Toward the model-based development of a GPCA reference implementation

Insup Lee
PRECISE Center
School of Engineering and Applied Science
University of Pennsylvania

SCC meeting, Annapolis
May 2, 2011

Penn Engineering PRECISE

Collaborators

- PRECISE members
  - Oleg Sokolsky
  - BaekGyu Kim
  - Anaheed Zaki
  - Dave Arney (CIMIT)
- FDA
  - Paul Jones
  - Raoul Jetley
  - Yi Zhang

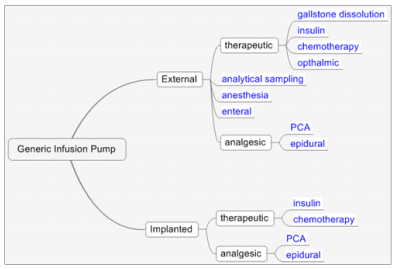
Infusion Pump Safety

- From 2005 through 2009, FDA received approximately 56,000 reports of adverse events associated with the use of infusion pumps, including serious injuries and deaths [1].
  - During this period, 87 infusion pump recalls were conducted by firms to address identified safety problems.
- The most common types of problems
  - Software Defect
  - User Interface Issues
  - Mechanical or Electrical Failure

[1] U.S. Food and Drug Administration, Center for Devices and Radiological Health. White Paper: Infusion Pump Improvement Initiative, April 2010.

Generic Infusion Pump (GIP) Project

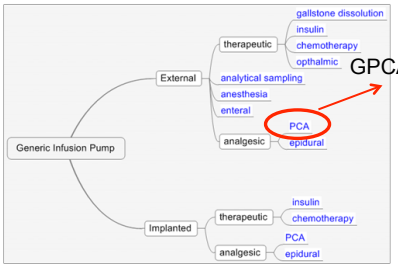
- The Goal of GIP Project
  - To develop a set of generic infusion pump (safety) models and reference specification that can be used as a reference standard to verify safety properties in different classes of infusion pumps
- GIP web site
  - provide a repository of medical device artifacts for use in projects that advance the science and practice of developing high-confidence medical devices, software, and systems, and
  - establish infusion pump safety reference models



GIP Class Diagram

Generic PCA (GPCA)

- Generic PCA (Patient Controlled Analgesic) Infusion pump
  - GPCA hazard analysis
  - GPCA safety requirements
  - GPCA reference model
- Motivation
  - Demonstrate the use of model-based development techniques for engineering medical device software
  - Provide a base open-source reference model that can be extended and modified to develop specific implementations of PCA pump software
  - Provide a reasonably complex medical design for researchers to use in developing, refining, and improving theories and methods needed to develop certifiably dependable medical devices



GIP Class Diagram

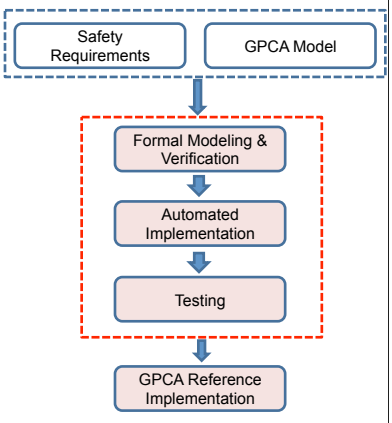
PCA Pump

Penn Engineering

PRECISE

GPCA reference implementation

- Given
  - GPCA Safety Requirements
  - GPCA Model (Simulink/ Stateflow)
- Develop a GPCA reference implementation
  - Provide evidence that the implementation satisfies the safety requirements



Model-Based Development of GPCA Reference Implementation

Penn Engineering

6

PRECISE

GPCA Hazard Analysis

- System Domain
  - Infusion pump, drug infusion set, networks, patient, environment, user/medic
- Pump Components
  - Infusion (pump) module, user-interface module, error handling module, power module, does error reduction module, communication module
- Hazards
  - Operational, Environmental, Electrical, Hardware, Software, Mechanical, Biological and Chemical, Use
  - For each hazard
    - Pump type, Cause, action, mitigation, safety requirement
  - Example Hazard : Overinfusion
    - Cause : Dose limit exceeded due to too many bolus requests.
    - Action : Alarm(), Log()
    - Mitigated by : Flow sensors.
  - Example Hazard : Underinfusion
    - Cause : Reservoir empty
    - Action : Alarm(), Log()
    - Mitigated by : Flow sensors and Drug library.

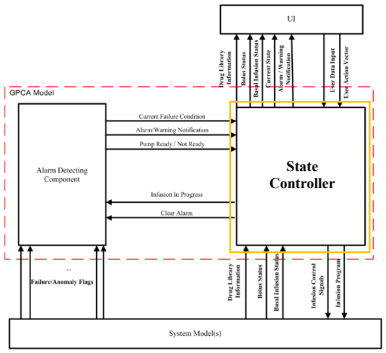
GPCA Safety Requirements

- A set of requirements that should be guaranteed in general PCA pumps for patient safety.
  - A minimum set of generic safety requirements that can be used to evaluate and verify software implementations for specific infusion pumps. [3]
    - No normal bolus doses should be administered when the pump is alarming (in an error state).
  - Developed based on the hazard analysis (e.g., underinfusion caused by empty reservoir.)
  - Contains symbolic parameters so that requirements can be instantiated on a range of PCA pump systems
    - If the calculated volume of the reservoir is *y ml*, and an infusion is in progress, an Empty Reservoir alarm shall be issued.
  - Informal

[3] Raoul Jetley and Paul Jones. Safety Requirements based Analysis of Infusion Pump Software. Proceedings of the Workshop on Software and Systems for Medical Devices and Services, December 2007.

GPCA Model

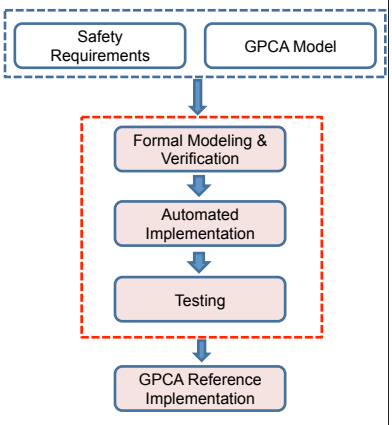
- An abstract representation of software used in a typical PCA infusion pump.
- The model is built in Simulink and Stateflow.
- State Controller
  - Describes a drug administration process such as parameter setting and bolus request.
- Alarm Detecting Component
  - Check hardware conditions and process alarm on any hardware failure.
- GPCA Environment
  - User Interface
  - System model
    - The GPCA model interacts with pump hardware such as motor and sensors through the System Model.



The System Architecture of GPCA Model

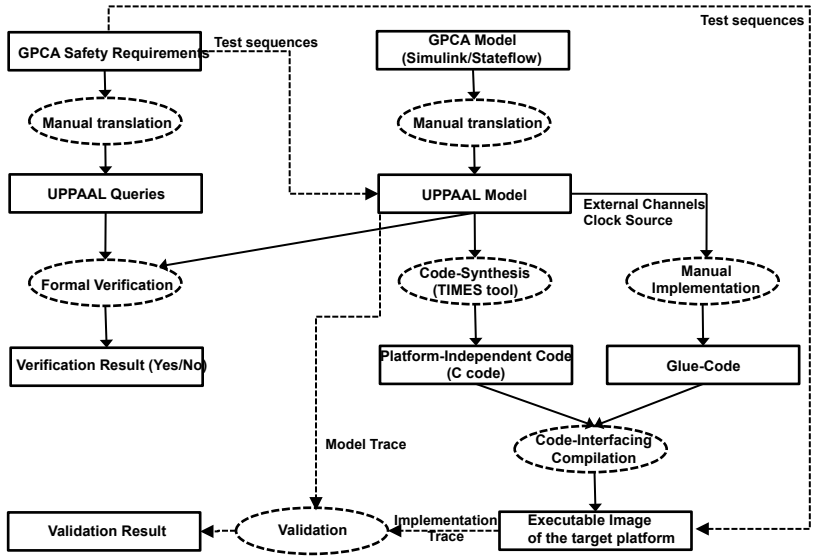
GPCA reference implementation

- Given
  - GPCA Safety Requirements
  - GPCA Model (Simulink/ Stateflow)
- Model-based development of a GPCA reference implementation

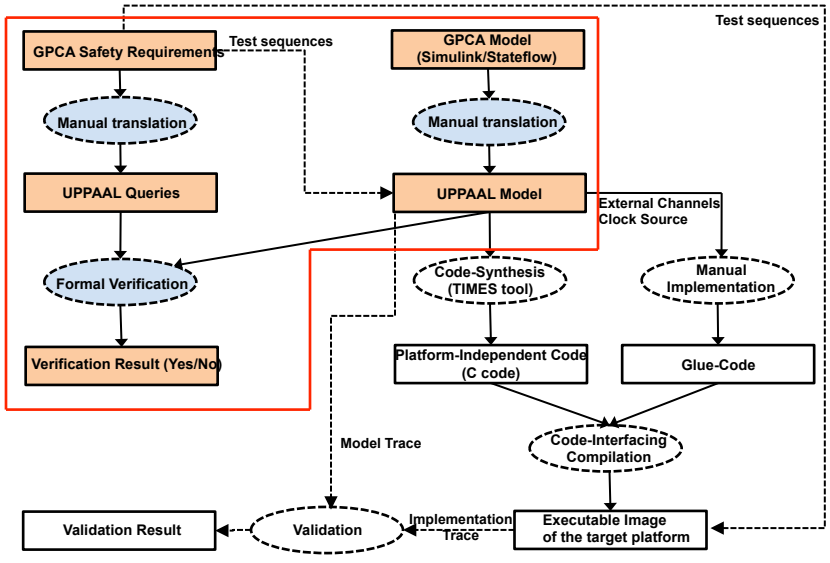


Model-Based Development of GPCA Reference Implementation

Overall Plan

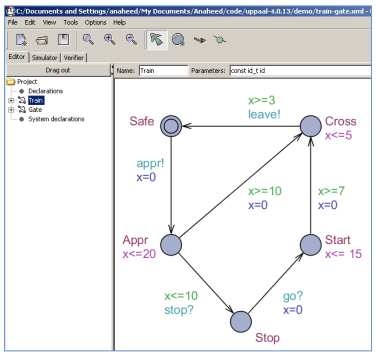


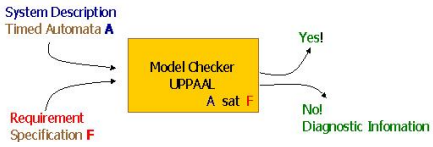
Part 1. Formal Modeling and Verification



UPPAAL (**UPP**sala + **AAL**borg = UPPAAL)

- **UPPAAL** is a tool for Modeling, Validation, and Verification
- Major functionalities:
  - A description language: network of timed automata extended with variables.
  - A Simulator : validation tool which enables examination of possible executions of a system.
  - A Model-checker: for automatic verification of safety properties by reachability analysis of the symbolic state-space.

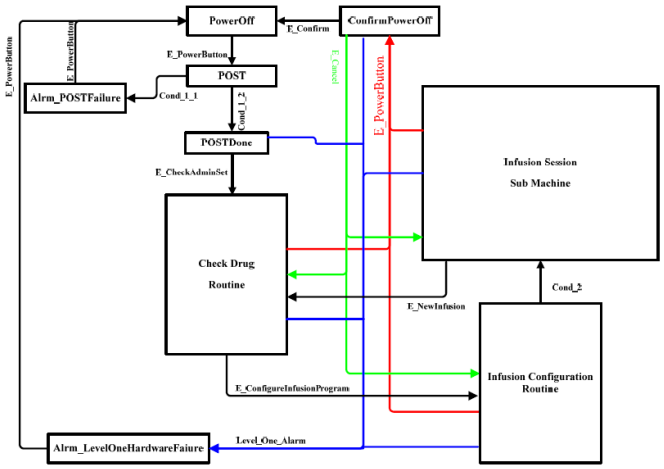




Formalization of the GPCA model

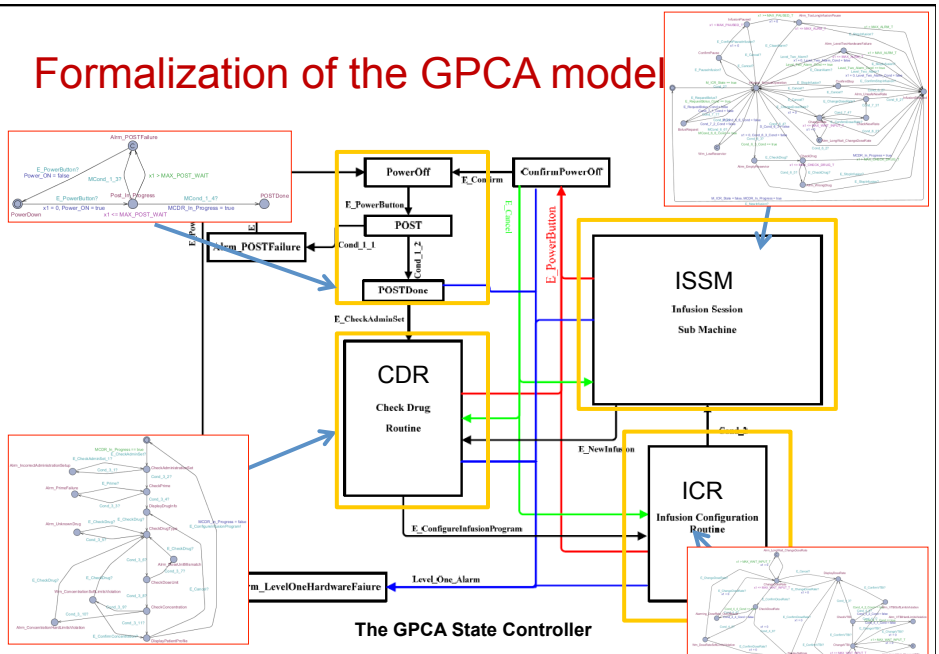
- Transform the GPCA model into a network of UPPAAL automata
  - Aims to retain as much of the syntactic structure of the GPCA model as possible following a rigorous manual process
  - Maintain one-to-one mapping between states, conditions, user actions, and transitions in the two models
    - State : Alarm-Empty-Reservoir
    - Condition : Cond-6-2 (An infusion error Empty Reservoir is detected during the ongoing infusion process.)
    - Action : E-RequestBolus (Request for a bolus dose by pressing a button)
  - Currently the UPPAAL model consists of approximately 50 states, 100 transitions, and 50 user actions and conditions

Formalization of the GPCA model



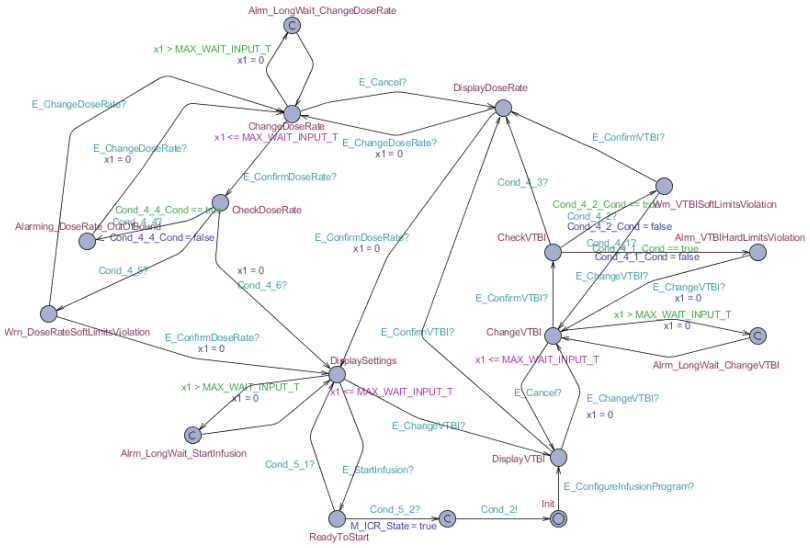
The GPCA State Controller

Formalization of the GPCA model

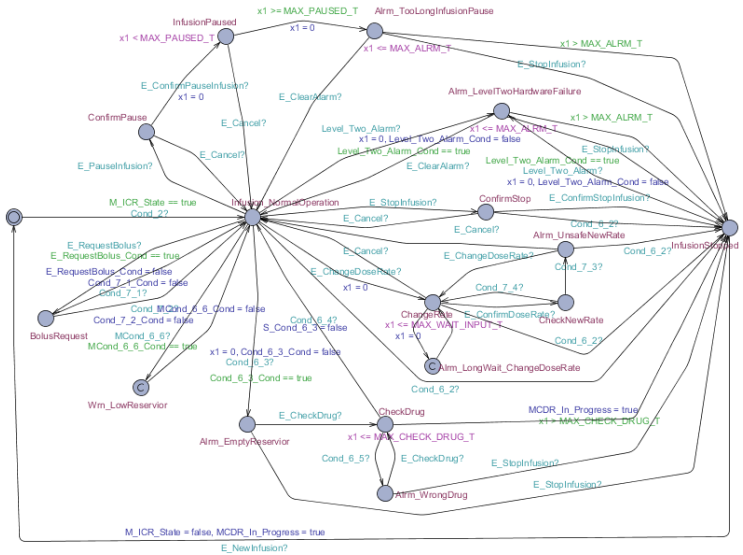


The GPCA State Controller

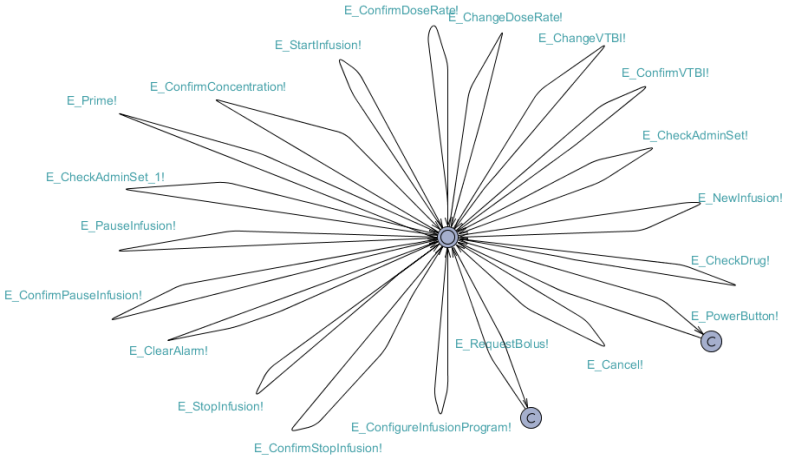
Example : Infusion Configuration Routine (ICR)



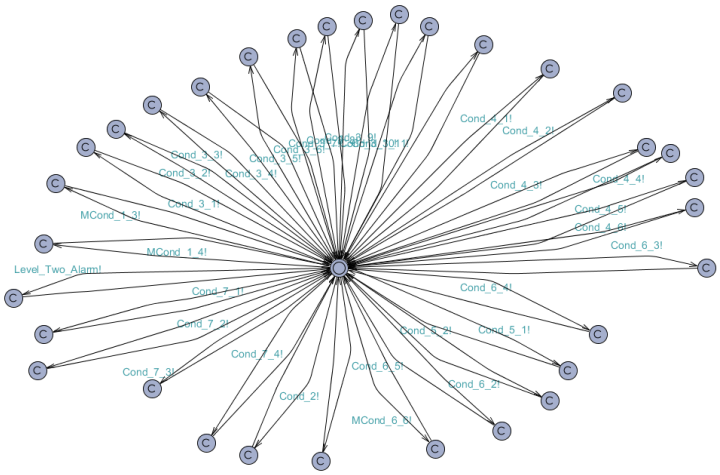
Example : Infusion Session Submachine (ISSM)



Environment : User Action



Environment : Hardware Conditions



Cond-6-3 implies “An infusion error *Empty Reservoir* is detected during the ongoing infusion process”

Formalization of the Safety Requirements

- Safety requirements are translated into temporal logic formula using the UPPAAL query language.
- Example of Safety requirement formalization
  - No bolus dose shall be possible during the POST
    - A[] (! (POST.Post-In-Progress && ISSM.BolusRequest))
  - No normal bolus doses should be administered when the pump is alarming (in an error state).
    - A[](! (ISSM.BolusRequest && CDR.Alrm-UnknownDrug))
  - The pump shall issue an alarm if paused for more than t minutes
    - (ISSM.InfusionPaused && x1 > MAX-PAUSED-T)
      -> ISSM.Alrm-TooLongInfusionPause
  - If the calculated volume of the reservoir is y ml, and an infusion is in progress, an Empty Reservoir alarm shall be issued.
    - (ISSM.Infusion-NormalOperation && Cond-6-3== true)
      -> ISSM.Alrm-EmptyReservior

Formalization of the Safety Requirements

- Not all safety requirements can be translated into temporal logic formula
- Categorization of the safety requirements:

  Category 1) A safety requirement can be formalized and verified in the UPPAAL model. (~20 out of 97 requirements)
  - No bolus dose shall be possible during the POST
  - The pump shall issue an alert if paused for more than t minutes

Formalization of the Safety Requirements

- Not all safety requirements can be translated into temporal logic formula.
- Categorization of the safety requirements.
  - Category 1) A safety requirement can be formalized and verified in the UPPAAL model. (~20 out of 97 requirements)
    - No bolus dose shall be possible during the POST
    - The pump shall issue an alert if paused for more than t minutes
  - Category 2) A safety requirement can be formalized, but the GPCA model needs additional information to verify it. (~23 out of 97 requirements)
    - If the suspend occurs due to a fault condition, the pump shall be stopped immediately without completing the current pump stroke.

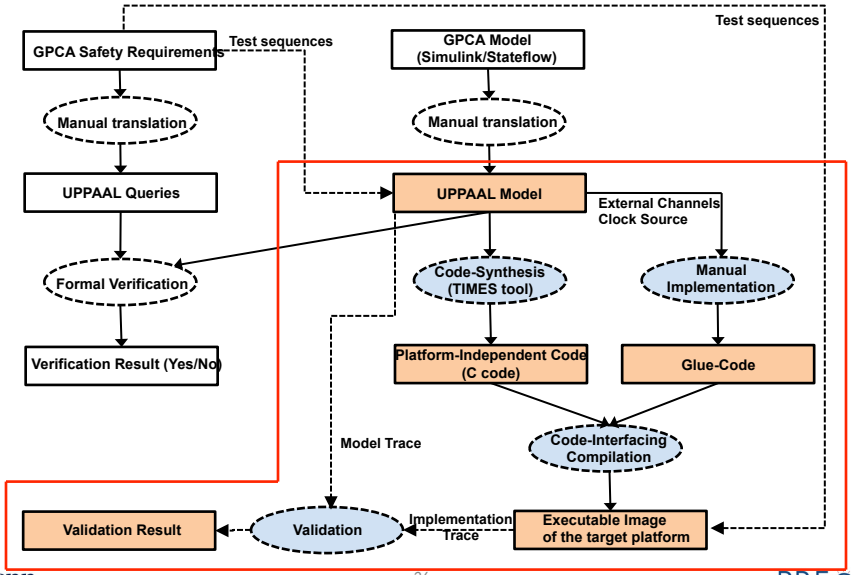
Formalization of the Safety Requirements

- Not all safety requirements can be translated into temporal logic formula.
- Categorization of the safety requirements.
  - Category 1) A safety requirement can be formalized and verified in the UPPAAL model. (~20 out of 97 requirements)
    - No bolus dose shall be possible during the POST
    - The pump shall issue an alert if paused for more than t minutes
  - Category 2) A safety requirement can be formalized, but the GPCA model needs additional information to verify it. (~23 out of 97 requirements)
    - If the suspend occurs due to a fault condition, the pump shall be stopped immediately without completing the current pump stroke.
  - Category 3) A safety requirement cannot be formalized, but can be validated at the implementation level. (~31 out of 97 requirements)
    - The flow rate for the bolus dose shall be programmable.

Formalization of the Safety Requirements

- Not all safety requirements can be translated into temporal logic formula
- Categorization of the safety requirements
  - Category 1) A safety requirement can be formalized and verified in the UPPAAL model. (~20 out of 97 requirements)
    - No bolus dose shall be possible during the POST
    - The pump shall issue an alert if paused for more than t minutes
  - Category 2) A safety requirement can be formalized, but the GPCA model needs additional information to verify it. (~23 out of 97 requirements)
    - If the suspend occurs due to a fault condition, the pump shall be stopped immediately without completing the current pump stroke
  - Category 3) A safety requirement cannot be formalized, but can be validated at the implementation level. (~31 out of 97 requirements)
    - The flow rate for the bolus dose shall be programmable
  - Category 4) A safety requirement cannot be formalized because the statement is too vague or related to the environment of the GPCA model. (~23 out of 97 requirements)
    - Flow discontinuity at low flows should be minimal ("minimal" not defined)
    - A key that is depressed shall not be identified as a distinct key press for a period of t seconds (related to UI)

Part 2: Implementation



TIMES

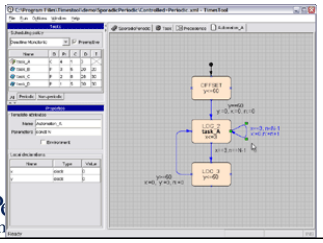
(Tool for Modeling and Implementation of Embedded Systems)

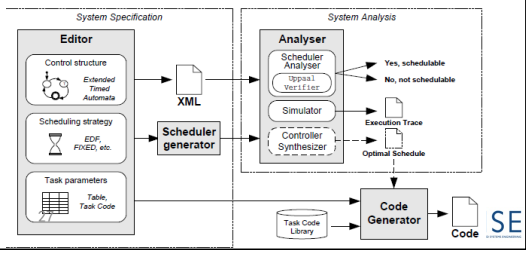
- **TIMES** is a tool set for modeling, schedulability analysis, synthesis of executable code.
- The tool is divided in three parts:
  - A system specification part :

    Designing timed automata extended with tasks.
  - A system analysis part :

    For validation using the simulator and verification using UPPAAL verifier.
  - **The Code Generation:**

    Synthesize the timed automata model into executable C-code either for Brick OS or platform-independent.





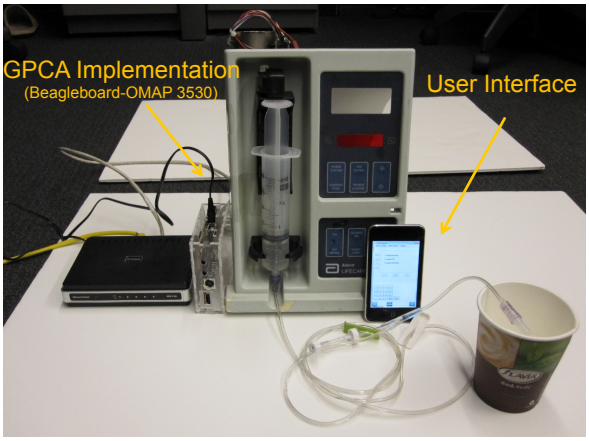
Automated Implementation

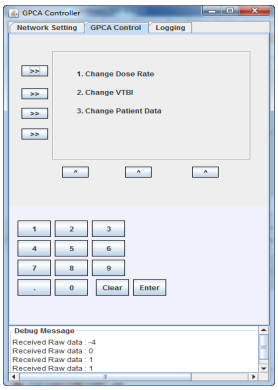
- Reasons for adopting automated implementation.
  - Preservation of verified properties (including timing properties*)
  - Manual implementation is error prone due to the large number of control states and variety of events that the code needs to react to
- Practical consideration of automated implementation.
  - The overall code structure is constrained by the code structure of the automatically generated code.
  - Abstracted functionalities need to be manually implemented.
  - Generated code needs to be customized for the target platform.

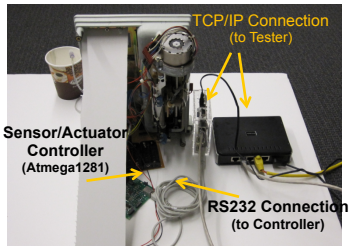
Our Implementation Platform

- Used or recalled commercial PCA pumps are available from E-Bay
- Remove the microcontroller provided by the pump manufacturer
- Interface the pump hardware (stepper motor, switches, buzzer…) to our microcontroller (Beagleboard and Atmega128 microcontroller)
- Add more hardware peripherals (e.g., sensors….)

GPCA Implementation Platform







Types of the GPCA Source Code

- GPCA model code (Platform-independent)
  - GPCA model is synthesized into C-code using TIMES tool
  - This code implements control-flow of the GPCA model depending on user-action and hardware conditions
- Glue code to interface to the target platform (Platform-dependent)
  - Clock implementation using the target platform APIs
  - Environmental interface (for user and GPCA hardware)
- Code for abstracted functionalities
  - Pump-motor driving code on transition to Infusion-Normal-Operation to inject drug to patient (e.g., providing electrical signal to the pump motor)
  - Code for updating dose rate on ChangeDoseRate state (e.g., maintaining variables for dose rate that is updated by user request)

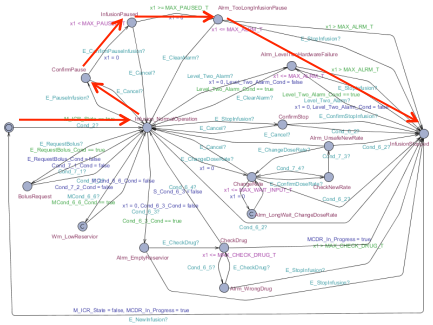
Validation of the GPCA Implementation

- Reason for requiring validation in automated implementation
  - Still exists many factors that cannot be formalized.
    - manually-written glue code, abstracted functionalities.
  - Gain higher confidence of the final implementation with the hardware peripherals
- Online conformance testing using a tester
  - A tester that consists of a monitor and input generator is additionally implemented for online conformance testing
    - The monitor observes the runtime behavior of the GPCA implementation
    - The input generator provides environmental stimulus (e.g., user input and hardware condition)
  - Safety requirements are used to generate test scenarios
    - The runtime trace of the implementation is compared to the verified UPPAAL model trace under the same test scenario

Part 3. Validation

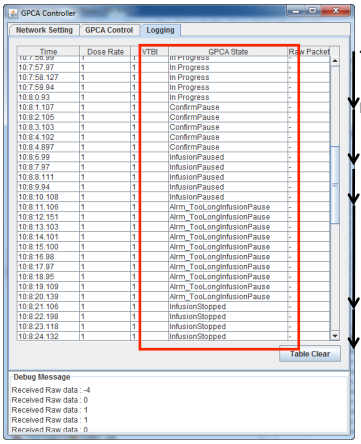
- Safety Requirement : The pump shall issue an alarm if paused for more then t minutes

<Model Trace>



The UPPAAL model
(Infusion Session Submachine)

<Implementation Trace>



The Tester screenshot

Current and Future Work

- Refine and complete the development…
  - Extend requirements to include security & privacy requirements
- Identify generic-platform dependent & specific-platform dependent glue code
  - How much need to be redone with a different pump hardware
- Assurance/safety cases for the GPCA reference implementation
  - Mock FDA submission
- Open source/testbed
- Related work
  - Network controllable medical device for MDPnP/MDCF, John Hatcliff
  - Test generation, Mats Heimdahl, Mike Whelan

Thank You!
Questions?