EECS4312 Isolette Assignment

Anton Sitkovets (cse31027@cse.yorku.ca)
Mina Zaki (zakim@cse.yorku.ca)

November 10, 2016

You may work on your own or in a team of no more than two students. **Submit only one document under one Prism account.**

Prism account used for submission: cse31027

Keep track of your revisions in the table below.

Revisions

Date	Revision	Description
10 November 2016	1.0	Final requirements document

Requirements Document: Temperature control for an Isolette

Contents

List of Figures

List of Tables

1 System Overview

The System Under Development (SUD) is a computer controller for the thermostat of an Isolette.[1] An Isolette is an incubator for for an infant that provides controlled temperature, humidity and oxygen (Fig. 1). Isolettes are used extensively in Neonatal Intensive Care Units for the care of premature infants.

This requirements document is specifically for the control of temperature. The purpose of the Isolette computer controller is to maintain the air temperature of an Isolette within a desired range. It senses the current temperature of the Isolette and turns the heat source on and off to warm the air as needed. If the temperature falls too far below or rises too far above the desired temperature range, it activates an alarm to alert the nurse. The system allows the nurse to set the desired temperature range and to set the alarm temperature range outside the desired temperature range of which the alarm should be activated. This requirements documents follows the specification in [**?**] (Appendix A) except where noted.

Figure 1: Isolette

[1] The image in Fig 1 is from: `www.nufer-medical.ch`.

2 Context Diagram

See Fig. A-1 in [?]. The System Under Description (SUD) is a computer *controller* to regulate the temperature of the Isolette. Everything else including the Operator Interface (described in [?]) is in the ecosystem (i.e. in the environment of the controller). The monitored variables and controlled variables for the controller are in Table 1 and Table 2, respectively. For clarity, simplicity and safety, there are some differences between the specifications in this document and the descriptions in [?].[2]

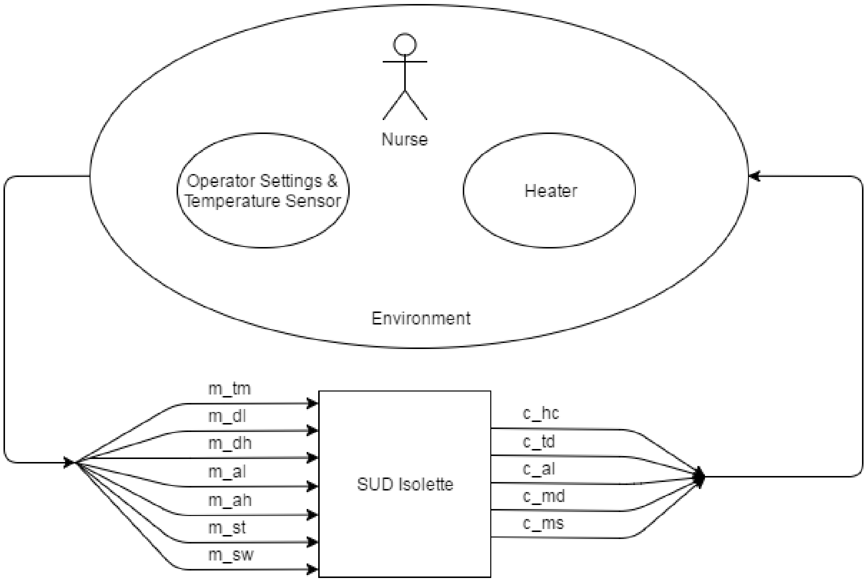


Figure 2: Context Diagram

3 Goals

The high-level goals (G) of the system are:

- G1—The Infant should be kept at a safe and comfortable temperature.
- G2—The Nurse should be warned if the Infant becomes too hot or too cold.
- G3—The cost of manufacturing the computer controller for the thermostat should be as low as possible.

[2] Documented in the write-up to this assignment: `assign1-spec.pdf`.

4 Monitored Variables

The monitored variables are a subset of those described in [?].[3] There is a single status variable m_st that is *invalid* whenever any one of the operator inputs or temperature sensor are in a failed state. Otherwise types and ranges are as in [?].

Name	Type	Range	Units	Physical Interpretation
m_tm	$\mathbb{R}$	$68 \mathrel{..} 105$	°F	actual temperature of Isolette air temperature from sensor
m_dl	$\mathbb{Z}$	$97 \mathrel{..} 99$	°F	desired lower temperature set by operator
m_dh	$\mathbb{Z}$	$98 \mathrel{..} 100$	°F	desired higher temperature set by operator
m_al	$\mathbb{Z}$	$93 \mathrel{..} 98$	°F	lower alarm temperature set by operator
m_ah	$\mathbb{Z}$	$99 \mathrel{..} 103$	°F	higher alarm temperature set by operator
m_st	Enumerated	{valid, invalid}		status of sensor and operator settings
m_sw	Enumerated	{on, off}		switch set by operator

Table 1: Monitored Variables

5 Controlled Variables

The controlled variables are a subset of those described in [?].[4] In addition, there is a mode display c_md and a message display c_ms.[5]

[3] With some change of nomenclature. Monitored variables have an “m” prefix.

[4] With some change of nomenclature. Controlled variables have a “c” prefix.

[5] The mode “off” is added to that of Fig. A-4 in [?], and the mode transitions have been changed.

Name	Type	Range	Units	Physical Interpretation
c_hc	Enumerated	{on, off}		heat control: command to turn heat source on or off
c_td	$\mathbb{Z}$	$\{0\} \cup \{68 \mathinner{..} 105\}$	°F	displayed temperature of Isolette (zero when Isolette is off)
c_al	Enumerated	{off, on}		sound alarm to call nurse
c_md	Enumerated	{off, init, normal, failed}		mode of Isolette operation (failed if $m\_st = invalid$)
c_ms	Enumerated	{OK, TooHot, TooCold, SensorIssue}		messages to display to nurse

Table 2: Controlled Variables

6 Mode Diagram

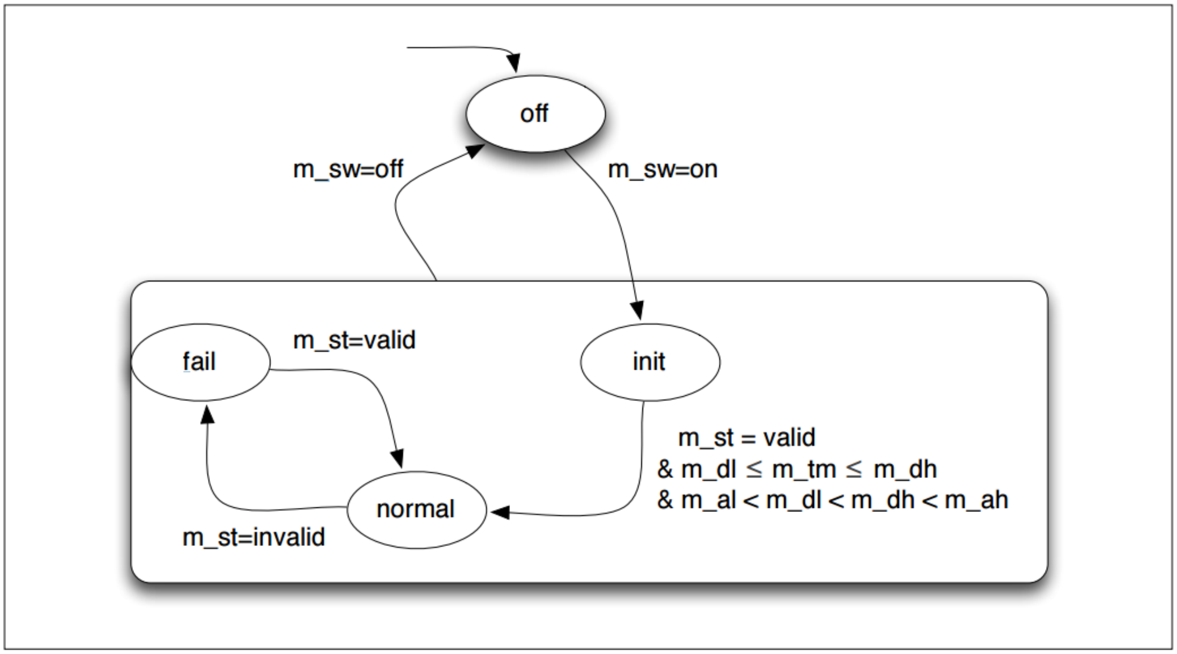


Figure 3: Mode Diagram

7 E/R-descriptions

REQ1	The *controller* shall operate in one of four modes: *off*, *init*, *normal* and *fail*.	See statechart in Fig. 3.

REQ2	In the *normal* mode, the temperature controller shall maintain current temperature inside the Isolette within a set temperature range (the *desired* range).	

REQ3	In the normal mode, the temperature controller shall maintain current temperature inside the Isolette within a set temperature range (the desired range).	If the sensor is not functioning, requires immediate attention and the system will be in fail mode.

REQ4	The system should not display two error messages at the same time.	Avoid bombarding the user with error messages

REQ5	Prioritize error message display to consider sensor status as most important followed by alarm temperature.	If the sensor is not functioning, requires immediate attention and the system will be in fail mode.

ENV6	The higher alarm temperature will always be higher than the lower alarm temperature.	To create an area that can be the desired temperature range.

ENV7	The displayed temperature is an integer	The current temperature reading is rounded to the nearest integer

ENV8	The status of the sensor and operator settings can either be valid or invalid	It is important to know if the sensor readings are accurate or else any decisions made may put the infant in danger

8 Abstract variables needed for the Function Table

Abstract variables are not needed.

9 Function Table

9.1 Function Table for mode display: c_md

				c_md(i)
i = 0				off
i >0	m_sw = off			off
	m_sw = on	c_md(i-1) = off		init
		c_md(i-1) = init	C1	normal
			$\neg C1$	c_md(i-1)
		c_md(i-1) = normal	m_st = invalid	fail
			m_st = valid	c_md(i-1)
		c_md(i-1) = failed	m_st = invalid	c_md(i-1)
			m_st = valid	normal

Table 3: Function Table for c_md

9.2 Function Table for heat control: c_hc

				c_hc (i)
i = 0				off
i >0	c_md (i-1) = off			off
	C4	m_dl (i) $<$ m_dh (i)	m_tm (i) $<$ m_dl (i)	on
			m_dl (i) $<=$ m_tm (i) $<=$ m_dh (i)	c_hc (i-1)
			m_tm (i) $>$ m_dh (i)	off
		m_dl (i) $>=$ m_dh (i)		c_hc (i-1)
	c_md (i-1) = failed			off

Table 4: Function Table for heat control: c_hc

9.3 Function Table for temperature display: c_td

		c_td(i)
i = 0		0
i >0	c_md(i-1) = off	0
	c_md(i-1) = init	0
	c_md(i-1) = normal	$\lfloor m\_tm + 0.5 \rfloor$
	c_md(i-1) = failed	0

Table 5: Function Table for temperature display: c_td

9.4 Function Table for temperature display: c_al

					c_al(i)
i = 0					off
i >0	C2				c_al(i-1)
	$\neg C2$	C3			on
		$\neg C3$	c_al(i-1) = off		c_al(i-1)
			c_al(i-1) = on	held_for(c_al,10)(i-1)	off
				$\neg held\_for(c\_al, 10)(i-1)$	on

Table 6: Function Table for temperature display: c_al

Condition	Meaning
C1	$(m\_st$ (i) = valid) $\wedge$ $(m\_dl(i) \leq m\_tm(i) \leq m\_dh(i))$ $\wedge$ $(m\_al(i) < m\_dl(i) < m\_dh(i) < m\_ah(i))$
C2	$(m\_al$ (i) $\leq m\_tm(i) < m\_al(i) + 0.5) \vee (m\_ah(i) - 0.5 < m\_tm(i) \leq m\_ah(i))$
C3	$(m\_tm$ (i) $< m\_al$(i)) $\vee$ $(m\_tm(i) > m\_ah(i)) \vee (m\_st(i) = invalid)$
C4	$(c\_md$(i-1) = init) $\vee$ $(c\_md(i-1) = normal)$

Table 7: Legend for Conditional Abbreviations

9.5 Function Table for message display: c_ms

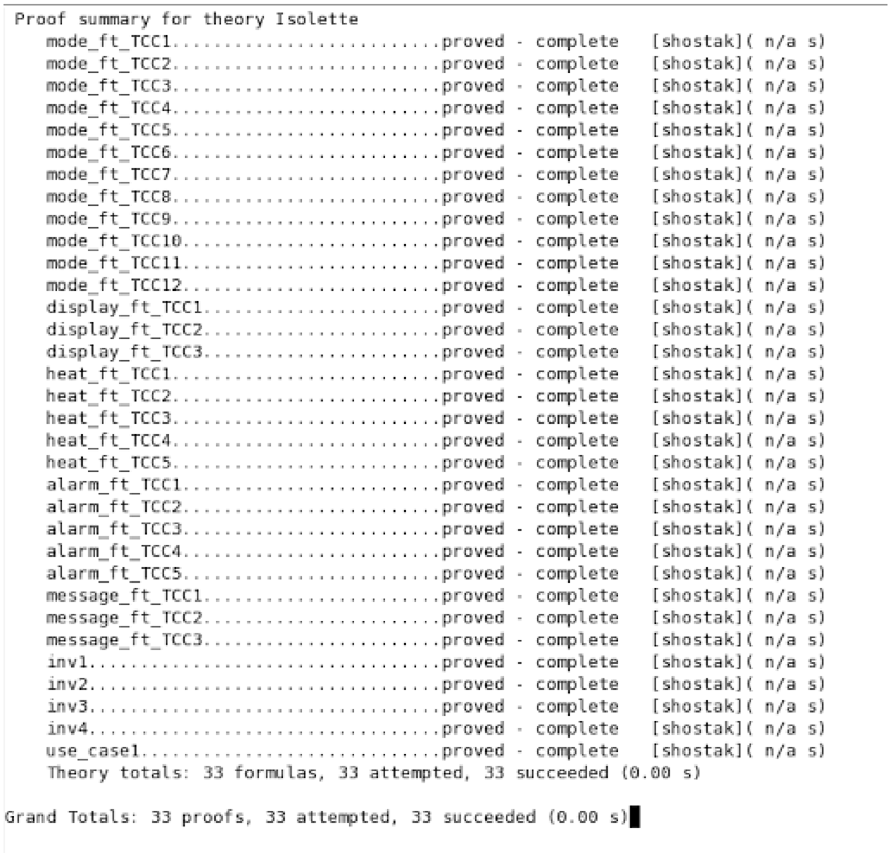
			c_ms	Meaning
i = 0			OK	All ok
i >0	m_st = invalid		SensorIssue	The temperature sensor or operator settings have failed.
	m_st = valid	m_tm >m_ah	TooHot	The current temperature is higher than the higher alarm temperature.
		m_tm <m_al	TooCold	The current temperature is lower than the lower alarm temperature.
		ELSE	OK	All ok

Table 8: Function Table for temperature display: c_ms

10 Validation

Proof of completeness and disjointness and validation of the requirements using PVS.

Include the PVS sources in the appendix to this document but summarize the proofs here.



```
 Proof summary for theory Isolette
    mode_ft_TCC1.........................proved - complete   [shostak]( n/a s)
    mode_ft_TCC2.........................proved - complete   [shostak]( n/a s)
    mode_ft_TCC3.........................proved - complete   [shostak]( n/a s)
    mode_ft_TCC4.........................proved - complete   [shostak]( n/a s)
    mode_ft_TCC5.........................proved - complete   [shostak]( n/a s)
    mode_ft_TCC6.........................proved - complete   [shostak]( n/a s)
    mode_ft_TCC7.........................proved - complete   [shostak]( n/a s)
    mode_ft_TCC8.........................proved - complete   [shostak]( n/a s)
    mode_ft_TCC9.........................proved - complete   [shostak]( n/a s)
    mode_ft_TCC10........................proved - complete   [shostak]( n/a s)
    mode_ft_TCC11........................proved - complete   [shostak]( n/a s)
    mode_ft_TCC12........................proved - complete   [shostak]( n/a s)
    display_ft_TCC1......................proved - complete   [shostak]( n/a s)
    display_ft_TCC2......................proved - complete   [shostak]( n/a s)
    display_ft_TCC3......................proved - complete   [shostak]( n/a s)
    heat_ft_TCC1.........................proved - complete   [shostak]( n/a s)
    heat_ft_TCC2.........................proved - complete   [shostak]( n/a s)
    heat_ft_TCC3.........................proved - complete   [shostak]( n/a s)
    heat_ft_TCC4.........................proved - complete   [shostak]( n/a s)
    heat_ft_TCC5.........................proved - complete   [shostak]( n/a s)
    alarm_ft_TCC1........................proved - complete   [shostak]( n/a s)
    alarm_ft_TCC2........................proved - complete   [shostak]( n/a s)
    alarm_ft_TCC3........................proved - complete   [shostak]( n/a s)
    alarm_ft_TCC4........................proved - complete   [shostak]( n/a s)
    alarm_ft_TCC5........................proved - complete   [shostak]( n/a s)
    message_ft_TCC1......................proved - complete   [shostak]( n/a s)
    message_ft_TCC2......................proved - complete   [shostak]( n/a s)
    message_ft_TCC3......................proved - complete   [shostak]( n/a s)
    inv1.................................proved - complete   [shostak]( n/a s)
    inv2.................................proved - complete   [shostak]( n/a s)
    inv3.................................proved - complete   [shostak]( n/a s)
    inv4.................................proved - complete   [shostak]( n/a s)
    use_case1............................proved - complete   [shostak]( n/a s)
    Theory totals: 33 formulas, 33 attempted, 33 succeeded (0.00 s)

Grand Totals: 33 proofs, 33 attempted, 33 succeeded (0.00 s)
```

Figure 4: Proof of completeness, disjointness and validation of the requirements

11 Use Cases

See Section A2 of [?] for some use cases. The use cases need to be adapted to the revised descriptions of the previous sections of this document.

12 Acceptance Tests

In this section, the use cases have to be converted into precise acceptance tests (using the function table to describe pre/post conditions) to be run when the design and implementation are complete.

13 Traceability

Matrix to show which acceptance tests passed, and which R-descriptions they checked.

14 Glossary

The definition of important terms is placed in this section. You are not required to complete this.