Generic Patient Controlled Analgesic Infusion Software Model Description

Version 0.1 [draft]

Author
Anitha Murugesan

Reviewers
Dr.Mats Heimdahl

University of Minnesota, Twin Cities

Updated on : February 28, 2013
Chapter 1

Introduction

This document contains the documentation required to understand the software model for infusion pump. Since the particular pump model described in this document is not based on any existing model currently on the market, we call the model described in this document the Generic Patient Controlled Analgesia Pump (GPCA) software model.

1.1 Document Purpose

This document shall serve as a "readme" file for the software model of the GPCA system. The scope of this document is limited to the behavioral requirements and constraints of the GPCA software model.

1.2 Conventions

To be both brief and precise, the document contains many abbreviations, acronyms and symbols; all abbreviations, acronyms and symbols are detailed in the glossary in Appendix 4.

The model is created using SIMULINK Stateflow tool, hence this document contains screen shots of the model and a brief description of the modules. Also while describing the modules, some terms that are tool specific are used, for example 'Execution Order' is a term used for specifying the order in which the state machine should be executed. A brief tutorial of the tool is provided in the appendix.

1.3 Intended Audience

The intendance audience for this document are researchers and practitioners that want to study the model in a format and style the authors of this document find appealing and appropriate. We are fully aware that others may disagree with our organization, style, and content; we welcome feedback and spirited discussion. We request that feedback on this document be directed to Mats Heimdahl at heimdahl@cs.umn.edu. It is strongly suggested that the reader first reads the requirements document prior to reading this document. It is also recommended that the reader is aware of the stateflow tool in order to understand the model.

1.4 Scope

The scope of this document is to describe the high level details of the GPCA software model. It doesn’t explain all the details of the model. The reader is expected to use this document as a guide in understanding the overall design of the model. This document provides some justification for certain design decisions wherever necessary.
Chapter 2

Overview

This chapter gives an overview of the system, providing a brief description of the components of the system and the system’s high level functions. Note that the descriptions in this chapter are intended as an overview only; the requirements of the system are defined in a separate requirements document.

2.1 System Overview

The discrete behavior of complex control systems is usually modeled in terms of (extended) finite state machines, which are a formal, mathematical representation that support sophisticated verification techniques. As part of a larger project to investigate techniques for assuring safety and efficacy of medical cyber-physical systems, we modeled an infusion pump, called a Generic Patient Controlled Analgesia (GPCA) infusion system to better understand its behavior and to analyze its properties. Essential to this model was the mode logic of the infusion system. For our work we used the Simulink/Stateflow modeling tools that provide a visual formalism for describing states and transitions in a modular and hierarchical fashion. The notations are also supported by a rich ecosystem of various analysis, translation, and verification tools. Since this modeling effort was carried out in the early stages of exploration of the system specification, one of the goals was to ease the burden of making changes to the model as our understanding of the system evolved. This, in particular, required a careful design so that additions, modifications, and removal of behaviors could be done in a quick and localized fashion without losing model integrity.

Figure 2.1: GPCA Infusion system.
GPCA Infusion system is a cyber physical system that controls the rate of infusion of analgesia and enables the patient to self-administer their own analgesic according to the programmed dosage of 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. The GPCA devices are usually built with capability to notify the clinician of exceptional conditions, for example, if the drug reservoir is running low or if there are air bubbles in the system.

The GPCA in its intended environment is illustrated in Figure 2.1. 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.

### 2.2 GPCA device Overview

![GPCA architectural components](image)
The architecture is a GPCA device illustrated in figure 2.2. The architecture was prepared using publicly available user manuals and previous research efforts. The software is one of the components of this architecture. This architecture was fixed in order to precisely define the inputs, outputs and functionality of the software.

2.3 GPCA Software Overview

The primary reason for modeling the GPCA software state machine was to understand and verify the requirements. Hence we incorporated certain patterns in our design that would make the model conceptually clean, easy to understand and maintain and be amenable to formal analysis. First of all, overall organization of the model to conceptually correlate the requirements was essential for easy understanding and maintainability of the model. In our model we incorporate the orthogonal concepts discussed in requirements as parallel...
state machines, which typically mimics the way the system is expected to work in reality.

Secondly, the priority of the parallel state machines was carefully planned by considering the dependency of the state machine with others. For example, the execution order of the alarm is higher than the infusion manager: So when an air-inline alarm is raised, the system will stop infusion in the same step the alarm is raised. If the execution order was different, then the system will stop infusion in the next step.

The major portion of our efforts was modeling mode logic in the system. While designing the mode logic, we found that certain structuring patterns allow adding new modes, transitions, maintenance of those modes as well as adding complex mode behaviors makes it easier than their alternates. In this model, a parallel structuring was chosen over sequential structuring for mode logic.

The state flow models also helped identifying unanticipated scenarios resulting in certain behaviors that needed requirements clarification. For example, in the GPCA software, there was a requirement to be able to change the prescription configuration when infusion is in progress. However the behavior of the system in response to a new configuration when a patient bolus is in progress, needed clarification from the requirements perspective. Since if a new configuration is entered when a patient bolus is in progress, the new configuration that clears the history of previous infusion allows another patient bolus, which violates a safety constraint that two patient boluses should be spaced at specific interval. This scenarios wasn’t considered while writing requirements. However, the FSM helped in identifying this behavior.

Please refer to our paper [1] for a detailed explanation of the mode logic modeling patterns. The state flow model of the GPCA software is illustrated in figure 2.3.

2.3.1 Inputs and Outputs

The various inputs and outputs of the software were determined from the GPCA architecture illustrated in figure 2.2. In the model all the inputs and outputs are either boolean or integers. Refer to the requirements document for further details on each input or output.

2.3.1.1 Inputs

1. DATA ENTRY Inputs - Data and commands entered by the user of the system.

2. HOSPITAL DATABASE Inputs- Drug safe value data for the drug that is programmed by the data entry interface. This data is present in the hospital database and is used by the GPCA system for validating the prescription data and ensuring system never infuses beyond these values.

3. BOLUS REQUEST Inputs - This represents the bolus request by the patient.

4. SENSORS Inputs - These are inputs from various sensors of the system listed below.

   - INFUSION SENSORS - Data from the various sensors that monitor the infusion of drug and provide feedback to the software. This data is used for monitoring alarming conditions when system is in use. These include inputs from- Flow rate sensor, AirInLine sensors, Occlusion sensors and reservoir Door sensor.

   - ENVIRONMENTAL SENSORS - Data from the sensors that monitor the environment where the system is in use. This data is used for monitoring exceptional environmental conditions. These include inputs from - Temperature sensor, Air pressure sensor and Humidity sensor.

   - POWER UNIT SENSORS - Data from the sensors that monitor the power unit of the system- such as input AC power and battery. This data is used for monitoring exceptional conditions that occur in the power unit when the system is in use These include inputs from - AC voltage sensor and Battery charge sensors.

   - PROCESSOR SENSORS - Data from the sensors that monitor the processor hardware of the system-. This data is used for monitoring exceptional conditions that occur in the processor hardware when the system is in use These include inputs from -CPU monitors, Real Time clock Monitors, Memory monitors and watch dog error monitors.
2.3. **GPCA SOFTWARE OVERVIEW**

- **LOG SENSORS** - Data from the sensors that monitor the logs of the system. This data is used for monitoring exceptional conditions that occur in the system logs when the system is in use. The inputs indicate if there is an issue in the logs.

- **PUMP SENSORS** - Data from the sensors that monitor the pump hardware. This data is used for monitoring exceptional conditions that occur in the pump hardware when the system is in use. The inputs indicate if the pump temperature is more than the acceptable value, if pump is primed and if the POST is successful.

5. **CONSTANTS** - Inputs - These constant values that are configured for the system such as maximum duration for programming etc. These are not variable when the system is in use.

### 2.3.1.2 Output

1. **INFUSION HARDWARE Output** - This interface receives the commanded flow rate from the system and infuses drug as per the commanded flow rate. The scope of this model is only to output the commanded flow rate to the infusion hardware. The logic of the infusion hardware delivering drug is out of scope of this model.

2. **DISPLAY INTERFACE Output** - The display interface is used for displaying output data of the system such as - current mode of the system, commanded flow rate, duration of infusion, volume infused, alarm messages, status messages and requests for user inputs.

3. **AUDIO INTERFACE Output** - The system provides the audio request when certain alarm conditions occur. The audio interface is used for producing audio output when the command from the system is received.

4. **LOG INTERFACE Output** - The system output two different kinds of log data.
   - Error log - Used to record error messages from the system to the logging subsystem. The error log is produced when the system has alarming conditions and active notifications by the Alarm component.
   - Operational log - Used to record normal operations and current system state of at various times and conditions such as - system transitions, configuration changes, etc.

5. **HOSPITAL DATABASE INTERFACE Output** - During prescription configuration, when the user enters the drug name, the system sends this value to the hospital database interface to get values for the drug safe values for validation. The logic of retrieving drug safe values from the hospital database is out of scope of this model.
Chapter 3

Software Model

The system is grouped into three major state machines as illustrated in the figure 3.1

1. OFF - The default state. The system is considered shutdown when it is in OFF state hence it doesn’t perform any operation.

2. STARTUP-SHUTDOWN - When the system is started, the system enters this state machine. It performs POST checks and authenticates the user during startup [Detailed implementation out of scope]. Prior to shutting down from ON, the systems enters this state machine, where it performs shutdown operations [Detailed implementation out of scope] and displays appropriate message and then transitions to OFF.

3. ON - Upon successful authentication of the user at startup, the system enters the ON state machine - the operational state of the system.

Figure 3.1: GPCA Software State Machine Overview.

3.1 ON subsystem

The ON state machine has six subsystems that run in parallel and is prioritized as necessary by specifying an execution order.
**Alarm** Responsible for Monitoring and notifying defined exceptional conditions.

**Configuration** Responsible for configuring prescription parameters.

**Infusion Manager** Handles drug infusion commands and parameters.

**System Monitor** Performs self tests at regular intervals of time.

**Confirm Stop** Handles the display and confirmation when trying to stop the system.

**Log** Handles data logging.

### 3.1.1 ALARMS

The alarm subsystem has the highest priority over all subsystems. Its main functionality is to monitor for exceptional conditions based on inputs and other internal system values. When exceptional conditions occur, the subsystem categorizes and prioritizes the errors and raise flags that are used by other subsystems to respond. The subsystem is also responsible for display and audio notification of the error. Additionally the alarm component also responds to cancel notification input from user, by canceling the current alarm condition and respective flags raised. The alarm subsystem has the two state machines that are executed in parallel, one for checking alarms and the other for notifications.

#### 3.1.1.1 CHECKALARM

The CheckAlarm state machine, illustrated in figure 3.2, is responsible for (1) canceling the current alarm based on input from user, (2) checking for Level 4, 3, 2 and 1 alarm conditions and (3) setting the current alarm and highest level alarm that is currently active. Multiple alarm levels could be active at the same time, but the system always takes the current highest level alarm actions and notifies the same.

![CheckAlarm state machine](image)

**Figure 3.2: CheckAlarm state machine**
The exceptional conditions in the system are categorized into four levels, depending upon their severity that decides that action to be taken. Note that all alarms in the same level have a common behavioral response except for their message display.

1. **Level 4** - The highest level alarm - These require stopping all infusion completely and transition to IDLE state.
2. **Level 3** - These alarms require stopping normal infusion and infuse at KVO rate.
3. **Level 2** - These alarms require inhibiting all bolus infusions. However, they allow basal infusion.
4. **Level 1** - These are warning messages that are informational. These don’t affect infusion.

### 3.1.2 NOTIFICATION

The notification subsystem depends on the current active alarm and highest level alarm values set in the CheckAlarm state machine. The notification state machine has two sub-state machines, visual and audio. The visual notification state machine produces output to display the message for the current active alarm. 

*Note: Model currently uses numeric identifiers to represent the message for a particular alarm condition. Implementation of exact message display is not in scope of this model.*

The audio notification state machine, illustrated in figure 3.3 produces audio output based on the highest level alarm. The audio notification can be disabled permanently or silenced temporarily based on user inputs. When its disabled or silenced, no audio output is produced. When it is silenced, its resumes to normal operation after a preset time interval. However, when it is disabled, it shall resume to normal operation only if there is a user action to remove the disable input.

![Figure 3.3: Audio notification state machine](image)

### 3.1.3 CONFIGURATION

The configuration subsystem, illustrated in figure 3.4, is responsible for configuring the prescription data for the infusion. This is invoked either (1) user requests configuration or (2) automatically when there is infusion initiation request and there is no prescription configured.
Once the configuration is initiated, it should be completed by a preset interval of time, otherwise the configuration in progress shall be canceled and the system transitions to the main configuration screen.

The user can either edit an existing prescription or add a completely new set of prescription. Configuration can be initiated when the system either currently infusing or not. When the system is infusing the drug, the user shall not be able to configure the patient or drug information; Only the infusion parameters can be changed. If the system is not infusing, all the parameters can be modified or added.

When the parameters are entered by user, the system validates the data using the drug safe values retrieved from the hospital database. The system doesn’t allow the user to proceed if any data validation fails. The user needs to correct the values and revalidate successfully before proceeding. After all validations are successfully validated, the system requests the user to confirm initiation of infusion with the newly entered values. After the confirmation, the system saves the modified or newly added prescription values and initiates infusion with the recent prescription values. The existing prescription values are over written by the new values only after confirmation and the infusion starts or continues with the newly configured values.

### 3.1.4 INFUSION MANAGER

The infusion manager subsystem is responsible for commanding the amount of drug delivery when system is in use. The Infusion manager has hierarchy of states. The first level has IDLE and THERAPY states, as illustrated in figure 3.5.

On successful authentication during system start-up, the system enters the IDLE state. There is no drug flow during IDLE state. Typically configurations of prescriptions are done when the system is in IDLE state.

The THERAPY state machine is the drug infusing state. It has multiple sub-states, that correspond to various modes of infusion of the system. The system transitions from IDLE to THERAPY when infusion is initiated and the system currently doesn’t have any Level4 Alarms. The system exits from THERAPY state to IDLE state when after infusion completion or when there is a Level4 alarm.

#### 3.1.4.1 THERAPY SUBSYSTEM

The therapy state machines has two sub-state machines.

**PAUSED** - Handles system response when it is paused due to alarm or pause action from user.

**ACTIVE** - Handles active drug infusion commands.
3.1. ON SUBSYSTEM

Figure 3.5: Infusion manager state machine

Design Notes: In PAUSED and ACTIVE, the individual infusion deciding sub-states are designed as parallel state machines, with individual ON and OFF states. Multiple sub-states could be ON at a current time depending upon the inputs and various system values. In order to arbitrate the common parameters (flow rate) among the sub-states such that only one value of the common parameter is outputted, another state machine, named Arbiter, decides the current mode and output of the system based on the priority of the states.

Figure 3.6: ACTIVE state machine

The PAUSED state machine has two sub-states that have different flow-rate behaviors based on their source - whether they are paused due to alarm conditions or paused due to user action.

The ACTIVE state machine has three sub-states:

- **BASAL** - Responsible for setting flow rate of the drug to basal flow rate. Maximum duration of this delivery is the prescribed total duration.

- **INTERMITTENT bolus** - Responsible for setting flow rate of the drug to intermittent bolus flow rate at prescribed intervals for a prescribed duration during prescribed total duration.

- **PATIENT bolus** - Responsible for setting flow rate of the drug to patient bolus flow rate for prescribed duration when patient bolus is requested by the patient. Once Patient bolus is initiated, when it ends or interrupted by an alarm - it enters a lockout period during which another patient bolus cannot be
delivered. The lockout state is maintained between configurations to avoid delivering two patient bolus close to each other.

3.1.5 SYSTEM MONITOR

The system monitor is a monitoring subsystem that performs some self checks at regular intervals when system is ON. These are typically tests that ensure system is in order and raises alarm if there is a problem detected. [Detailed implementation of system monitors are out of scope].

3.1.6 CONFIRM STOP

This subsystem is invoked when there is a system stop command from the user. The system displays a message which requests a additional confirmation before the system exists ON state machine. The system continues to infuse and checks alarm, until the user confirms the stop action. If there is no confirmation from the user within a predefined amount of time, the system exists this subsystem without any further action. This was done to prevent hazards due to accidental stop command and prolonged wait for stop confirmation.

3.1.7 LOG

The Log subsystem is responsible for recording all major actions in the system. It has two types of logging capability.

Error log - records details of all alarms raised by the system.

Operational log - As illustrated in 3.7, it records major system operations such as transitions, prescription change, infusion completion and system status every predefined intervals of time. This is typically used for auditing the system operations. The scope of the log subsystem is to demonstrate the various logging capabilities and not list all the details of logging.

Figure 3.7: Log state machine
Chapter 4

Appendix

4.1 Acronyms

G 1 GPCA : Generic Patient Controlled Analgesic

G 2 KVO : Keep vein open

4.2 Definitions

G 3 Basal Delivery : Prescribed Infusion therapy characterized by a constant fixed-rate dose

G 4 Bolus Delivery : Prescribed additional amount of medication given to patient for a short period of time.

G 5 Drug Safe Limits : The actual drug safe usage information provided by the drug manufactures. This will typically provide the maximum and minimum values for the drug usage. For patient safety, the dosage of the drug to the patient should never exceed these values.