Formal Modelling and Analysis of CARA Control Software

Rance Cleaveland, Arnab Ray, Scott A. Smolka,
Eugene W. Stark (SUNY at Stony Brook)

Arne Skou (Aalborg University)

Support:

Ray, Cleaveland, Smolka, Stark:

ARO grants DAAD190110003, DAAD190110019

Skou:

Danish SNF grant 51-01-0014

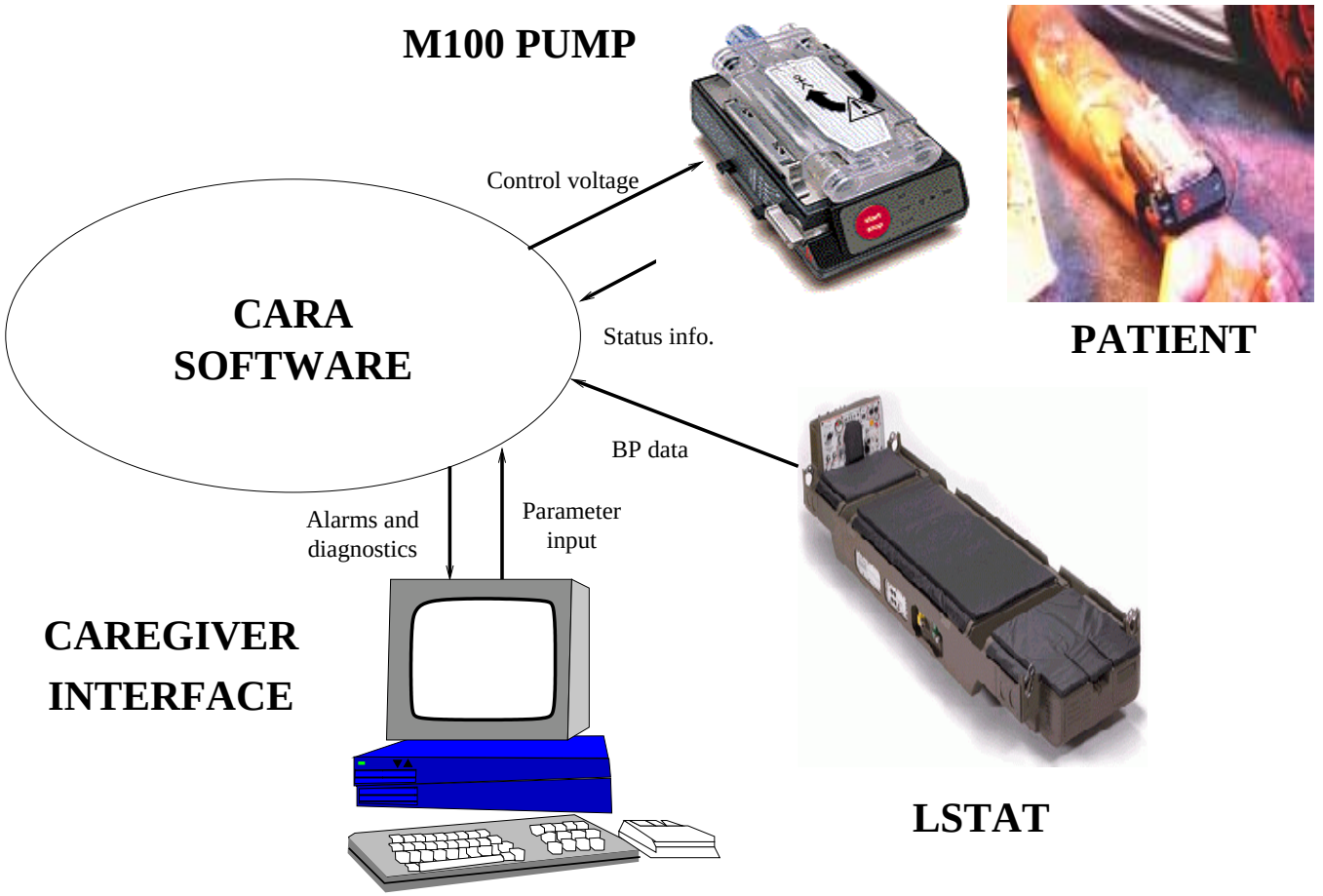
What is CARA?

Computer Assisted Resuscitation Algorithm

- *Purpose*: automate delivery of intravenous fluids to injured personnel in battlefield situations.
- *Comprises*: software to:
  - monitor patient's blood pressure.
  - control a high-output infusion pump.

CARA is a project of Walter Reed Army Institute of Research (WRAIR).

CARA System Components



Documents we Received

Received	Author	Date	Description
Feb 5	WRAIR	1/25/01	Narrative system description
Feb 5	WRAIR	1/24/01	Q&A about requirements
Apr 9	WRAIR	3/19/01	Tagged requirements
Apr 9	WRAIR	3/20/01	Updated Q&A
Apr 9	??	7/30/99	Hazards analysis and SOP

Format of the Documents

- **Narrative**: Informal prose describing system components and operation.
- **Tagged requirements**: Formal list of numbered requirements covering all aspects of system operation:
  - *"20.7 CARA will recorroborate the blood pressure control source with the cuff every 30 minutes."*
- **Q&A**: Formal list of numbered questions about the requirements, together with answers and resulting clarifications.

Notes about the Documents

An excellent starting point for formal modeling.

. . . BUT . . .

- Some critical information found *only* in the narrative description.
- Inconsistencies between narrative and formal requirements.
- Interpreting requirements often requires background assumptions (*e.g.* w.r.t. "calibration")
- Difficulty in forming an integrated view of *all* requirements (esp. w.r.t. timing).

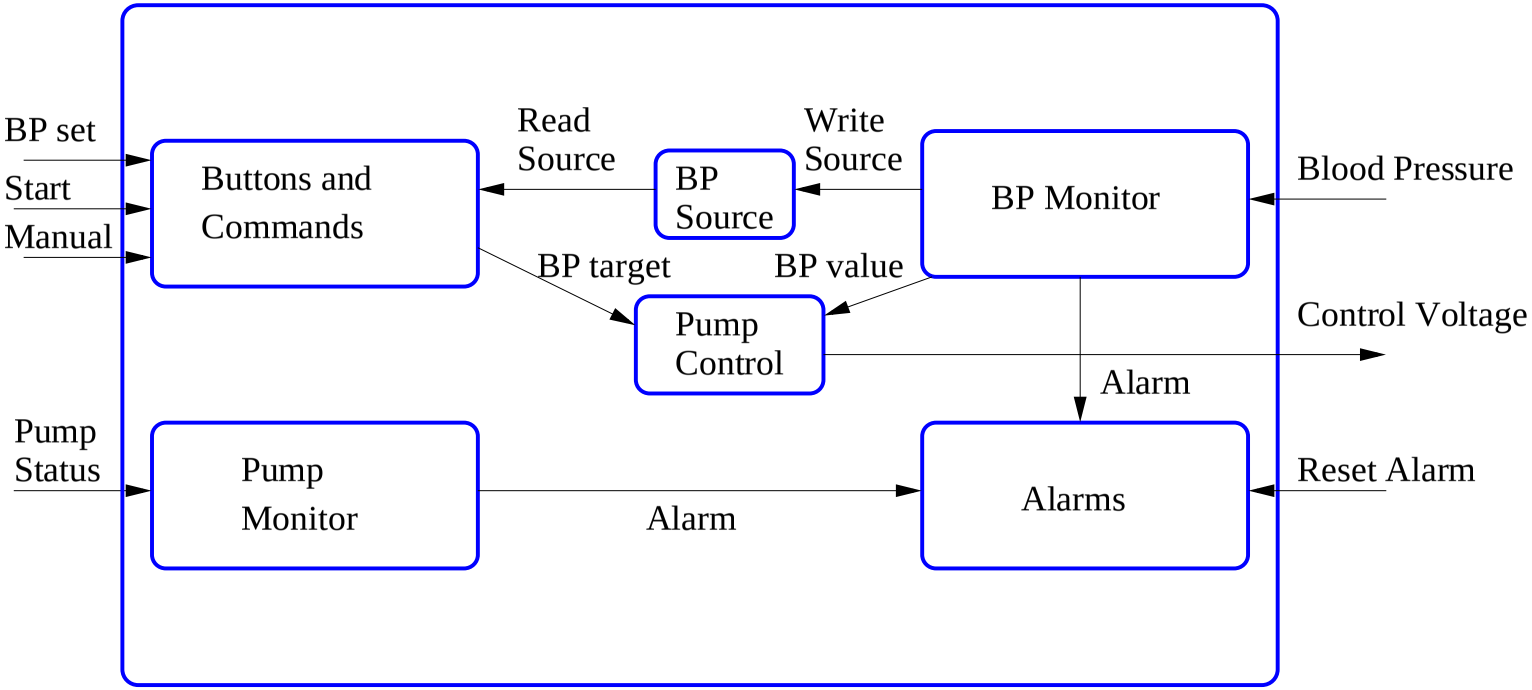
Our Objectives

- **Construct** formal models based on the requirements. The models should:
  - *Integrate* all the requirements.
  - *Refine* the requirements.
  - Serve as a *reference*.
  - Provably *satisfy* the requirements.
- **Analyze** the formal models with automated tools.
  - *Validate* the existing requirements.
  - Determine *consequences* of the requirements.

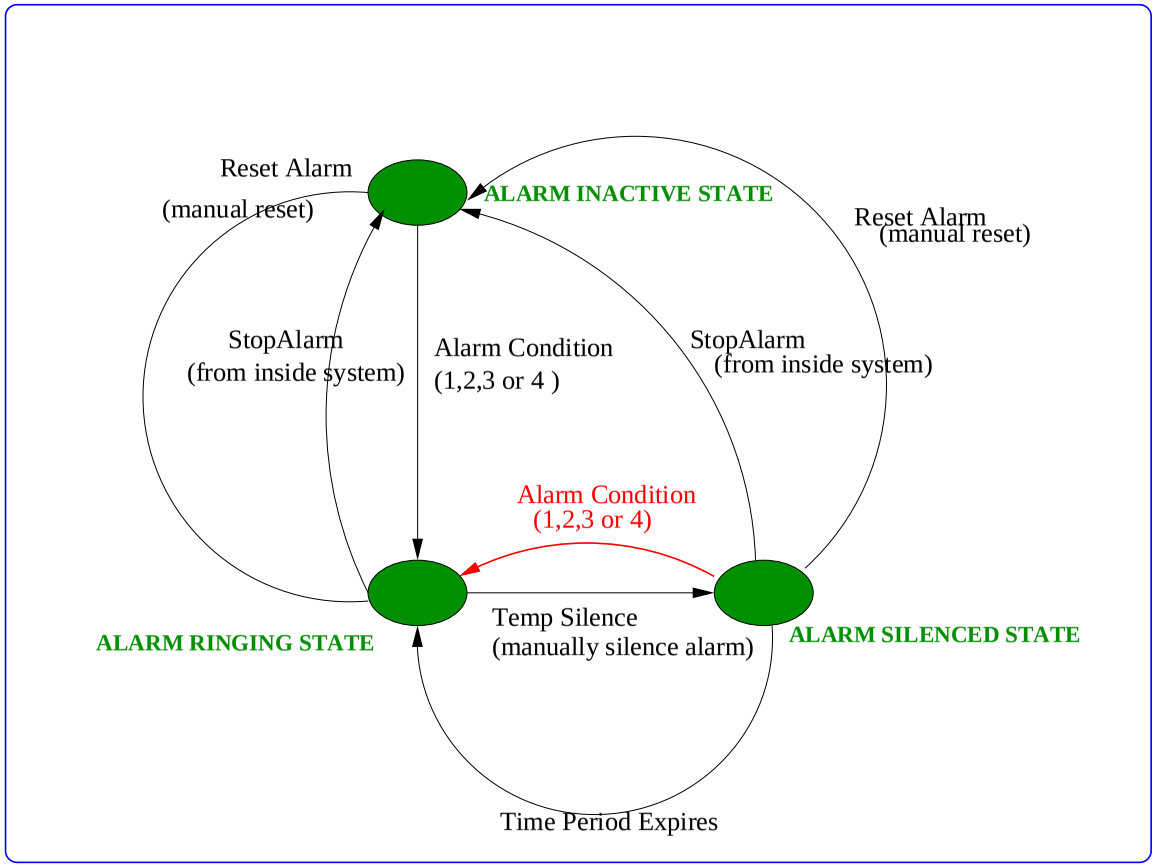
What We Did So Far

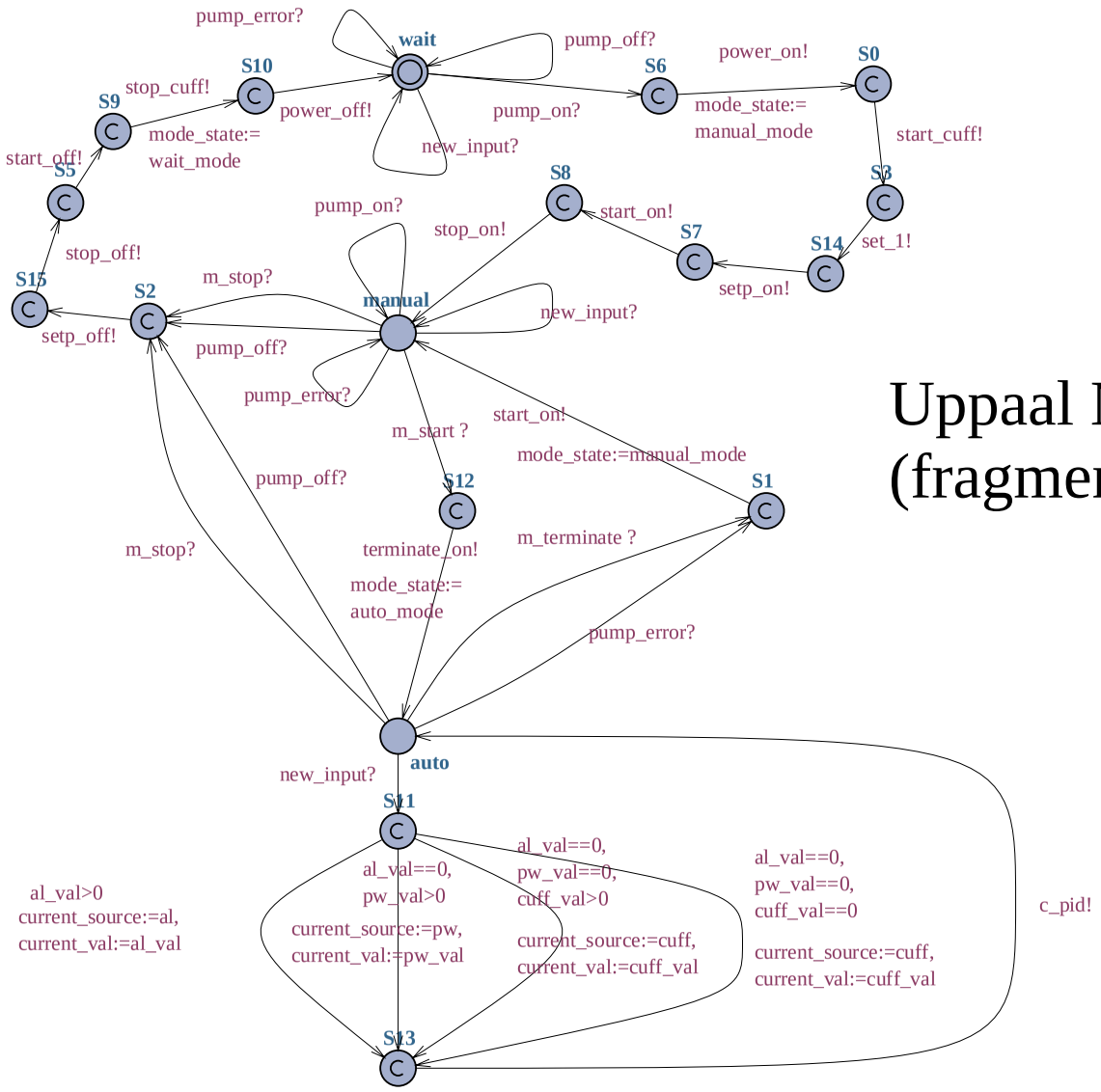
- Identified a system decomposition based on the requirements.
- Constructed two formal models:
  - One using *timed automata* and the *UPPAAL* tool.
  - One using *timed CCS* and the *Concurrency Workbench* tool.
- Performed some analyses on these models.
  - Some problems with state-space size (as usual).
  - Some analyses succeeded on abstractions of the model.

System Decomposition



Common High-Level Structure (example)





Uppaal Model (fragment)

TCCS/CWB Model (fragment)

```
proc Autocontrol = Autocontrol_int[MANUAL_STATE>
                    'MANUAL_STATE          .Manual_state
                   + MANUAL_BUTTON_PRESSED.Manual_state
                   + STOP                 .Wait

proc Autocontrol_int = (PID_Active_init
                        |Pump_init
                        |BP_monitor_init
                        |Alarm_inactive
                        |Input_active
                        )\Internals4

proc PID_Active_init = PID_Active_int[>
                        new_input_parameter .PID_Active_init

proc PID_Active_int = SETPT_NOT_REACHED.'VOLTAGE1.PID_Active_int
                      +  SETPT_REACHED.'VOLTAGE0.PID_Active_int
                      +  SETPT_EXCEEDED.'VOLTAGE0.PID_Active_int
```

A Timing Issue Studied with UPPAAL

Uppaal was used to investigate the following question:

- Is it consistent with the requirements for a nonzero pump control voltage to persist after the BP set point has been reached?

The following answer was found:

- The requirements do not rule this out for up to 10 minutes after BP setpoint reached.
- It is not possible for a nonzero pump control voltage to persist for longer than 10 minutes after the BP setpoint has been reached.

Some Analyses Performed with CWB

- "Whenever an error condition occurs an alarm is sounded."
- "The system always notices when the pump gets plugged or unplugged."
- "It is not possible for the system to 'lock up.' "

Other Benefits of Formalization

The construction of formal models has benefits even if complete analysis is not possible:

- Forces an examination of interactions between requirements.
- Models can be simulated or executed to further explore requirements.
- Our modeling led to the discovery of several issues with the requirements.

Requirements Issue #1

There exist inconsistencies between the documents, but none can be regarded as the sole authority:

- R25 states that if a cuff pressure should be used, then there should be 5 initial readings taken at 1-minute intervals.
- Q30 says that all references to 5 initial readings should be removed.
- The requirements document is more recent than the Q&A.

Requirements Issue #2

- R20.8 states that a higher priority BP source that comes online should be corroborated using the current source.
- R20.7 states that cuff pressure will be used for corroboration. every 30 minutes.

Apparently there is a difference between the 30-minute corroboration cycle and immediate corroboration of a new source.

Requirements Issue #3

- Q70 specifies a specific order for checking the status of some pump parameters that have to be carried out at 5 second intervals.
- Q74 states that to flag an error three successive "bad" readings at 1 second intervals have to occur.
- It is not clear whether satisfying Q74 can cause Q70 to be violated.

Requirements Issue #4

Several requirements require that cuff pressures be read:

- R27 states that cuff pressures will be taken at a rate of one every 1 minute, 2 minutes, 5 minutes, or 10 minutes, depending on the mean BP.
- R44.3 concerns the initiation of a request for cuff pressure when only the cuff is used and an invalid reading is obtained.
- R20.7 states that CARA will re-corroborate the BP control source with the cuff every 30 minutes.

- R20.7.1 describes a situation in which the 30 minute re-corroboration must be delayed due to an existing active corroboration attempt.
- Apparently stricken requirements dealing with PW calibration also require cuff readings.

How is contention for the cuff to be handled safely?

Requirements Issue #5

Under what conditions does termination of auto-control mode raise a dialog box?

- R48 concerns the "auto-control termination sequence," calling for a confirmation dialog when "terminate auto-control" is selected.
- R6, R7, R8, and R11 describe conditions under which auto-control is terminated due to error conditions, but do not mention confirmation dialog.

Requirements Issue #6

Under certain conditions, the system will wait indefinitely for a response to an override dialog:

- R20.3.2 states that a corroboration failure should bring up a dialog box with YES/NO override buttons.
- R20.8.1 states that an override question must be answered before corroboration of a new higher-priority source can begin.

Use of executable specifications in a rapid prototype could help determine if this behavior is intended and reasonable.

Requirements Issue #7

The requirements concerning the BP cuff are complex:

- R27 states that cuff pressures will be taken at 1, 2, 5, or 10-minute intervals, depending on the mean BP.
- R30 (stricken) states that new voltages should be calculated after every blood pressure reading when the cuff is used.
- R25 requires 5 BP readings at one minute intervals under certain conditions.

Apparently the pump control voltage recalculation interval varies with the frequency of cuff readings.

Other Results so Far

We identified some shortcomings and relative strengths of the tools we used:

- UPPAAL: strong on graphics, weak on hierarchy.
- CWB: weak on graphics, but readily extended to improve handling of hierarchy.

Planned Work

- Make a more comprehensive reference model with fewer simplifying assumptions and better coverage of the requirements.
- Make a more systematic attempt to identify and verify system properties.
- Construct an additional formal model using probabilistic I/O automata.
- Revisit our previous work on timing abstraction (treats timing as priority) for TCCS.

Conclusion

A very nice case study, exactly the right size and complexity for our current tools.

- It has already pointed up some strengths and weaknesses of existing tools.
- We've identified some issues with the requirements, which we hope might benefit the CARA project.