustrated by the model[1] in the first half of Figure 2 (labeled "Infusion System"). This infusion system architecture is consistent with the conceptual view of the overall system (Figure 1). In this view, the GPCA system, clinician, patient, etc. are all components of the overall infusion system. The requirements of the GPCA system were all stated only using the input and output variables for this subsystem of the overall infusion system. The infusion system architecture was prepared using publicly available user manuals and previous research efforts [1].

The choice of the scope for a requirements effort is of course debatable, we do not attempt to suggest an optimal scope. Nevertheless, we would like to emphasize that whatever the scope is chosen to be, it must be well defined and all requirements and environmental assumptions should be stated based on that scope.

[1] Comprehensive GPCA system architectural models are available at http://crisys.cs.umn.edu/gpca.shtml

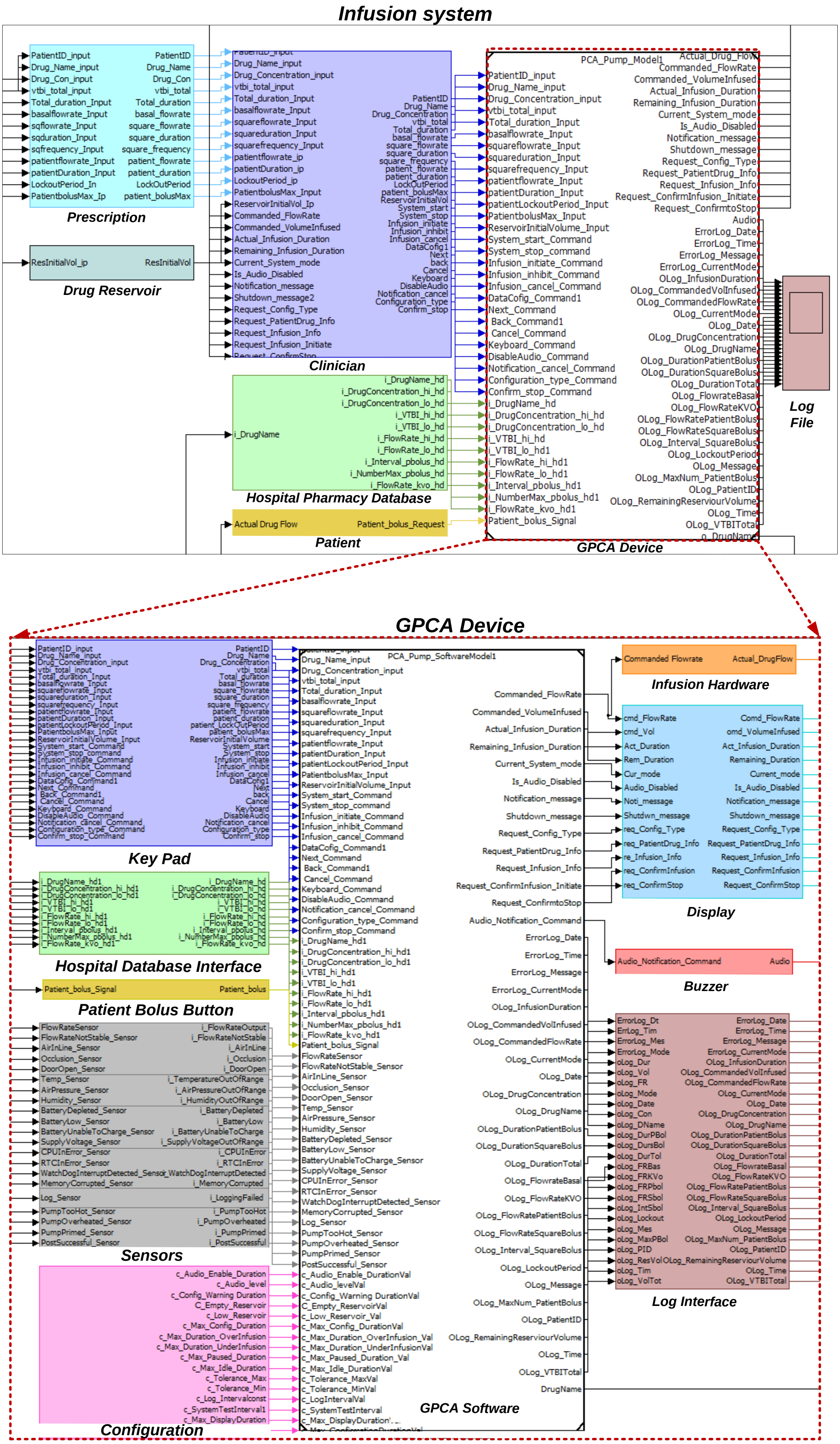

Figure 2: Architectural Model

Note here that scoping the system is not a one time task. As the system development process progresses, modeling in finer granularity helps identify the way the individual components within the system working together will meet the system requirements [16].

In the GPCA project, to visualize scoping at various levels, we modeled the the decomposed view of the GPCA device as illustrated by the model in the second half of Figure 2 (labeled "GPCA Device"). In this decomposed view the sensors, data entry interface, software, etc. are components of the GPCA system. This kind of model helped visualizing the decomposition of the system and helped identify the scope of the software requirements.

4. REQUIREMENTS IN THE CONTINUOUS DOMAIN

In a cyber-physical system, the physical portion of the system resides in the continuous and continual domain. Thus, on the physical side of cyber-physical systems we will have to contend with not only real-time requirements but also requirements related to the continuous and continual nature of the system.

This poses a new set of challenges for requirements engineering; we must write well defined requirements to address crucial issues not commonly addressed in the software requirements domain. For example, the rate of change of a controlled variable, the time it takes for a controlled variable to settle sufficiently close to a set-point, and the cumulative errors built up over time may be of critical importance.

4.1 Continuous Requirements Challenge

In the GPCA project, identifying all the requirements that must be stated to address the inherent complexity in the continuous domain was a challenge. Consider, for example, the requirement below.

> *"A patient bolus dose shall be given when requested by the patient."*

The above requirement looks complete and quite simple—a patient bolus dose is simply a higher flow-rate of the drug for a specified period of time. This simplicity is quite deceiving, however. This requirement does not address crucial aspects of the continuous and continual domain such as the acceptable rate at which flow is increased from the basal rate to the bolus rate, the time allowed to reach the bolus rate, etc. Such requirements are—in our experience—not specified leaving such decisions up to the designers of the system. In some systems, omitting requirements on rate of change, rise time, settling time, etc. might be acceptable. In other systems, these requirements are crucial for the system's success. For example, in an automotive cruise control application, acceleration and possible oscillation in the system are highly influential for passenger comfort—here, requirements defining these aspects of the system are absolutely essential.

As a second example, consider the GPCA safety requirement below.

> *"An air-in-line alarm shall be triggered if air bubbles larger than 5 ml are infused into the patient."*

This requirement concerns the response to a single air-bubble of a certain volume. Nevertheless, a closer examination of the problem of air-bubbles quickly reveals that, for example, 10 closely spaced consecutive air bubbles each of a volume of 4 ml might be infused without an alarm. Again, this is an example of where the continuous and continual nature of the physical side of a cyber physical system must be carefully considered during requirements elicitation. Unfortunately, in our work we have found little guidance on how to systematically approach the requirements elicitation process in cyber-physical systems. Similarly to the requirements and architecture synergy discussed in the previous section, we have found behavioral modeling in the physical domain extremely useful in the elicitation, discovery, and refinement of the requirements for cyber-physical systems.

4.2 Control System Modeling Approach

In order to address the CPS requirements challenge, a different category of models that help clarify the control behavior of the system in the continuous domain are needed. These are traditional control models including the controller, sensor and actuator models, as well as a plant model. These models are traditionally used to evaluate various control strategies, tune the controllers, etc. In our case, however, we developed the models to better understand the requirements needed to adequately constrain the desired system behavior. Figure 3 is a snapshot of the plant model[2] we created for the infusion system.

Through these modeling efforts one has an opportunity to explore various system responses and investigate how a proposed system might behave in its intended environment. For example, as mentioned in Section 4.1, in an automotive cruise control, requirements governing response slope, rise time, overshoot, settling time, etc. are absolutely crucial. In our GPCA system on the other hand, the system dynamics do not allow for any overshoot (there is no inertia to speak of in the pump) and settling time is negligible for the same reasons. Thus, one can argue that requirements governing these quantities are superfluous for the GPCA. Without modeling in the continuous domain, discovering which requirements are needed and which ones are superfluous (and the reasons why they are superfluous) will, in our opinion, simply not be possible.

Similarly, the system's response to cumulative effects of certain events and conditions, for example, deviations from the set-points, were not adequately discussed in the existing requirement. For example, consider the requirement below.

> *"If the flow-rate exceeds the programmed rate setting by more than 10% over a period of more than 10 minutes... issue an alarm to indicate over infusion".*

[2] Comprehensive infusion control system model is available at http://crisys.cs.umn.edu/gpca.shtml

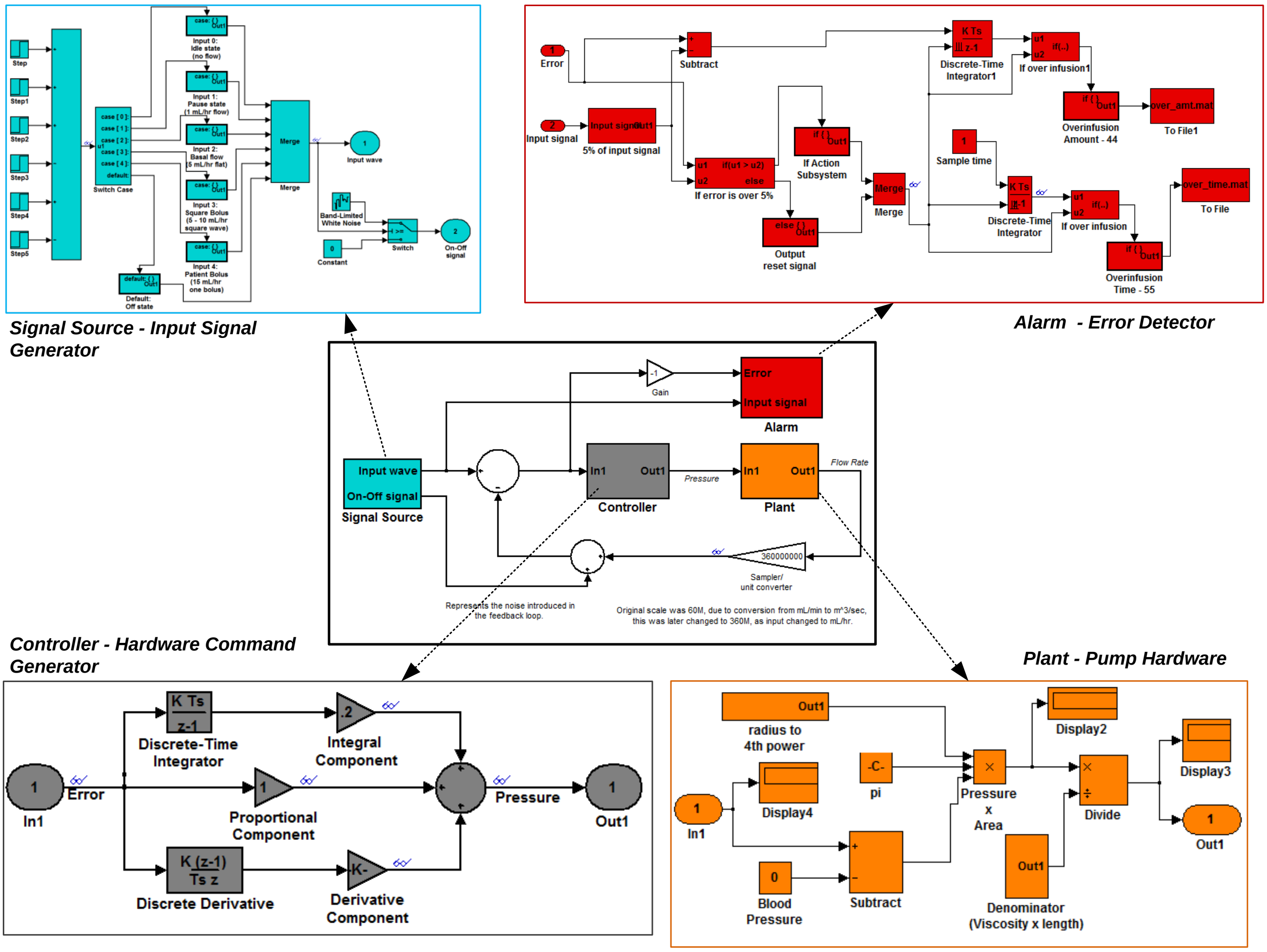

Figure 3: Expanded Control System Model

This requirement does not define if the *10 minutes* is the cumulative time over the entire infusion duration or a continuous 10 minute interval somewhere during the infusion duration. Naturally, this is a crucial issue to address with the domain expert since 10 cumulative minutes is a far more conservative requirement than 10 consecutive minutes. Nevertheless, such issues are easily overlooked until modeling efforts force a decision when modeling rasing the over-infusion alarm.

5. NEXT STEPS

The modeling steps we did to address our challenges seem promising. Nevertheless, since some parts of the work was trial and error, we are currently working on identifying a systematic method to perform these activities that can serve as a guidance for other developers of cyber-physical systems facing similar issues. In particular, we are developing guidelines for scoping a system during the requirements effort sa well as guidelines for the flow-down of requirements to the system components identified during architectural modeling. In addition, we are developing a catalogue of requirements patterns describing the various requirements needed to adequately describe the system dynamics in a continuous and continual cyber-physical systems.

6. CONCLUSION

In the process of documenting requirements for a GPCA system we came across challenges related to (1) scoping the system for the requirements effort and (2) finding and defining requirements for the continuous and continual aspects of a cyber-physical system. To address these problems we adopted an architectural modeling approach that helped us identify system interfaces. These well defined interfaces subsequently assisted in determining what was part of our system under development and what was part of the surrounding environment. A related modeling effort focusing on the desired system behavior in the continuous domain (as opposed to the structural aspects of the architectural models) helped in the identification of the requirements needed to adequately constrain the desired system behaviors to ensure predictable and safe operation.

Although our approach to date has been somewhat ad-hoc, it has been quite successful and we are in the process of developing a requirements engineering method suitable for cyber-physical systems, a requirements reference model supporting the hierarchical decomposition and flow-down of requirements in a complex system, and a collection of requirements patters to help capture the subtle requirements needed to adequately constrain the continuous and continual aspects of a cyber-physical system.

7. REFERENCES

[1] Generic infusion pump project, http://rtg.cis.upenn.edu/gip.php3.

[2] Simulink, http://www.mathworks.com/products/simulink/.

[3] U. Food and D. Administration. Infusion Pumps, http://www.fda.gov/infusionpumps.

[4] U. Food and D. Administration. Guidance for Industry and FDA staff - Total Product Life Cycle: Infusion Pump - Premarket Notification [510(k)] Submissions. April 2010.

[5] U. Food and D. Administration. White Paper:Infusion Pump Improvement Initiative. April 2010.

[6] C. A. Gunter, E. L. Gunter, M. Jackson, and P. Zave. A reference model for requirements and specifications. *IEEE Software*, 17(3):37–43, May/June 2000.

[7] T. Kelly. Reviewing assurance arguments-a step-by-step approach. In *Proceedings of Workshop on Assurance Cases for Security*, 2007.

[8] J. Knight. Safety critical systems: challenges and directions. In *Proceedings of the 24th International Conference on Software Engineering*, pages 547–550. IEEE, 2002.

[9] E. Lee. Cyber physical systems: Design challenges. In *11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369. IEEE, 2008.

[10] S. P. Miller, M. P. Heimdahl, and A. Tribble. Proving the shalls. In *Proceedings of FM 2003: the 12th International FME Symposium*, September 2003.

[11] B. Nuseibeh. Weaving together requirements and architectures. *Computer*, 34:115–117, 2001.

[12] D. Parnas and J. Madey. Functional documentation for computer systems engineering. *Science of Computer Programming*, 25(1):41–61, 1991.

[13] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang. Cyber-physical systems: A new frontier. *Machine Learning in Cyber Trust*, pages 3–13, 2009.

[14] N. H. C. Software and S. C. Group. High-confidence medical devices : Cyber-physical systems for 21st century health care. technical report. http://www.nitrd.gov/about/meddevice-final1-web.pdf.

[15] P. Tate. Model based requirements elicitation. In *3rd IET International Conference on System Safety*, pages 1–5. IET, 2008.

[16] M. Whalen, A. Murugesan, and M. Heimdahl. Your what is my how: Why requirements and architectural design should be iterative. In *First International Workshop on the Twin Peaks of Requirements and Architecture*, 2012.

[17] S. Wilson, T. P. Kelly, and J. A. McDermid. Safety case development: Current practice, future prospects. In *Proceedings 1st ENCRESS/12th Annual CSR Workshop*, 1995.

[18] E. Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *Proceedings of the Third IEEE International Symposium on Requirements Engineering*, pages 226–235. IEEE, 1997.