A Coloured Petri Nets Reference Model of Insulin Infusion Pump Control Systems: Assisting the Certification Process

Tássio Fernandes Costa
Federal Rural University of the Semi-Arid
Pau dos Ferros, Brazil
tassiokp206@gmail.com

Alvaro Sobrinho
Academic Unit of Garanhuns
Federal Rural University of Pernambuco
Garanhuns, Brazil
alvaro.sobrinho@ufrpe.br

Lenardo Chaves e Silva
Engineering and Technology Department
Federal Rural University of the Semi-Arid
Pau dos Ferros, Brazil
lenardo@ufersa.edu.br

Leandro Dias da Silva
Computing Institute
Federal University of Alagoas
Maceio, Brazil
leandrodias@ic.ufal.br

Angelo Perkusich
Electrical Engineering Department
Federal University of Campina Grande
Campina Grande, Brazil
perkusic@dee.ufcg.edu.br

***Abstract*—Insulin infusion pumps are safety-critical systems that require the evaluation of government regulatory agencies (e.g., FDA) before commercialization. This paper presents a Coloured Petri Nets (CPN) reference model of insulin infusion pump control systems to assist manufacturers to generate safety evidence and evaluate quality for certification purposes. It also describes a case study on the ACCU-CHECK Spirit system version 2.XX to evaluate the CPN model. Therefore, the reference model is available and verified. Additionally, the case study is useful to show how manufacturers can reuse it during a certification process. The main contribution of this paper consists on a project artifact to decrease costs and development time of insulin infusion pump control systems, in addition to increase confidence on safety properties.**

***Index Terms*—Modeling, Simulation, Industrial Informatics**

I. Introduction

Embedded medical systems can be devided into three classes: monitoring (i.e., acquires and/or verifies bio-markers), diagnosis (i.e., identify risks of diseases), and treatment (i.e., actuates to conduct a medical prescription). For instance, Sharanya, Abhishek, and Reddy (2017) [12] present a monitoring system for cardiac diseases. The system proposed by Muangprathub and Boonjing (2014) [9] conducts the diagnosis of 50 differente diseases. Sjaaheim et al. (2014) [14] describe a portable medical system to treat and diagnose traumatic brain injuries.

The present paper focuses on treatment medical systems. More specifically, the treatment of diabetes using insulin infusion pump control systems. This type of system contains hardware and software components. On the one hand, hardware components compose the infusion pump that simulates the behaviors of the pancreas. On the other hand, an embedded software controls the system to ensure the correct functioning, aiming to prevent hazard situations [8].

As a safety-critical system, manufacturers should analyze the behaviors of insulin infusion pump control systems to provide, at least, the minimum required guarantee of correct functioning. In this context, formal modeling languages are useful mathematical tools to represent requirements with a high level of accuracy. Coloured Petri Nets (CPN) is an example of this type of language [7].

The CPN language enables one to combine the concepts of the traditional Petri nets formalism with the CPN/ML functional programming language, hierarchy, and timing resources. The CPN (along with the CPN/Tools) fits to the purpose of this work considering that insulin infusion pump control systems require characteristics such as concurrency, communication, and synchronization.

Therefore, the usage of the CPN formal modeling language is an alternative for manufacturers to increase confidence on the correct functioning of insulin infusion pump control systems. However, in order to apply this type of formal technique, it is necessary to get expert knowledge and to increase development time, resulting in project costs. In this paper, we argue that reusable CPN reference models of insulin infusion pump control systems have the potential to reduce the impact of these issues [15].

Additionally, regulatory government agencies (e.g., the US Food and Drug Administration - FDA) require manufacturers to formally demonstrate that insulin infusion pump control systems do not put users in hazard situations [2]. In 2010, the FDA released the infusion pump improvement initiative [6]. It concerned with issues such as software defects, user interface, and mechanical or electrical failures. As some results of the infusion pump improvement initiative, the FDA developed and

The authors would like to thank CNPq and CAPES for the financial support and research grants.

978-1-7281-4878-6/19/$31.00 ©2019 IEEE

released some artifacts, such as the 2014 guidance for industry and FDA staffs during the development of insulin infusion pumps, considering the product's life cycle [3].

However, up to this date, there still exists a high number of regularoty recalls regarding systems' malfunctioning [11]. In order to face this problem, manufacturers submit the system under development to a certification process conducted by regulatory agencies based on prescriptive standards (e.g., ISO 14971 [5]) and quality attributes (e.g., safety). The isolated usage of informal and semi-formal specifications contributes to decreasing confidence in the correct functioning of insulin infusion pump control systems.

This paper presents the formal specification of a CPN reference model of insulin infusion pump control systems as a project artifact to increase confidence on system behaviors, generate safety evidence and evaluate quality for certification purposes. It also describes a case study on a commercial system (the ACCU-CHECK Spirit version 2.XX [1]) to evaluate the reference model by means of the model checking technique. Although not presented in this paper (due to space constraints), the evaluation was also carried out through simulations. Additionally, it shows how manufacturers can reuse the reference model during a certification process. This work extends the contributions presented by Silva et al. (2015) [13]. The main contributions of this paper consist of:

- a modular and parameterized reference model of insulin infusion pump control systems; and
- a case study to show how manufacturers can generate evidence for certification.

The paper is structured as follows. Section II details related works. Section III describes the formal definition of CPN used to specify the reference model. Section IV presents the CPN reference model of insulin infusion pump control systems. Section V describes a sample of the model evaluation by means of the model checking technique. Section VI concludes the work and envision future research directions.

II. Related Work

Although the usage of CPN as the formal modeling language, there exist other languages and graphical representations for the same purpose (e.g., Automata). An example of the usage of Automata is the work presented by Niezen and Eslambolchilar (2016) [10]. The authors apply hybrid Automata to specify a model of human operator to interact with medical devices.

Silva et al. (2015) [13] describe a clinical scenario based on an insulin infusion pump control system to evaluate an approach to assist the modeling and validation (e.g., safety properties) of medical cyber-physical systems. Simulink block diagrams represent the behaviors of the system. The basic components of an insulin infusion pump control system consist of the needle to pump the insulin, controller, display, power source, clock, and insulin cartridge. Examples of safety properties of theses systems include:

- if cartridges level is equal to 0, then the cartridges should be empty and the pump should stop; and
- if cartridges level is lower than the administered insulin dosage then the pump should stop.

The FDA is concerned with quality attributes of infusion pumps. On the one hand, the FDA Generic Infusion Pump (GIP) project is an example of initiative to increase confidence on insulin infusion pump control systems. For instance, the project faces the risk analysis by a generic architectural specification [16]. It is a high-level representation that does not enable the execution of an instance considering time constraints and the formal verification. On the other hand, Hatcliff et al. (2018) [4] conduct the open patient controlled analgesic pump project to provide artifacts such as use cases, testing and simulation infrastructure, risk management artifacts, and assurance cases. However, up to the present date, there is not a freely available generic insulin infusion pump executable model to assist manufacturers during the certification process.

In a CPN related work, Sobrinho et al. (2017) [15] propose a reference model to assist the certification of biomedical systems using CPN. It specifies hardware and software components, and applies simulations and the model checking technique to evaluate the model.

III. Background

The CPN reference model of insulin infusion pump control systems is composed of elements such as places (ellipses), transitions (rectangles), data types, and hierarchy. A *Coloured Petri Net Module* is a tuple CPN_M = (P, T, A, Σ, V, C, G, E, I, T_{sub}, P_{port}, PT):

1) P is a finite set of places.
2) T is a finite set of transitions such that $P \cap T = \theta$.
3) $A \subseteq P \times T \cup T \times P$ is a set of directed arcs.
4) Σ is a finite non-empty set of colors..
5) V is a finite set of typed variables such that $Type[v] \in \Sigma$ for all variables $v \in V$.
6) $C : P \rightarrow \Sigma$ is a color set function that assigns a color set to each place.
7) $G : T \rightarrow EXPR_V$ is a guard function that assigns a guard to each transition t such that $Type[G(t)] = Bool$.
8) $E : A \rightarrow EXPR_V$ is an arc expression function that assigns an arc expression to each arc a such that $Type[E(a)] = C(p)_{MS}$[1],where p is the place connected to the arc a.
9) $I : P \rightarrow EXPR_\theta$ is an initialisation function that assigns an initialisation expression to each place p such that $Type[I(p)] = C(p)_{MS}$.
10) $T_{sub} \subseteq T$ is a set of substitution transitions.
11) $P_{port} \subseteq P$ is a set of port places.
12) $PT : P_{port} \rightarrow IN$,$OUT$, I/O is a port type function that assigns port types to places.

Therefore, a *Hierarchical Coloured Petri Net* is a four-tuple CPN_H = (S, SM, PS, FS):

1) S is a finite set of *modules*. Each module is a *Coloured Petri Net Module* s = ((P^s , T^s , A^s,Σ^s , V^s , C^s , G^s , E^s , I^s), T^s_{sub}, P^s_{port} , PT^s). It is required that

[1]MS refers to "multiset".

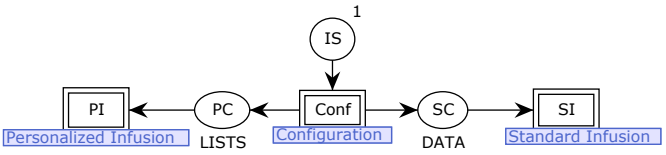

Fig. 1. `Insulin Infusion Pump` module of the reference model

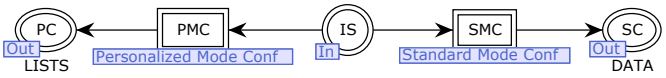

Fig. 2. `Configuration` submodule of the reference model

$(P^{s_i} \cup T^{s_i}) \cap (P^{s_j} \cup T^{s_j}) = \theta$ for all s_i, $s_j \in S$ such that $i \neq j$.

2) $SM : T_{sub} \rightarrow S$ is a *submodule* function that assigns a submodule to each substitution transition. It is required that the *module hierarchy* is acyclic.
3) PS is a port-socket relation function that assigns a *port-socket relation* $PS(t) \subseteq P_{sock}(t) \times P_{port}^{SM(t)}$ to each substitution transition t. It is required that $PT(p) = PT(p'), C(p) = C(p')$ and $I(p)\langle\rangle$ for all $(p, p') \in PS(t)$ and all $t \in T_{sub}$.
4) $FS \subseteq 2^P$ is a family of non-empty *fusion sets* such that $C(p) = C(p')$ and $I(p)\langle\rangle = I(p')\langle\rangle$ for all p, $p' \in fs$ and all $fs \in FS$.

IV. Reference Model

The reference model consists of a CPN model, considering color sets to represent the insulin dosages and the pump cartridge. The use of integer color sets enables one to apply the model checking technique, reducing the state space.

The reference model contains five modules. The composition of the modules represents the entire insulin infusion pump control system. Fig. 1 illustrates the main module (`Insulin Infusion Pump`). This is the start point of the intermediate modular refinement (or decomposition). Each module relates to a step of intermediate modular refinement. The module is divided into two parts: pump configuration and insulin infusion. In order to enable the insulin infusion, it is necessary to configure the pump. More specifically, the module `Insulin Infusion Pump` contains the substitution transitions `Conf`, `PI`, and `SI` $\in T_{sub}$.

The substitution transitions `Conf` relates to the submodule `Configuration` (i.e., $SM(Conf) = Configuration$). This submodule defines the profile that guides the system operations: standard or personalized. Fig. 2 presents the `Configuration` submodule. The configuration of values of the insulin dosages changes depending on the profile.

The substitution transition `SMC` is related to the submodule `Standard Mode Conf` (i.e., $SM(SMC) = StandardModeConf$). In this case, the insulin dosages for basal, bolus, and bolus corrective have standard values, implying in the usage of the constant `CONF` to define the marking of the place `DC` (Fig. 3). The cartridge capacity is

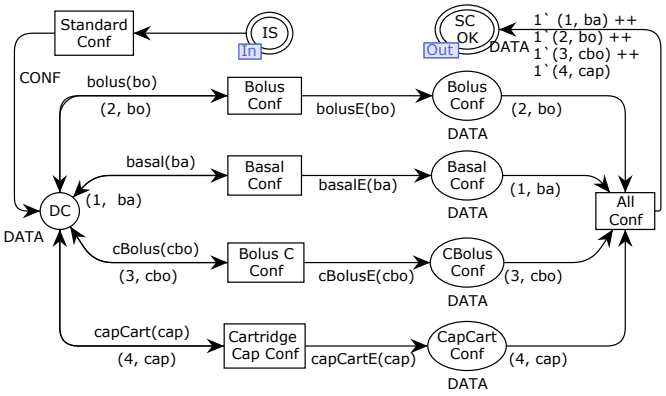

Fig. 3. `Standard Mode Conf` submodule of the reference model

also configured using this constant. Additionally, these values must respect upper and lower bounds. The functions `bolus`, `basal`, `cBolus`, and `capCart` verify these bounds. The system is able to inject the insulin when the transition `All Conf` fires.

Using the personalized mode, it is necessary to define different values for different dosages at a daily basis. A file loads the configuration dosages to define the infusion mode values as list data structure. These values depend on the medical prescription for a specific case. The remaining of the configuration follows the same approach of the standard mode. In the end of the configuration, two lists represent the basal and bolus dosages (`listBa` and `listDC`). The lists are the inputs for the insulin infusion.

The second part of the insulin infusion pump control system modeling is the insulin infusion. It contains the specifications for the standard and personalized modes. Fig. 4 presents the submodule for the standard infusion. The first step is to record the configuration data. The system notifies that the pump is executing and that the cartridge is loaded. Afterward, it is possible to select a specific mode by firing one of the transitions `Adm Basal`, `Adm CBolus`, and `Adm Bolus`. The user can choose one or more types of insulin during the system execution. If the sum of all dosages does not exceed a safety limit, the system applies an unique dosage composed of all types of insulin. Otherwise, it applies just the maximum quantity allowed. The transition `Apply Insulin` represents the event that triggers the insulin infusion. Four 2-tuple represent the basal insulin, corrective bolus insulin, bolus insulin and cartridge's capacity.

Once the system verifies the compliance with all preconditions, it applies the insulin and updates the cartridge capacity. When the cartridge is empty, the system transits to a state that indicates that it is necessary to recharge the pump. The system also reaches this state when the cartridge's capacity is below the current insulin dosage being applied. Considering both situations (`cap`[2] = 0 or below the minimum required dosage), the pump transits from the partial state of executing

[2]CPN variable that represents the cartridge's capacity.

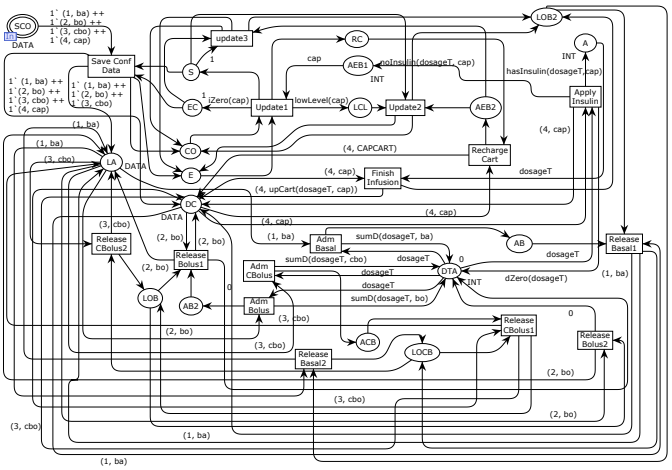

Fig. 4. `Standard Infusion` submodule of the reference model

(removing an uncoloured token from the place `E`) to the partial state of stopping (marking the place `S` with an uncoloured token). When the user recharges the pump, the system returns to the partial state of executing.

The personalized insulin infusion mode behaves similarly to the standard mode. However, instead of using constant values of dosages, it allows the user to personalize the desired dosages from a file. It initially uses list data structures to represent the insulin dosages and converts them to the original tuple format. It removes the first basal dosage from a list named `listBa` (daily basal insulin dosages) and inserts it in the head of the list named `listDC` (bolus and corrective bolus dosages, and the cartridge's total capacity). A specific place is responsible for recording the remaining basal dosages. When necessary, the system applies these basal dosages. The infusion pump representation remains almost equal to the standard mode. The most significant change regards to the usage of the remaining basal dosages. It repeats the infusion until there exist tokens of basal dosages representing the remaining of the insulin. Finally, the specification of the process of recharging the cartridge is also almost equal to the standard mode. In this case, the personalized mode contains new places and transitions to control the following situations:

- the system applied the current basal dosage, and it is necessary to apply the remaining dosages near in the future;
- it is necessary to apply the current basal dosage, and there are remaining dosages recorded; and
- there is no current basal dosage, nether near in the future.

Table I presents the input parameters of the reference model along with descriptions and examples of values. The values illustrate how an insulin infusion pump manufacturer can set the reference model to represent a specific product. The parameters `BASAL`, `BOLUS`, `CBOLUS`, `CAPCART`, `UPPERDOSELIMIT`, `LOWERDOSELIMIT`, `CAPCARTLIMIT`, and `INFUSIONLIMIT` use integer values to configure the pump, whereas the parameter `FILE` receives a string that represents the name of a file containing basal insulin dosage values. This model refinement is composed of integer data types to reduce the state space explosion problem and simplify the execution of the model checking technique.

TABLE I
INPUT PARAMETERS OF THE REFERENCE MODEL

Parameter	Description	Hypothetical Value
BASAL	The dose of basal insulin	3
BOLUS	Bolus insulin dosage	3
CBOLUS	Corrective bolus insulin dosage	3
CAPCART	Cartridge capacity	9
UPPERDOSELIMIT	Upper limit of any type of insulin	3
LOWERDOSELIMIT	Lower limit of any type of insulin	1
CAPCARTLIMIT	Cartridge's capacity limit	9
INFUSIONLIMIT	Insulin infusion limit	6
FILE	File name with custom basal dosage values	"BasalV1MR.txt"

V. Evaluation of the Reference Model

This section describes a case study on the commercial infusion pump system named ACCU- CHECK Spirit version 2.XX [1]. The case study extends the reference model to evaluate it by means of the model checking technique. Additionally, it is useful to demonstrate how manufacturers can reuse the model during the design and certification process of insulin infusion pump control systems. Therefore, the configuration of the reference model parameters follows the technical requirements of ACCU-CHEK Spirit version 2.XX.

The verification of the reference model comprises the parameters' configuration in order to apply the model checking technique. Considering that it concerns with system's properties, the configuration of the model is according to the technical requirements of the ACCU-CHEK Spirit. Table II presents the configuration defined for the verification activity (initial marking of the model). The file named `BasalV1AC.txt` contains 24 hypothetical values of basal dosages for the personalized mode.

TABLE II
INPUT PARAMETERS OF THE REFERENCE MODEL FOR THE ACCU-CHEK SPIRIT VERSION 2.XX

Parameter Name	Integer/String Value
BASAL	10
BOLUS	8
CBOLUS	6
CAPCART	315
UPPERDOSELIMIT	25
LOWERDOSELIMIT	1
CAPCARTLIMIT	315
INFUSIONLIMIT	25
FILE	"BasalV1AC.txt"

This activity consists of the verification of the two safety properties of insulin infusion pump control systems described in Table III. The specification of the ACCU-CHEK Spirit version 2.XX considers these properties. Given that the model checking technique using CPN is based on the Computation tree logic (CTL), the functions available on the ASK/CTL

library of CPN/Tools represent the specifics of these properties. ASK/CTL is a CPN-based extension of the CTL temporal modal logic.

TABLE III
SAMPLE OF SAFETY PROPERTIES OF INSULIN INFUSION PUMP CONTROL SYSTEMS

ID	Property	*ASK-CTL* Formula
1	If the cartridge level equals 0, then the cartridge status should become "EMPTY" and the state of the pump should be "STOP".	OR(NOT(EV(NF("a_",cartZeroed))), EV(AND(NF("c_",bombStop), NF("d_",cartEmpty))))
2	If the cartridge level is less than the administered insulin dosage, then the state of the pump should be "STOP".	OR(NOT(EV(NF("a_",lowCartLevel))), EV(NF("c_",bombStop)))

The description of both properties suggests the usage of a composed formula using the implication connective. Considering that ASK/CTL does not provide this connective, the formula represents the following logical equivalence: $\phi \rightarrow \psi \equiv \neg\phi \vee \psi$, where $\rightarrow$, $\neg$ and $\vee$ are symbols that represent the logical connective implication, negation and disjunction, respectively.

A. Model Checking Property 1

The first property (Table III, ID 1) defines that when the level of the cartridge is zero (i.e., the pump is empty), the system must be stopped. Therefore, the formula contains the following logical propositions:

- $\phi =$ the level of the cartridge is 0;
- $\psi_1 =$ the state of the cartridge is `EMPTY`; and
- $\psi_2 =$ the state of the pump is `STOP`.

Thus, it implies in the formula $\phi \rightarrow (\psi_1 \wedge \psi_2)$. When applying the logical equivalence, $\neg\phi \vee (\psi_1 \wedge \psi_2)$. The ASK/CTL formula simply translates this logical equivalence. Afterward, the model checker verified that the model complies with this property (see Fig. 5). The functions `cartZeroed`, `cartEmpty`, and `bombStop` represent the propositions ϕ, ψ_1, and ψ_2, respectively. They use the place instances named `Standard_Infusion'DC`, `Standard_Infusion'S` and `Standard_Infusion'EC` to verify the first property considering the standard mode of insulin infusion. For the personalized mode, the unique difference is the name of each place instance. It is possible to observe that the reference model is according to the first usual property of insulin infusion pump control systems for the standard and personalized modes.

The format of the ASK-CTL formula differs from the logical equivalence $\neg\phi \vee (\psi_1 \wedge \psi_2)$ due to the `EV` and `NF` operators. The first one aims to specify that the formula considers all paths of the state space in the future, whereas the second one specifies that it is a node formula. The function `eval_node` from ASK-CTL runs the model checking algorithm. The

```
val cartZeroed = fn : Node -> bool
val bombStop = fn : Node -> bool
val cartEmpty = fn : Node -> bool
val myASKCTLformula =
  OR
    (NOT (FORALL_UNTIL (TT,NF ("a_",fn))),
     FORALL_UNTIL (TT,NOT (OR (NOT (NF ("c_",fn)),NOT (NF ("d_",fn)))))) : A
val it = true : bool

fun cartZeroed b = Mark.Standard_Infusion'DC 1 b = 1`(4,0);
fun bombStop r = Mark.Standard_Infusion'S 1 r = 1`();
fun cartEmpty s = Mark.Standard_Infusion'EC 1 s = 1`();

val myASKCTLformula = OR(
                          NOT(
                              EV(NF("a_",cartZeroed))),
                              EV( AND( NF("c_",bombStop), NF("d_",cartEmpty) ) )

                         );
eval_node myASKCTLformula InitNode;
```

(a) Property 1 for default application mode

```
val cartZeroed = fn : Node -> bool
val bombStop = fn : Node -> bool
val cartEmpty = fn : Node -> bool
val myASKCTLformula =
  OR
    (NOT (FORALL_UNTIL (TT,NF ("a_",fn))),
     FORALL_UNTIL (TT,NOT (OR (NOT (NF ("c_",fn)),NOT (NF ("d_",fn)))))) : A
val it = true : bool

fun cartZeroed b = Mark.Personalized_Infusion'DC 1 b = 1`(4,0);
fun bombStop r = Mark.Personalized_Infusion'S 1 r = 1`();
fun cartEmpty s = Mark.Personalized_Infusion'EC 1 s = 1`();

val myASKCTLformula = OR(
                          NOT(
                              EV(NF("a_",cartZeroed))),
                              EV( AND( NF("c_",bombStop), NF("d_",cartEmpty) ) )

                         );
eval_node myASKCTLformula InitNode;
```

(b) Property 1 for personalized application mode

Fig. 5. Model checking property 1 for the standard and personalized modes

Boolean value `true` means that the algorithm confirms this property for the model.

B. Model Checking Property 2

The second property (Table III, ID 2) defines that the pump cannot run when the insulin dosage is greater than the level of the cartridge. The formalization of this property followed the same approach as the first property. Firstly, it considers some propositions:

- $\phi =$ the level of the cartridge is less than the insulin dosage being applied; and
- $\psi =$ the state of the pump is `STOP`.

It resulted in the formula $\phi \rightarrow \psi$. Once again, when applying the logical equivalence: $\neg\phi \vee \psi$. Afterward, this formula was simply translated to its respective ASK-CTL formula. Fig. 6 presents the model checking results. The propositions ϕ and ψ are equivalent to the functions `lowCartLevel` and `bombStop`, respectively. These functions relate with the place instances `Standard_Infusion'AEB` and `Standard_Infusion'S` to run the model ckecking algorithm considering the standard mode. For the personalized mode, the unique difference is the name of each place instance.

Similarly to the first property, it contains the `EV` and `NF` ASK-CTL operators. It enabled the specification of the formula, whereas the execution of the model checking algorithm was possible using the ASK-CTL function `eval_node`.

```
val lowCartLevel = fn : Node -> bool
val bombStop = fn : Node -> bool
val myASKCTLformula =
  OR (NOT (FORALL_UNTIL (TT,NF (''a_'',fn))),FORALL_UNTIL (TT,NF (''c_'',fn))) : A
val it = true : bool

fun lowCartLevel i = Mark.Standard_Infusion'AEB1 1 i <> [];
fun bombStop r = Mark.Standard_Infusion'S 1 r = 1`();


val myASKCTLformula = OR(
                          NOT(EV(NF(''a_'',lowCartLevel))),
                              EV(NF(''c_'',bombStop))
                         );
eval_node myASKCTLformula InitNode;
```

(a) Property 2 for default application mode

```
val lowCartLevel = fn : Node -> bool
val bombStop = fn : Node -> bool
val myASKCTLformula =
  OR (NOT (FORALL_UNTIL (TT,NF (''a_'',fn))),FORALL_UNTIL (TT,NF (''c_'',fn))) : A
val it = true : bool

fun lowCartLevel i = Mark.Personalized_Infusion'AEB1 1 i <> [];
fun bombStop r = Mark.Personalized_Infusion'S 1 r = 1`();


val myASKCTLformula = OR(
                          NOT(EV(NF(''a_'',lowCartLevel))),
                              EV(NF(''c_'',bombStop))
                         );
eval_node myASKCTLformula InitNode;
```

(b) Property 2 for personalized application mode

Fig. 6. Model checking property 2 for the standard and personalized modes

Once again, the algorithm presented the Boolean value `true`, meaning that the model is according to the second property of insulin infusion pump control systems.

VI. Conclusions and Future Work

This paper presented the formal specification of a reference model of insulin infusion pump control systems aiming to assist the certification process. Additionally, it described a case study on the commercial insulin infusion pump control system named ACCU-CHEK Spirit version 2.XX. A reference model is relevant in the context of certification because these are safety-critical systems handled to treat patients with diabetes.

The reference model was specified using the CPN modeling language (along with the CPN/Tools software), and it includes the essential features that ensures the correct functioning of an insulin infusion pump control system. CPN enabled the specification considering parameters, modules, simulation, and verification.

The model presented in this paper is relevant because it has characteristics not completely considered by existing solutions. These include, for example, parameters (model instantiated for different systems), modules (management of specification complexity), and execution (simulation of real behaviors).

The case study was useful to extend the reference model conducting verifications. The verification of two safety properties of the system for the control of insulin infusion was considered. In addition, the case study demonstrates that manufacturers can reuse the model during the certification process of insulin infusion pump control systems.

The CPN model is a project artifact used to conduct verification activities aiming to increase confidence on insulin infusion pump control systems. Additionally, it is useful to carry out quality assessment of system properties, such as safety, during a certification process. For instance, once a regulatory agency issues a recall of a system, the reference model may assist manufacturers to evaluate the current specification by comparing the physical solution with the formal specification (simulated solution). This can decrease cost and development time by enabling the prompt correction of the defects/failures based on the model of the system under evaluation. Therefore, the main contribution of this paper is the reference model of insulin infusion pump control systems in a way that manufacturers can reuse it during a certification process (along with the case study).

As an example of future work, we envision to extend the reference model of insulin infusion pump control systems to represent time features, and a closed-loop insulin infusion pump. The closed-loop version will include the sensing of glucose, monitoring, and treatment of patients with diabetes. Additionally, it is envisioned to define a more generic model to represent any type of infusion pump system, such as analgesic infusion pumps systems.

References

[1] Accu-chek spirit insulin pump system: Pump user guide.

[2] Fda: Medical device classification procedures, Revised April 2016.

[3] Guidance for industry and fda staff: Infusion pumps total product life cycle.

[4] Hatcliff, J., Larson, B., Carpenter, T., Jones, P., Zhang, Y. and Jorgens, J. The open pca pump project: An exemplar open source medical device as a community resource, in: Medical Cyber Physical Systems Workshop, 2018.

[5] Imagawa, K., Mizukami, Y. and Miyazaki, S. Regulatory convergence of medical devices: a case study using iso and iec standards, Expert Review of Medical Devices (2018) 497504.

[6] FDA Infusion 2010. US FDA Infusion Pump Improvement Initiative. (April 2010).

[7] Jensen, K. and Kristensen, L. M. Colored petri nets: a graphical language for formal modeling and validation of concurrent systems, Communications Of The ACM (2015) 6170.

[8] Mertz, L. Automated insulin delivery: Taking the guesswork out of diabetes management, IEEE Pulse (2018) 8-9.

[9] Muangprathub, J. and Boonjing, V. Online thai medical diagnostic system using case-based reasoning, in: 2014 International Computer Science And Engineering Conference, 2014, pp. 114-117.

[10] Niezen, G. and Eslambolchilar P. A human operator model for medical device interaction using behavior-based hybrid automata, IEEE Transactions On Human-machine Systems (2016) 291-302.

[11] Rathore, H., Wenzel, L., Al-Ali, A. K., Mohamed, A., Du, X. and Guizani, M. Multi-layer perceptron model on chip for secure diabetic treatment, IEEE Access (2018) 44718-44730.

[12] Sharanya, G., Abhishek, C. S. and Reddy, K. M. Health monitoring device, in: 2017 2nd International Conference On Communication And Electronics Systems, 2017, pp. 668-671.

[13] Silva, L. C., Almeida, H. O., Perkusich, A. and Perkusich, M. A model-based approach to support validation of medical cyber-physical systems, Sensors (2015) 27625-27670.

[14] Sjaaheim, H., Albert, B., Setchi, R., Noyvirt, A. and Strisland, F. A portable medical system for the early diagnosis and treatment of traumatic brain injury, in: 2014 IEEE International Conference On Systems, Man, And Cybernetics, 2014, pp. 2529-2534.

[15] Sobrinho, A., da Silva, L.D., Perkusich, A., Cunha, P., Cordeiro, T.D. and Lima, A.M.N. Formal modeling of biomedical signal acquisition systems: source of evidence for certification. Software and Systems Modeling, pp.1-19. 2017.

[16] Zhang, Y., Jones, P. L. and Jetley, R. A hazard analysis for a generic insulin infusion pump, Journal Of Diabetes Science And Technology (2010) 263-283.