applied sciences

MDPI

Article

Coloured Petri Nets-Based Modeling and Validation of Insulin Infusion Pump Systems

Tássio Fernandes Costa [1,†], Álvaro Sobrinho [1,2,*,†], Lenardo Chaves e Silva [3,†], Leandro Dias da Silva [1,†] and Angelo Perkusich [4,†]

[1] Computing Institute, Federal University of Alagoas, Maceió 57072-900, Brazil; tfc@ic.ufal.br (T.F.C.); leandrodias@ic.ufal.br (L.D.d.S.)
[2] Computer Science, Federal University of the Agreste of Pernambuco, Garanhuns 55292-270, Brazil
[3] Computer Department, Federal Rural University of the Semiarid, Mossoró 59625-900, Brazil; lenardo@ufersa.edu.br
[4] Virtus Research, Development and Innovation Center, Federal University of Campina Grande, Campina Grande 58428-830, Brazil; perkusic@dee.ufcg.edu.br
* Correspondence: alvaro.alvares@ufape.edu.br
† These authors contributed equally to this work.

Abstract: Safety and effectiveness are crucial quality attributes for insulin infusion pump systems. Therefore, regulatory agencies require the quality evaluation and approval of such systems before the market to decrease the risk of harm, motivating the usage of a formal Model-Based Approach (MBA) to improve quality. Nevertheless, using a formal MBA increases costs and development time because it requires expert knowledge and thorough analyses of behaviors. We aim to assist the quality evaluation of such systems in a cost-effective and time-efficient manner, providing re-usable project artifacts by applying our proposed approach (named MBA with CPN—*MBA/CPN*). We defined a Coloured Petri nets MBA and a case study on a commercial insulin infusion pump system to verify and validate a reference model (as a component of *MBA/CPN*), describing quality assessment scenarios. We also conducted an empirical evaluation to verify the productivity and reusability of modelers when using the reference model. Such a model is relevant to reason about behaviors and quality evaluation of such concurrent and complex systems. During the empirical evaluation, using the reference model, 66.7% of the 12 interviewed modelers stated no effort, while 8.3% stated low effort, 16.7% medium effort, and 8.3% considerable effort. Based on the modelers' knowledge, we implemented a web-based application to assist them in re-using our proposed approach, enabling simulation-based training. Although a reduced number of modelers experimented with our approach, such an evaluation provided insights to improve the *MBA/CPN*. Given the empirical evaluation and the case study results, *MBA/CPN* showed to be relevant to assess the quality of insulin infusion pump systems.

Keywords: simulation; coloured Petri nets; modeling

Citation: Fernandes Costa, T.; Sobrinho, Á.; Chaves e Silva, L.; da Silva, L.D.; Perkusich, A. Coloured Petri Nets-Based Modeling and Validation of Insulin Infusion Pump Systems. *Appl. Sci.* **2022**, *12*, 1475. https://doi.org/10.3390/app12031475

Academic Editors: João Paulo Barros and Luis Gomes

Received: 28 December 2021
Accepted: 26 January 2022
Published: 29 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

The treatment of diabetes usually requires the usage of insulin infusion pump systems. Hardware components compose the infusion pump that simulates the pancreas' behavior, while software components are related to the embedded software used to control the pump [1]. As a safety-critical system, manufacturers should analyze the behaviors of insulin infusion pump systems to provide, at least, the minimum required guarantee of correctness [2].

Formal methods [3] play a significant role in verifying systems requirements and guaranteeing developed systems' correctness, reliability, and safety. In the medical domain, regulatory agencies, such as the United States Food and Drug Administration (FDA), need effective means to evaluate the devices to certify the developed systems and assure each system's safe behavior [4,5]. Regulatory agencies are striving for rigorous techniques and methods to provide safety assurance. Formal methods can help develop dependable,

safe, and secure systems and provide sound evidence for the required features to certify dependable medical systems.

In this article, we used Coloured Petri Nets (CPN) formal method [6] for modeling and validation of insulin infusion pump systems. CPN is a formal graphical language for modeling and validating systems in which concurrency plays a major role. CPNs have been successfully applied for the formal specification, analysis, and verification of different communication protocols, embedded systems, and data networks, among other systems.

Handling formal methods usually requires expert knowledge and increased costs and development time, which is one of the main motivations to conduct this study. Reusable reference models have the potential to reduce the impact of such costs. In a previous research work [7], a reference model to assist the certification of biomedical systems using CPN has been introduced, by specifying hardware and software components and applying simulations and the model checking technique. In another similar previous research work [8], a reference model to assist the certification of insulin infusion pump systems using CPN has also been introduced. In such research works, we showed the relevance of using a CPN reference model to generate evidence for certification. A limitation of these previous study is the lack of evaluation of the model considering reusability and productivity, reducing confidence in the capability of decreasing costs and development time in practice.

Regulatory government agencies require manufacturers to demonstrate that insulin infusion pump systems do not put users in hazard situations [9]. In 2010, the FDA released the infusion pump improvement initiative [10], concerning problems such as software defects, inadequate Graphical User Interfaces (GUI), and mechanical or electrical failures. As a result, the FDA released in 2014 a guidance for industry and FDA staff to follow during the pump's life cycle [11]. According to the same guideline, the most commonly reported problems of infusion pumps include software error, human factors, broken components, battery failure, alarm failure, and over infusion and under infusion. Some of the reported problems are related to the design activity, while others are described by manufacturers as unknown problems, making it difficult to achieve solutions.

There still exists a large number of recalls reported by regulatory government agencies regarding insulin infusion pump systems [12]. Gao et al. [13] state that from the 70 infusion pump recalls released by FDA between 2001 and 2017, 17 recalls were caused by software failures. To address this problem, regulatory agencies have increased the surveillance stringency when manufacturers submit the system under development to the certification process. Manufacturers usually present a set of quality evidence about systems based on prescriptive standards (e.g., ISO 14971) and quality attributes. Besides the requirements required by regulatory agencies, a large number of recalls also motivates the proposal and usage of a formal Model-Based Approach (MBA) to improve quality. However, it is known that the usage of such approaches usually increases costs and development time.

To address the problem of increased costs and development time, we present an MBA of insulin infusion pump systems (named MBA with CPN—*MBA/CPN*) focusing on CPN reference models as modeling artifacts to increase confidence in system behaviors and provide quality assessments. The specification of the pump system is linked to the proposed CPN model by the definition of CPN modules that represent critical parts of such a system. We describe a case study on a commercial system, i.e., the ACCU-CHEK Spirit [14], to evaluate a reference model, as part of *MBA/CPN*, by simulations and the model checking technique, describing quality assessment scenarios. The case study is also relevant to show how manufacturers can re-use the *MBA/CPN* during a certification process. Therefore, this research faces challenges such as (1) the integration of requirements specification and assurance cases to provide re-use and (2) the re-use of a CPN reference model in a time-effective and cost-effective manner.

The proposed approach may benefit the certification process by the re-use of reference models, along with an assurance cases-based requirements specification with the Goal-Structuring Notation (GSN), in the initial phases of the developing process. The refer-

ence model was carefully validated to decrease the possible negative impacts of a manual specification. We also conducted an empirical evaluation with 12 interviewed modelers to evaluate the *MBA/CPN*. Although a reduced number of modelers experimented with our approach, such an evaluation provided insights to improve the *MBA/CPN*. This study extends the results of our previous research works [7,8], consisting of a new main contribution: an MBA to assess the quality of insulin infusion pump systems in a cost-effective and time-efficient manner. In our previous study [7,8], we did not address (1) the integration of requirements specification and assurance cases to provide re-use (goal-oriented requirements engineering) and (2) the re-use of a CPN reference model in a time-effective and cost-effective manner. Thus, we address the following main Research Question (RQ): is the *MBA/CPN* able to provide a cost-effective and time-efficient quality assessment of insulin infusion pump systems? The quality assessment term stands for evaluating quality attributes such as safety and effectiveness, as required by regulatory agencies.

Existing related works (described in Section 3) do not provide a generic, parametric, and timed insulin infusion pump systems model to assist manufacturers in conducting detailed analyses (e.g., infusion control and common recalls) during development and certification. The *MBA/CPN* addresses this limitation, considering an executable, generic, parametric, and timed insulin infusion pump systems model, that includes infusion control and considers the FDA guidelines and reported recalls. The novelty of our proposed approach also relates to the usage of GSN assurance cases during a goal-oriented requirements engineering and the simulation-based training (not requiring the CPN/Tools) of modelers to improve reusability.

Therefore, although it is a consensus that the proposal and the use of formal methods to validate the behavior of safety-critical systems is valuable, our proposal also works as a guide, along with reusable project artifacts, for developers of insulin infusion pump systems. We aim to assist them to improve the quality of systems using an assurance case-based specification and CPN reference models. To answer our research question, we present the case study results and the empirical evaluation of models.

The article is structured as follows: Section 2 presents the fundamental concepts of GSN and CPN. Sections 3 and 4 describe related works and the *MBA/CPN*, respectively. Section 5 describes the quality assessment scenarios of insulin infusion pump systems using simulations and model checking. Section 6 presents an empirical evaluation of the *MBA/CPN* and Section 7 presents a web-based application implemented based on such evaluation to improve productivity and reusability of models. Section 8 discusses the results, while Section 9 concludes the study and presents future research directions.

2. Preliminaries

2.1. Goal-Structuring Notation

GSN is a standard used to represent assurance cases graphically [15]. Figure 1 illustrates the main components of GSN: `Goal` (rectangle), `Solution` (circle), `Strategy` (parallelogram), `Context` (rectangle with rounded edges), `Justification` (ellipse), `Undefined` (diamond), `SupportedBy`, and `InContextOf`. The `SupportedBy` and `InContextOf` components are used to connect goals to solutions and perform context associations, respectively. Goals are useful to provide claims about systems quality, while solutions are linked to evidence of such a claim. `Strategy`, `Context`, and `Justification` are complementary components to improve the clarity and completeness of the assurance case specification. The `Undefined` component provides a way to illustrate parts of the assurance case that are still under specification.

Besides, the GSN includes modular components: `Away Goal`, `Module`, `Contract`, `Away Solution`, `Away Context`, and `Public Indicator`. The modular components assist the modularization of the assurance case, aiming to provide more compact representations, simplify the maintenance, and improve readability. The usage of modules enables the grouping of augmentations to clearly support the claims about systems quality. The modular GSN is one of the fundamental concepts used to define the *MBA/CPN*.

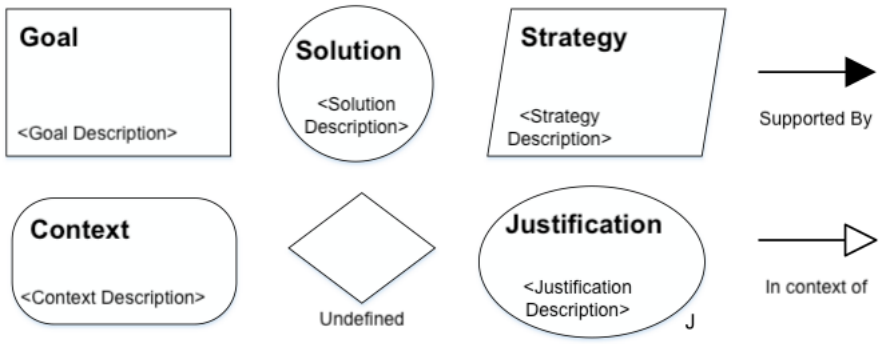

Figure 1. The main components of GSN: `Goal`, `Solution`, `Strategy`, `Context`, `Justification`, `Undefined`, `SupportedBy`, and `InContextOf`.

2.2. Coloured Petri Nets and Access/CPN

We present a partial formalization of a CPN (elements and structure) because it is linked to different requirements in an assurance case-based specification using GSN. Thus, a CPN model is composed of elements such as places, transitions, data types, and hierarchy. A coloured Petri net module is a tuple $CPN_M = (P, T, A, \Sigma, V, C, G, E, I, T_{sub}, P_{port}, PT)$:

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 $Type[E(a)] = C(p)_{MS}$, where p is the place connected and MS refers to "multiset". 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 $s = ((P^s, T^s, A^s, \Sigma^s, V^s, C^s, G^s, E^s, I^s), T^s_{sub}, P^s_{port}, PT^s)$. It is required that $(P^{s_i} \cup T^{s_i}) \cap (P^{s_j} \cup T^{s_j}) = \theta$ 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, requiring that the module hierarchy is acyclic.
3. PS is a port-socket relation function that assigns a port-socket relation $PS(t) \subseteq P_{sock}(t) \times P^{SM(t)}_{port}$ to each substitution transition t, requiring 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 $fs \in FS$.

The CPN/Tools software is used to model, conduct simulations, and run CPN/ML codes. However, it is possible to handle CPN models without requiring the usage of a GUI. The Access/CPN framework enables users to handle model components, simulations, and CPN/ML code using the Java programming language [16]. For instance, such a framework provides methods to fire the enabled transitions and obtain tokens related to specific places. This type of functionality is relevant, for instance, to embed the CPN models in software applications for training modelers on how to re-use the MBA/CPN.

3. Related Works

The application of MBAs is a trend when developing complex safety-critical systems [17]. For example, Mian et al. [18] present a framework to translate a state machine-based error model to a fault-tree model representation. Entezari-Maleki et al. [19] describe a timed CPN to evaluate the web service composition in multi-cloud environments, verifying the MBA accuracy using two application scenarios. Valls et al. [20] propose an approach composed of parametric models for designing adaptive Cyber-Physical Systems (CPS). We use a similar idea, reusing parametric insulin infusion pump systems models to assess quality attributes.

Montecchi et al. [21] formally define a concept of model templates to assist the definition of libraries of generic submodels and model composition. The authors apply the proposed MBA using a case study on a large-scale distributed system. Kanoun and Ortalo-Borrel [22] proposed a modular approach for modeling the dependability fault-tolerant systems using generalized stochastic Petri net for submodels and model composition. Nencioni et al. [23] present a modular approach for quantitative assessment of the properties of software-defined networking considering failure correlation. Rabah and Kanoun [24] provide an MBA for evaluating performability measures of multipurpose, multiprocessor systems using architectural models, service-level models, and maintenance policy models. Similarly, we use the concept of reference models as templates, improving the state-of-the-art by integrating the models with the requirements specification based on assurance cases and evaluating the level of understandability and adaptability of reusable models.

Silva et al. [25] describe clinical scenarios to evaluate an approach to assist the modeling and validation of medical CPS. Simulink block diagrams represent the behaviors of the system. However, the modeling does not consider the specification of time constraints and precise insulin infusion control. Besides, the FDA is also concerned with the quality attributes of infusion pumps. The generic infusion pump project is an example of an FDA initiative to increase confidence in insulin infusion pump systems. For example, the project addresses the risk analysis by a generic architectural specification [26], being a high-level representation that does not enable the execution of a model instance considering time constraints and formal verification. Hatcliff et al. [27] conducted the open patient-controlled analgesia pump project to provide artifacts such as use cases, testing and simulation infrastructure, risk management artifacts, and assurance cases.

However, there is no executable, generic, parametric, and timed insulin infusion pump systems model to assist manufacturers in conducting detailed analyses (e.g., infusion control and common recalls) during development and certification. The *MBA/CPN* addresses this limitation, considering an executable, generic, parametric, and timed insulin infusion pump systems model, that includes infusion control and considers the FDA guidelines [11] and reported recalls [13]. As another improvement, our proposed approach addresses a goal-oriented requirements engineering based on GSN assurance cases. These characteristics are relevant, e.g., to assist the quality assessment of insulin infusion pump systems under development or identify certified systems problems. Thus, we improve the state-of-the-art by defining an MBA that provides the following characteristics: (1) definition of an eXtensive Markup Language (XML)-based standard to apply GSN assurance cases in the requirements engineering (earlier in the development process); (2) guidelines for definition and availability of parametric, timed, and executable modules of an insulin infusion pump systems reference model for re-use during the development process (and integration with GSN assurance cases); (3) guidelines for the assessment of quality attributes of insulin infusion pump systems based on the re-use of models; and (4) definition of a reusable, parametric, timed, and executable insulin infusion pump systems reference model.

4. Model-Based Approach

Figure 2 illustrates an overview of the *MBA/CPN*, consisting of modeling and analysis steps: hardware and software modularization, reference model definition, reference model instantiation, safety analyses, effectiveness analyses, and abstract test generation. In the

first modeling step, manufacturers elicit CPN hardware and software modules from sources of requirements. In the second step, a reference model is defined by two system refinements. The first one results in a more abstract model that does not consider real colour sets, while the second refinement generates a more detailed model considering real colour sets and time constraints. The system's refinement specification is relevant to conducting analyses using two perspectives: safety and effectiveness. In the third modeling step, manufacturers instantiate the model refinements to analyze the system's behaviors assessing quality. In the first analysis step, the first system's refinement is used to verify safety properties, preventing the state space explosion problem. Afterward, the verified reference model can be reduced (removing hardware components and adjusting initial markings) to apply abstract test generation approaches. The same properties shall be verified again for the reduced model to ensure consistency. Finally, the second system's refinement is used to analyze safety and effectiveness properties in the second analysis step. Manufacturers document insulin infusion pump systems requirements using assurance cases for all steps.

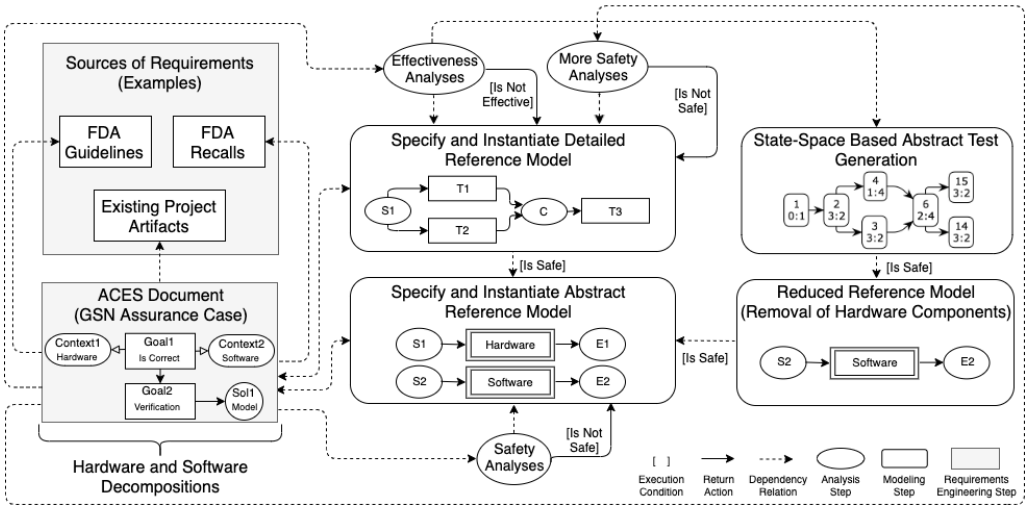

Figure 2. Overview of *MBA/CPN* for quality assessment of insulin infusion pump systems.

We recommend using two different versions of models to reduce the state-space explosion problem, avoiding the application of more complex techniques for state-space reduction. The definition of a less expressive first refinement does not negatively impact the approach because it maintains the desired safety properties, verified using the model checking technique. We assure that the CPN model represents the functionality of a specification according to the FDA guidelines by conducting model simulations. The first refinement of the reference model can be automatically or manually generated from the assurance case-based specification. This article describes manually specified CPN models due to our focus on providing a verified and validated basis for further extensions, as detailed below.

4.1. Hardware and Software Decomposition

The *MBA/CPN* requires manufacturers to develop assurance cases based on the requirements derived from sources such as similar systems, literature reviews, recalls, and guidelines. Claims, evidence, and other assurance cases have specific representations in the *MBA/CPN* using the XML and the GSN. In this article, we proposed and defined the Assurance Case Exchange Standard (ACES) to assist manufacturers and regulatory agencies in specifying and exchanging assurance cases during the development and certification processes. For example, the ACES includes features that enable manufacturers to carry out the traceability of the system requirements. The usage of ACES is the starting point for the application of *MBA/CPN*, and it remains being used during the whole development process due to its connection to the CPN reference models. Thus, instead of documenting and verifying goals by applying a classical approach, such as keeping all objectives satisfied

(i.e., KAOS approach), to conduct the goal-oriented requirements engineering, we propose and use ACES.

The XML specification is the basis for associating all of the remaining steps of the *MBA/CPN*. The ACES considers the main concepts of the requirements engineering process based on modular GSN. An ACES document contains, at least, the graphical notations defined in the GSN specification. Each graphical notation of the GSN elements has a representation in ACES, relating the elements to specific tags and attributes of an ACES document. For example, the evidence element contains an attribute link to enable regulatory agencies to access a design artifact provided by system manufacturers to support an argument. The usage of assurance cases based on a well-defined and independent standard platform to represent and share results obtained by manufacturers with regulatory agencies may improve the design and evaluation of systems.

The start and end of an ACES document is the `<assuranceCase>` tag. To enable the version control, each document has a general identification named `generalId`, along with the version identification (`versionId`) and local identification (`localId`). The `generalId` attribute of the `<assuranceCase>` tag is a unique identifier for all the versions of the ACES document, while the `versionId` and `localId` have different identifiers for each new version to represent the modifications in the document. The beginning of the document also contains specific data about the product under development by means of the `<device>` tag.

Considering that assurance cases contain a set of related arguments about quality attributes of a system (e.g., safety and efficacy), ACES represents them using the `<parentArgument>` and `<childArgument>` tags. The `<parentArgument>` tag composes the body of the `<assuranceCase>` tag and represents the main structure of the assurance case in modular GSN. In contrast, the `<childArgument>` tag represents structures that are parts of the body of the `<parentArgument>` tag. In this case, an ACES document only contains one `<parentArgument>` that may consist of multiple assurance case modules: child arguments relate to specific GSN modules. Each ACES argument contains the `<legalAuthenticator>` tag, aiming to record the author of modifications in the ACES document.

The ACES specification includes the main GSN elements: `Goal`, `Solution`, `Strategy`, `Context`, `Assumption`, `Justification`, `SupportedBy`, and `InContextOf`. In addition, it includes the modular GSN elements: `Away Goal`, `Module`, `Contract`, `Away Solution`, `Away Context`, and `Public Indicator`. ACES structures these elements in its body using the `<group>` tag. This element has the attribute named `type`, that constrains it to group GSN elements of the same type. For instance, to represent a `Goal`, it is necessary to define an ACES tag to represent a specific goal in the body of the `<group>` tag. Each `<group>` tag related to a type of ACES tag is only defined once in the body of each `<parentArgument>` and `<childArgument>` tags.

The `<goal>` tag represents a GSN `Goal` element. All the GSN elements (except relationships) contain, at least, an attribute named `id` and a child tag named `<description>`. The `<goal>` tag can also contain the optional attributes named `public`, `undeveloped`, and `toBeSupportedByContract`. Therefore, goals represent assurance case claims, supported by a set of sub-claims. In ACES, goals can also represent requirements, when the attribute named `requirement` is set to `true`. It enables manufacturers to document quality requirements using ACES. Manufacturers can document product artifacts related to these requirements using GSN solutions. For each goal of a GSN module, a CPN module (XML specification) or a temporal logic formula can be embedded in the ACES document (`<formalDefinition>` tag) to maintain the formal description of the requirements. The `<formalDefinition>` tag may contain the `required` and `provided` tags (interfaces) to enable the specification of module composition.

The `<solution>` tag defines an ACES solution to represent evidence that supports claims. The attribute named `artifact`, when set as `true`, associates the solution with a product artifact. A tag named `externalArtifactUrl` connects a solution to a specific evidence. Defining solutions as product artifacts are relevant to enable the requirements traceability. For reasoning about connections among claims (possibly requirements), manufacturers use `Strategies`. The `<strategy>` tag represents a strategy that contains an additional optional

attribute named undeveloped. Another important characteristic of the requirements engineering considered using ACES is the source of requirements. For assurance cases, the GSN Context element provides information about specific claims, represented in ACES using the <context> tag (with the additional attribute named public). In ACES, GSN Context elements define the source of requirements, setting the attribute named source to true. When this attribute is true, a new tag named <externalSourceUrl> associates the source with the location of the declared source. Defining contexts as requirements source is also relevant to perform the requirements traceability using ACES.

It may also be necessary to improve confidence in the validity of claims and strategies using the GSN Assumption element, defined by the ACES <assumption> tag. Additionally, manufacturers may provide justifications about the definition of claims and strategies. Therefore, the ACES <justification> tag represents a GSN Justification element. In the ACES-based goal-oriented requirements engineering, the <justification> tag enables manufacturers to justify changes in requirements. The justifications add information in the obsolete version of the requirement defined in the ACES document (version control), i.e., a justification is attached to the <goal> tag used to represent the obsolete requirement.

The ACES represents connections between elements with the <relationships> tag, containing at least one child tag. The <relationSupportedBy> tag is a binding notation used to indicate relationships between requirements and project artifacts (evidence), requiring the attributes id, type, and relID. The relID attribute is the identifier that relates GSN elements, respecting the rules defined in the GSN standard. There is also a binding notation used to indicate contextual relationships using the <relationInContextOf> element, and also contains attributes named id, type, and relID. The <relationSupportedBy> and <relationInContextOf> elements are part of the body of the <relationships> tag.

For quality assessment, regulatory agencies can include evaluation results in the ACES document under analysis by the tags <accepted> and <rejected>. The evaluation of ACES documents relates to individual arguments. The body of the tag <rejected> contains a description of the rejection. Manufacturers and regulatory agencies can exchange documents until a final decision about the system's certification is under evaluation. For example, the regulatory agency may ask for specific evidence before the approval of the system. We provide a more detailed description of the ACES specification, as Supplementary Materials, in the *MBA/CPN* repository [28].

As highlighted, assurance cases based on ACES enable manufacturers to carry out the requirements traceability and verification of regulatory requirements. However, there are elements of assurance cases that do not play a fundamental role during these activities. We formally represent ACES documents based on the most relevant assurance cases for requirements traceability. We consider only goals (<goal>) and solutions (<solution>), representing requirements and project artifacts, respectively. An ACES-based assurance case is an oriented graph $T = (V, A)$, where V is a set of nodes related to goals and solutions, and A is a set of edges that connect nodes. Formally, it is defined as a 5-tuple of five elements $ACES = (V_g, V_s, v_r, A, R)$, where

- V_g is a set of nodes defined as goals;
- V_s is a set of nodes defined as solutions such that for all $v_s \in V_s$ its degree is 1;
- $V_g \cup V_s$ is a set of nodes V of an acyclic connected graph T such that $V_g \cap V_s = \theta$;
- $A \subseteq V_g \times V_g \cup V_g \times V_s$ is a set of edges of an acyclic connected graph T;
- $v_r \in V_g$ is a specific node named the root; and
- R is a function $R : V_g \cup V_s \rightarrow 2^D$ (D are descriptions of nodes). Each node v relates to a set $R(v)$ of descriptions (e.g., source of requirements).

Figure 3 illustrates the relationship between manufacturer and regulatory agency using the *MBA/CPN*. The manufacturer and regulator access two repositories: CPN Models Repository and Evidence Repository. They maintain CPN modules of reference models and evidence on the fly, respectively. Manufacturers search for existing system modules under development in the repository to access and compose modules, generating specific systems versions. When there is no reference model (or module) available that fits the

system, it is necessary to generate a new model that may be published in the repository. The regulatory agency evaluates the ACES-based assurance cases by analyzing arguments with linked evidence (published in the repository). The regulator may verify the model of the system available on the fly.

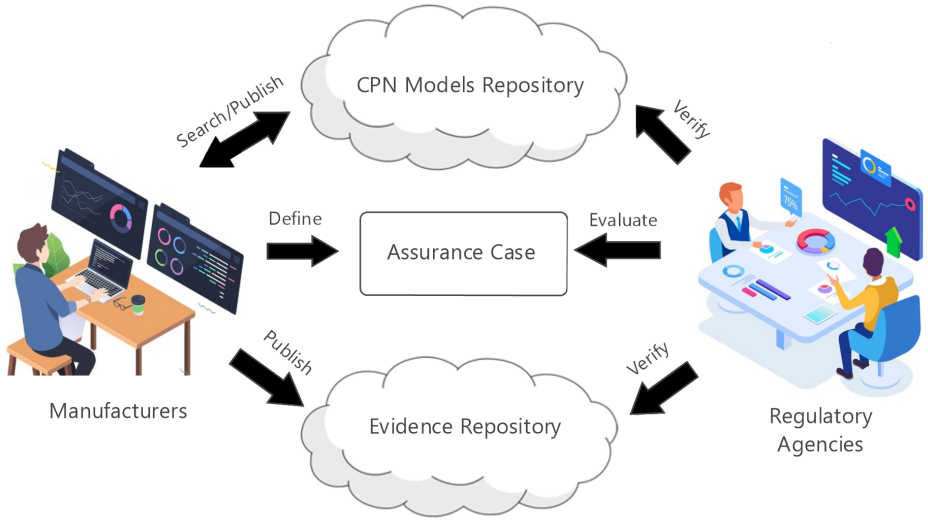

Figure 3. Relationship between manufacturer and regulatory agency.

For example, Figure 4 illustrates the highest level of the assurance case for the insulin infusion pump systems. The GSN element named `SYSTEM-G1` represents a claim about the safety and effectiveness of the system under development. The goal relates to the context of software (`SYSTEM-C1`) and hardware (`SYSTEM-C2`). Strategies support arguments about the process (`SYSTEM-S1`) and product (`SYSTEM-S2`) requirements. The modules named `SYSTEM-M1` and `SYSTEM-M2` contain arguments following these strategies. Figure 5 presents a sample of the ACES document for the assurance case of Figure 4, composed of some of the basic GSN elements.

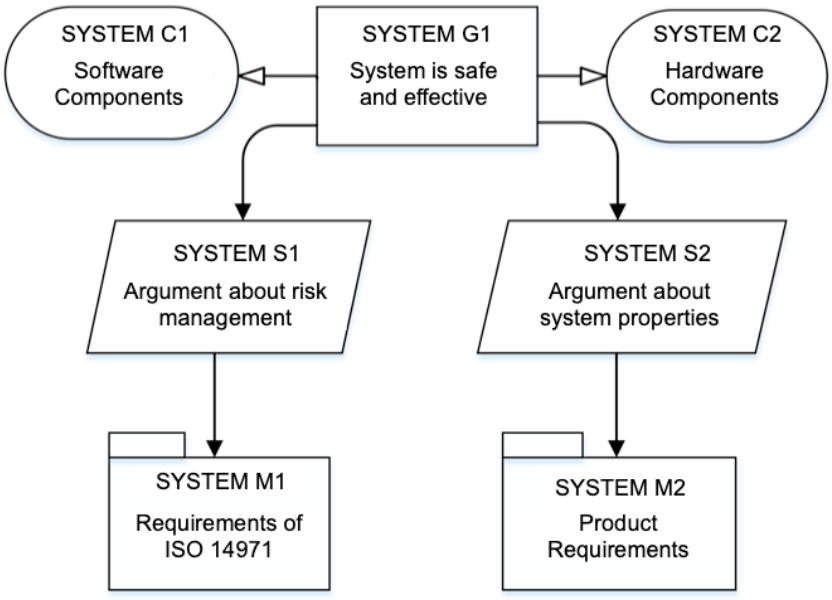

Figure 4. Top level GSN assurance case of insulin infusion pump systems.

```
<parentArgument>
  <legalAuthenticator>
     <time> 25200000 </time>
     <author> John </author>
     <organization> System </organization>
  </legalAuthenticator>
  <group type="goal">
     <goal id="SYSTEM-G1">
        <description>System is safe and effective<
            /description>
        <relationships>
          <relationIncontextOf id="r1" type="
              context" relId="SYSTEM-C1"/>
          <relationIncontextOf id="r2" type="
              context" relId="SYSTEM-C2"/>
          <relationSupportedBy id="r3" type="
              strategy" relId="SYSTEM-S1"/>
          <relationSupportedBy id="r4" type="
              strategy" relId="SYSTEM-S2"/>
          </relationships>
      </goal>
  </group>
  <group type="context">
    <context id="SYSTEM-C1">
      <description>Software Components</
          description>
    </context>
    <context id="SYSTEM-C2">
      <description>Hardware Components</
          description>
    </context>
    </group>
     ...
</parentArgument>
```

Figure 5. Sample of the ACES specification for the assurance case of Figure 4.

4.2. First System Refinement

Each goal element of an ACES document that contains the `<formalDefinition>` tag relates to CPN specifications considering system refinements. The first system's refinement of the reference model of insulin infusion pump systems has two main modules, representing the entire system based on hardware and software requirements. Thus, the modeling follows the architectural structures of module decomposition and usage.

4.2.1. Hardware Module

We provide the complete reference model in the *MBA/CPN* repository due to the size limitation of figures and to make it available for re-use and simulation [28]. Figure 6 illustrates the hardware module, i.e., the starting point of the intermediate modular hardware decomposition. The hardware module sends messages to a software module by two output interfaces called `Events_H` and `States_H`, representing not valued and valued message exchanges. The basic configuration of the hardware is composed of one starting button, one battery, and one cartridge. However, manufacturers can configure the number of hardware components using the modularization feature of CPN, e.g., adding more batteries to the specified system. The reference model includes this architectural redundancy tactic to help manufacturers deal with common FDA recalls caused by battery malfunctioning. Battery recalls may result in problems such as data loss, communication loss, abrupt therapy interruption, and over infusion and under infusion [11,13].

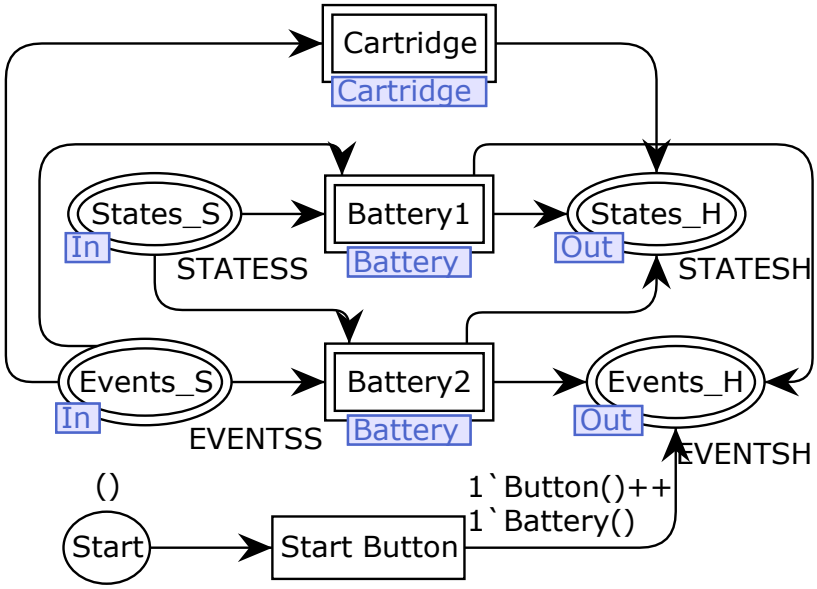

Figure 6. The hardware module of insulin infusion pump systems with two batteries.

The modularization features of CPN allow the re-use of the same battery module to represent multiple battery components. The battery status is defined as charged (1) or discharged (0) using a `Value` place, and it is sent to the software as valued messages (`States_H` place). When the battery is recharged, a message is sent to the software using the `Events_H` place. A fusion place is used to stop the pump when malfunctioning occurs. The model also presents the pump's cartridge module, responsible for recording the maximal capacity of insulin.

4.2.2. Software Module

We divide the software module into three parts related to the steps of intermediate modular decomposition: battery verification, pump configuration, and insulin infusion. The battery verification starts from the `Battery Situation` transition (Figure 7), sending a message to the hardware to obtain the current status of the battery. The software receives a valued message from the hardware (`ValueCharge(b1)`) carrying the current status and verifies if the battery is charged enough to continue working. Otherwise, in addition to releasing the battery verification event, the software releases the battery recharge sending a `Recharge(1)` message to the `Battery` module. If there is more than one battery, the software sends the notification and continues working normally. The initial marking of the fusion place called `Ok_B` specifies the system contains only one battery. If there is more than one battery, the software can send a recharge message to a discharged battery and keep using the charged ones. When all batteries are discharged, the pump stops running, activated again after charging at least one battery. The initial marking of the fusion place `Ok_B` specifies that the system contains only one available battery.

To start using the system, the user should configure the pump by defining the profile that guides operations: standard or personalized. The values' configuration for insulin dosages changes depending on the profile under consideration. The software sends a message to the hardware requiring the current cartridge's capacity to start configuring the pump. The capacity and the insulin dosages for basal, bolus, and bolus corrective are sent to a place named `DC`. The constants `BASAL`, `BOLUS` and `CBOLUS` are used to configure predefined dosage values. Thus, the system can only administrate insulin dosages when these rules are satisfied.

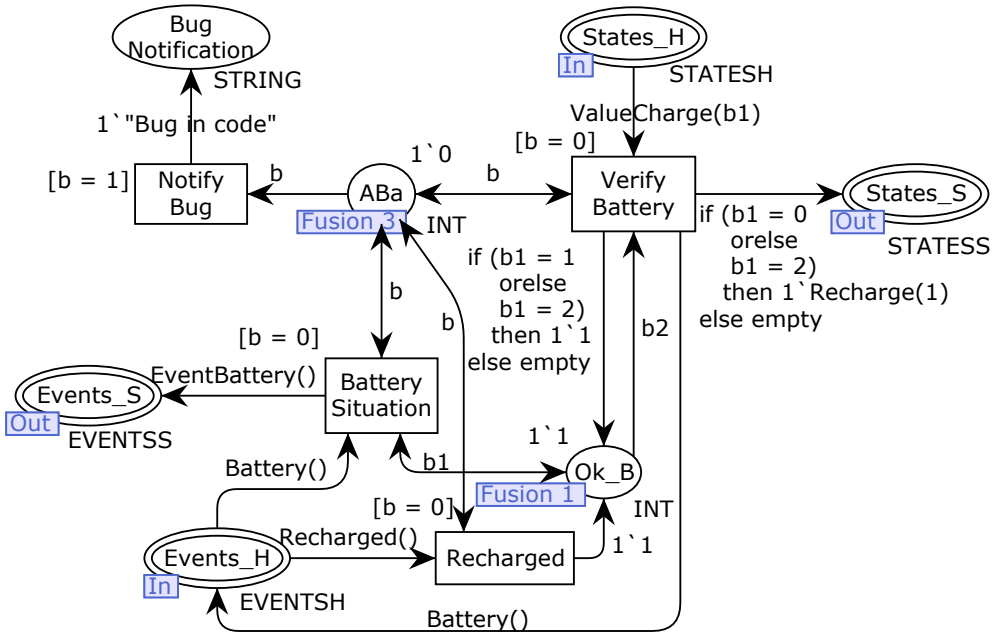

Figure 7. `Verify Battery` module of the first system refinement.

For the personalized mode, it is necessary to define values for different dosages daily. A text file, identified using a `FILE` constant, loads the configuration dosages to define the infusion mode values as list data structures, depending on the medical prescription for a specific clinical case. The remainder of the configuration follows the same approach as the standard mode. At the end of the configuration, two lists represent the basal and bolus dosages (called `listBa` and `listDC`), being the inputs for the insulin infusion step.

Besides describing the insulin infusion pump system's configuration step, the reference model contains standard and personalized infusion modes specifications. Finally, the model includes a submodule for the standard infusion. The first step comprises recording the configuration data, notifying that the pump is executing and loading the cartridge. Afterward, it is possible to select a specific mode by firing one of the transitions called `Adm Basal`, `Adm CBolus`, and `Adm Bolus`. The pump's 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 a unique dosage composed of all insulin; otherwise, the system applies the maximum allowed quantity of insulin dosage.

Once the system is according to all preconditions, it applies the insulin and updates the cartridge's capacity. When there is no more insulin, the system transits to a state that indicates the need to fulfill the cartridge. The system also reaches this state when the cartridge's capacity is below the current insulin dosage being applied. For both situations, the pump transits from the partial state of executing. When the user recharges the pump, the system returns to the partial state of executing.

Three types of software failures are critical to the pump's functioning due to risks to the safety of users: (i) failures that happen when the insulin dosages are being selected (no dosage has been selected), (ii) failures that happen when the insulin dosages are being selected (at least one has been selected), and (iii) failures that happen when insulin dosages are being applied. When failures occur, the system should stop running immediately, informing the user about the malfunctioning. The personalized insulin infusion mode behaves similarly to the standard mode; however, instead of the dosages' constant values, the personalized mode allows the user to define the desired dosages from a text file. This mode applies list data structures to represent the insulin dosages. It converts them into original tuple format, removing the first basal dosage from the list called `listBa` to add the dosage in the head of the list called `listDC`. A place is responsible for recording the remaining basal dosages to be applied. The specification remains almost equal to the

standard mode, where the most significant change regards the usage of the remaining basal dosages, repeating the infusion until there exist no tokens of basal dosages representing the remaining insulin. The specification of the process of recharging the cartridge is also almost equal to the standard mode, adding new places and transitions to control three situations: (i) the system applied the current basal dosage and it is necessary to apply the remaining; (ii) it is necessary to apply the current basal dosage and there are remaining dosages recorded; and (iii) there is no current basal dosage.

The `BASAL`, `BOLUS`, `CBOLUS`, `CAPCART`, `UPPERDOSELIMIT`, `LOWERDOSELIMIT`, `INFUSIONLIMIT`, and `QTDBASAL` are integer input parameters to configure the pump, while the `FILE` parameter receives a string that represents the file name containing basal insulin dosage values. This model refinement consists of integer data types to reduce the state space explosion problem, simplifying the execution of the model checking technique. We previously reported preliminary results related to the first refinement [8], improved in this article by considering FDA guidelines and common recalls.

4.3. Second System Refinement

The second refinement of the reference model of insulin infusion pump systems provides an extended version of the first system refinement, including time constraints and real data type values. This refinement is composed of the same higher-level modules of the first refinement; however, the specification improves the reference model with a more detailed representation of insulin infusion pump systems to comply with stringent regulatory requirements (e.g., the correct rate of infusion) with real and timed data types.

The main modification comprises battery verification, administration control, and insulin infusion, using the timed data types called `UNITTIMED`, `REALTEMP`, and `DATATEMP`. These data types allow manufacturers to handle timed unit data types, timed real data types, and an integer and real values product. In this refinement, the battery verification is guided by controlling the frequency of verification, conducted each 2-time unit, instead of releasing the next verification as soon as the current verification is conducted. The administration control is similar to the standard and personalized infusion modes. The reference model refinements are available for readers who wish to analyze and re-use the specifications [28], along with an example of an ACES-based assurance case template. Figure 8 describes the second model refinement for the standard infusion that shows the administration control, starting from the `Prepare Partial Application` transition to the `Finish Total App` transition.

A user-defined administration rate guides the administration control, and thus, considering an insulin dosage, it does not mean that the pump applies the dosage at once. The system split the dosage up to schedule the infusion based on the administration rate, and the infusion depends on the number of insulin units allowed. For example, the model represents the administration control of basal insulin using the `TABA` place, which maintains basal insulin administration time. Once the system conducts the basal insulin infusion, the `Release Basal1` transition fires, get the next insulin infusion time from the `TABA` place, and configures the next basal dosage. Additionally, the system sends the next time to conduct the insulin infusion to the `TABA` place, enabling the correct configuration for the basal insulin infusion.

The second refinement maintains the same parameters used for the first refinement. However, except for parameter `QTDBASAL`, parameters are assigned with the real data types. The refinement also includes six new input parameters (i.e., `DADMPD`, `TADMPD`, `TADMBPD`, `DADMPS`, `TADMPS`, and `TADMBPS`) related to the administration rate and time.

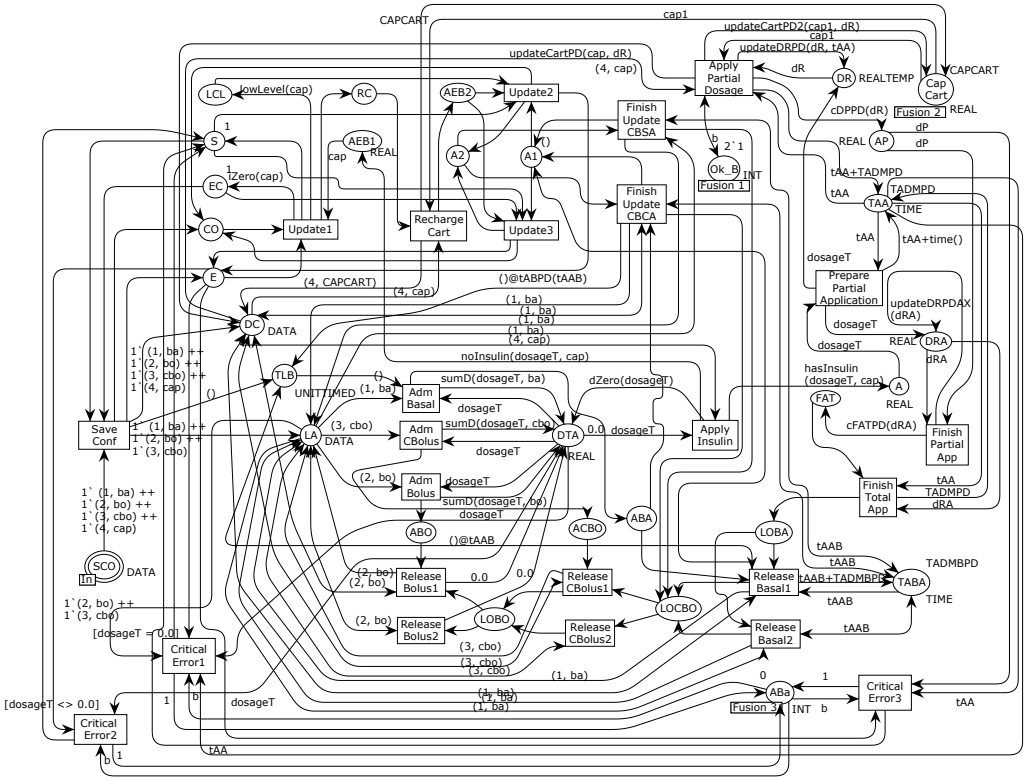

Figure 8. `Standard Infusion` submodule of the second system refinement.

5. Quality Assessment Scenarios

We instantiated the reference model to represent the technical requirements of the ACCU-CHEK Spirit [14], presenting quality assessment scenarios of insulin infusion pump systems using model checking and simulations. Each solution element of an ACES document, defined as an artifact, links to verification and validation results generated using the insulin infusion pump systems model.

5.1. Verification of the First System Refinement

The verification concerns the quality assessment of safety properties. A file called `BasalV2AC.txt` contains 24 hypothetical values of basal dosages for the personalized infusion mode. This activity consists of verifying two safety properties of insulin infusion pump systems that are part of the specification of the ACCU-CHEK Spirit. CPN's model checking technique is based on the Computation Tree Logic (CTL). The functions available on the ASK/CTL library of CPN/Tools represent the specifics of the properties.

The first property defines that when the level of the cartridge is 0, the pump must be stopped (AF$\neg$($cartZeroed$) $\vee$ AF($pumpStop \wedge cartEmpty$)). The ASK-CTL model checker verified that the model complied with this property. The reference model is according to the first property of insulin infusion pump systems for the standard and personalized modes. The second property defines that the pump cannot run when the insulin dosage is greater than the level of the cartridge (AF$\neg$($cartLevel$) $\vee$ AF($pumpStop$)). The ASK-CTL model checker also verified that the model is according to the second property of insulin infusion pump systems. Thus, these are two examples of project artifacts associated with solution elements of the ACES document.

Following the *MBA/CPN*, we reduced the reference model by removing hardware components and adjusting initial markings, verified it again to ensure conformance, and generated abstract tests to guide the validation. To illustrate the abstract test generation, we applied the MBT/CPN tool [29] using the state space of the insulin infusion pump systems model, resulting in the eight abstract tests discussed in the next section.

5.2. Validation of the Second System Refinement

The simulation resources of the CPN/Tools software enabled the validation of the second refinement, analyzing functionalities related to the model's time constraints. Simulations are used to conduct quality assessments of safety and effectiveness properties. The format convention adopted to represent time constraints is a 1-time unit regarding 1 s, 60-time units regarding 1 min, and 3600-time units regarding 1 h.

The parameter called `FILE` has the string data type and keeps track of the file name used to load the insulin dosages for the reference model's personalized mode. Figure 9 illustrates a sample of this simulation step. For the basal insulin infusion, the model obtains the next basal dosage from the `DBR` place and sends it to the `Next Basal` place firing the `Release Basal1` transition. Then, with the fire of the `Release Basal2` transition, the model generates the new duple of basal dosage and sends it to the places called `LA` and `DC`, comprising the next time for the basal dosage. Given that the bolus and corrective bolus insulin dosage values have no administration rate and are constant, the step of releasing them consists of merely firing the transitions called `Release CBolus1` and `Release Bolus1`, enabled because of the infusion of bolus and corrective bolus dosages. Otherwise, the model would enable the transitions called `Release CBolus2` and `Release Bolus2`. The `LA` place carries a real timed data type to record the 2-tuples with the current clock time, maintaining the configuration values of the two types of insulin (`DC` place).

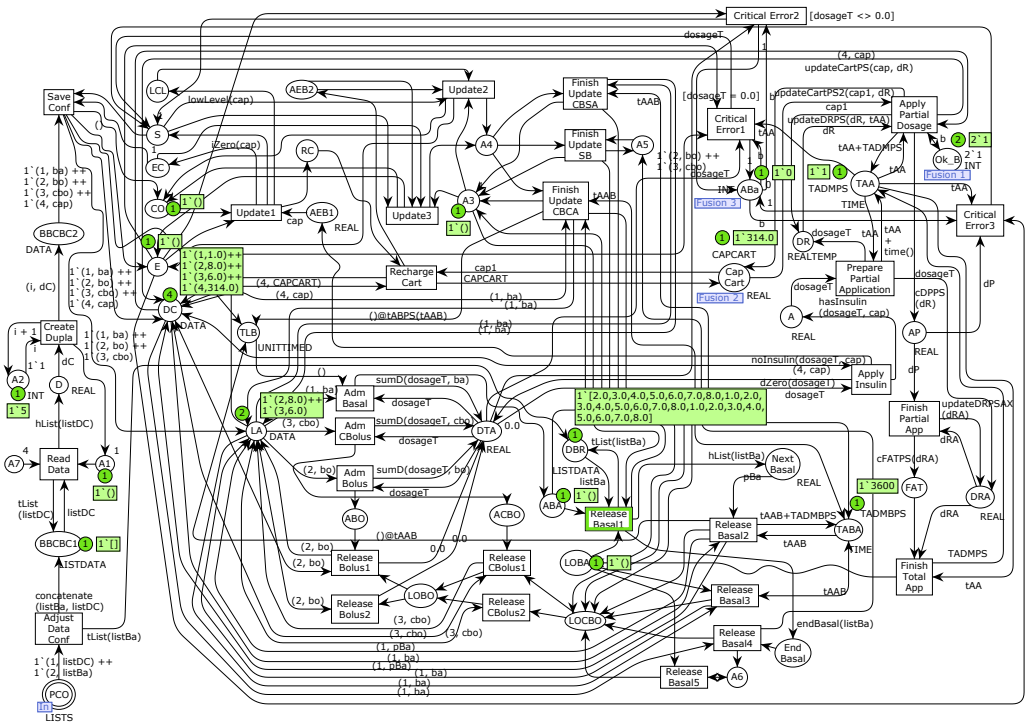

Figure 9. Simulation sample of the ACCU-CHECK Spirit `Personalized Infusion` submodule.

The firing of the `Basal Read` transition calls a function that reads the basal dosage values from a file and adds them to the `LBD` place as a list. The model verifies each value based on predefined upper and lower bounds, uses the `listBa` variable to carry the basal dosages, and adds them to the `DBV` place, waiting for the verification of the remaining values. Table 1 describes the simulations (more than 3000 steps) conducted to evaluate the completeness and correctness of the model based on the eight abstract tests generated with the MBT/CPN tool. Besides, we conducted a large number of random simulations to increase confidence, considering other characteristics such as desired and undesired battery charges. Requirement's simulation results are other examples of project artifacts associated with ACES solutions.

Table 1. Description of simulations conducted based on abstract tests.

Id	Description
1	The pump runs correctly (standard infusion mode).
2	The pump runs correctly (personalized infusion mode).
3	The pump finished executing due to a software critical failure when the insulin doses were selected in the standard infusion mode (no dosage selected).
4	The pump finished executing due to a critical software failure when the insulin doses were selected in the standard infusion mode (at least one selected).
5	The pump finished executing due to a critical software failure when insulin doses are being applied in the standard infusion mode.
6	The pump finished executing due to a software critical failure when the insulin doses were selected in the personalized infusion mode (no dosage selected).
7	The pump finished executing due to a critical software failure when the insulin doses are being selected in the personalized infusion mode (at least one selected).
8	The pump finished executing due to a critical software failure when insulin doses were being applied in the personalized infusion mode.

6. Empirical Evaluation

Aiming to answer the previously presented main RQ, the empirical evaluation verifies whether the *MBA/CPN* can promote reusability and productivity, considering the modelers' viewpoint.

6.1. Scoping, Modelers, and Variables

The goal-question-metric methodology [30] guided the definition of the empirical evaluation through analyzing the usage of the reference model into two aspects: (i) evaluation concerning productivity from the viewpoint of the modelers by instantiating the model; and (ii) evaluation for reusability from the viewpoint of the modelers by instantiating and extending the model. Therefore, we defined the following secondary RQs: does the reference model increase modelers' productivity? (`RQ1`), and is the reference model reusable? (`RQ2`). These RQs guided the specification of the following hypotheses: productivity is not increased (`H0-1`), productivity is increased (`HA-1`), the reference model is not reusable (`H0-2`), and the reference model is reusable (`HA-2`). `H0-1` and `HA-1` are related to `RQ1` and `H0-2` and `HA-2` to `RQ2`.

We selected 12 modelers using the convenience sampling technique and evaluated them at a federal university located in Brazil. Table 2 presents the profiles of the modelers who participated in the study, including questions about age, knowledge about formal methods, opinion about the training phase, resolution of the list of exercises, knowledge about CPN, and knowledge about the present work.

There are two dependent variables defined based on the research goals: modelers' productivity and reference model's reusability. There are four independent variables: the reference model, the modelers' experience, the tool support, and the environment. These variables are controlled at a fixed level, meaning that each group of modelers (i.e., control and treatment) has the same reference model, experience, tool holder, and environment.

Table 2. Questionnaire identifying the modelers' profiles.

Question	1	2	3	4	5	6	7	8	10	11	12	
Age	19	21	23	21	24	23	23	22	22	30	27	26
Knowledge about formal methods [1]	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Opinion about the training phase [2]	Great	Great	Great	Optimum	Optimum	Good	Great	Great	Great	Great	Great	Optimum
Answered the list of exercises [2]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Knowledge about CPN [2]	Good	Very good	Good	Good	Regular	Good	Good	Very good	Good	Good	Good	Regular
Knowledge about the work [2]	Very good	Very good	Good	Good	Good	Very good	Very good	Very good	Good	Good	Good	Regular

[1] After the training phase; [2] Before the training phase; Y: Yes; N: No.

6.2. Procedure and Measures

We used CPN/Tools software to conduct the evaluation based on four phases: first training stage, first evaluation stage, second training stage, and second evaluation stage. We prepared the modelers in the training phase to specify CPN models using the CPN/Tools. We conducted the following activities for the first training stage: (i) we asked modelers to answer a questionnaire regarding personal information, experience with formal methods, and experience with reusability; (ii) the trainer presented a short course regarding concepts and examples related to CPN, CPN/Tools, and the reference model. The trainer also presented an overview of the ACCU-CHEK Spirit; and (iii) the modelers applied the learned techniques to build simple examples assisted by the trainer. The trainer used slides and CPN examples to enable audio-visual presentations while using the particular problem-solving method to support hands-on learning activities.

During the first evaluation stage, the modelers instantiated the reference model to represent ACCU-CHEK Spirit's commercial insulin infusion pump system. They responded to a questionnaire regarding the models' reusability attributes. We divided the modelers into control (size eight) and treatment (size four). The treatment group comprises modelers who have few experiences using CPN (i.e., less than ten hours of a CPN course), while the control group comprises modelers who finished a six months class about CPN and, consequently, at least, six months of experience using CPN. We asked each subject to instantiate the reference model within two hours. We prepared the evaluation material, and the trainer was available during the evaluation to help to solve misunderstandings on the experiment's guidelines and questionnaire wording.

The next step was the second training stage, in which the reference model was explained in more detail. This step was followed by the second evaluation stage, comprising the reference model's usage to extend the specification. The following activities were conducted in the second evaluation stage: (i) each modeler implemented two new requirements within two hours: add a new battery representation and add a feature for recharging the battery; (ii) the modelers were asked to answer a questionnaire about effort and re-use.

We defined metrics to evaluate the hypotheses and assess the RQs. Thus, we addressed time to measure productivity [31], computing the time required by each group to finish the problems that we asked modelers to solve during the evaluation phase. Moreover, we measured reusability using two factors [32]: understandability and adaptability. Under-

standability is related to how easily the modeler recognizes the meaning of a component of the reference model and its applicability, while adaptability stands for how to ease the modeler can extend the reference model to comply with a new system's requirement.

We formalized hypotheses to conduct statistical analyses for the RQ1: the null hypothesis H0-1-1, i.e., $vU > 1$ h, in which vU is the meantime (in minutes) needed by the modelers to conclude instantiating the reference model; and the alternative hypothesis HA-1-1, represented by $vU \leq 1$ h. One hour corresponds to six times the time spent by the researchers to instantiate, for the first time, the reference model based on the ACCU-CHEK Spirit. We also formalized the hypotheses to conduct statistical analyses related to the effort factor: the null hypothesis H0-1-2, i.e., $vU > 3$, in which vU represents the mean classification of the responses for the effort questions; and the alternative hypothesis HA-1-2, represented by $vU \leq 3$.

We also formalized hypotheses to conduct statistical analyses related to the RQ2. Considering understandability, there is a null hypothesis H0-2-1, i.e., $vU \leq 3$, in which vU represents the mean classification of the responses for the understandability questions, while the alternative hypothesis HA-2-1 is represented by $vU > 3$. Considering adaptability, there is a null hypothesis H0-2-2, i.e., $vU \leq 3$, in which vU represents the mean classification of the responses for the adaptability questions, while the alternative hypothesis HA-2-2 is represented by $vU > 3$.

Finally, we applied a questionnaire at the end of the experiments to collect metrics for the effort and reusability factors by providing a 5-point Likert scale (1) to (5). The scale interpretation concerning the metrics is based on the effort of instantiating the reference model during experiment 01: (1) represents the best result and (5) represents the worst result. For the understandability and adaptability measures conducted during experiment 02, the scale's interpretation is (1) for the worst and (5) for the best. The modelers answered the questionnaire after extending the reference model.

6.3. Analysis

Figure 10 depicts the answers for effort and each reusability factor evaluated in experiments 01 and 02, respectively. We did not apply more complex hypotheses tests such as the t-test and Wilcoxon test because the hypotheses were defined to evaluate RQ1 and RQ2 only use the mean responses concerning the factors considered in this experiment. The basic descriptive statistics enabled the evaluation of the hypotheses. For RQ1, we evaluated the reference model's re-use based on the analysis of the improvement of the productivity of the modelers. The 12 modelers who participated in the experiment concluded instantiating the model, and in the average case, within six minutes.

When asked to respond to the questionnaire, 66.7% of the modelers (8 of the 12 modelers) stated that the task of instantiating the reference model presented no effort. In contrast, 8.3% stated low effort, 16.7% stated medium effort, and 8.3% stated considerable effort. Thus, the hypotheses H0-1-1 and H0-1-2 were refuted, showing that the model presented a positive response to the RQ1 (the modelers used only 10% of the estimated time).

We evaluated the reference model's re-use to analyze the RQ2 based on the understandability and adaptability factors. Figure 11 presents the number of modelers stating to extend the model considering the requirements 01 and/or 02, also including the number of modelers who experimented 02 correctly. All the modelers implemented requirement 01 (i.e., adding a new battery).

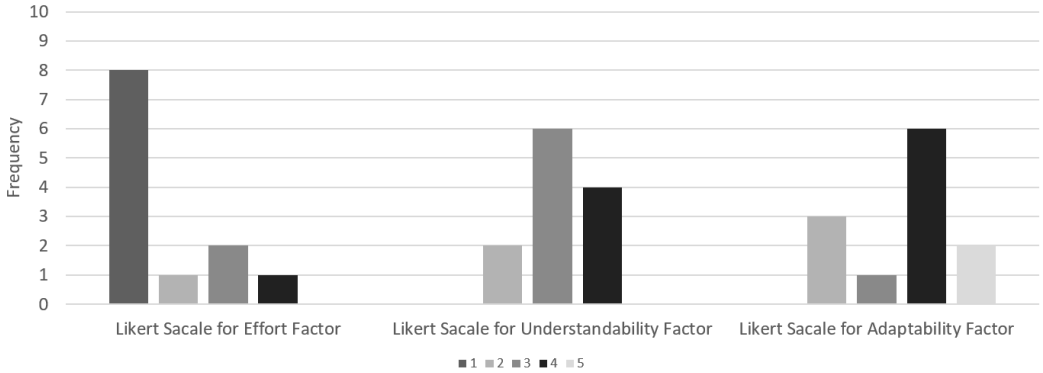

Figure 10. Answers distribution in accordance with the effort and reusability factors.

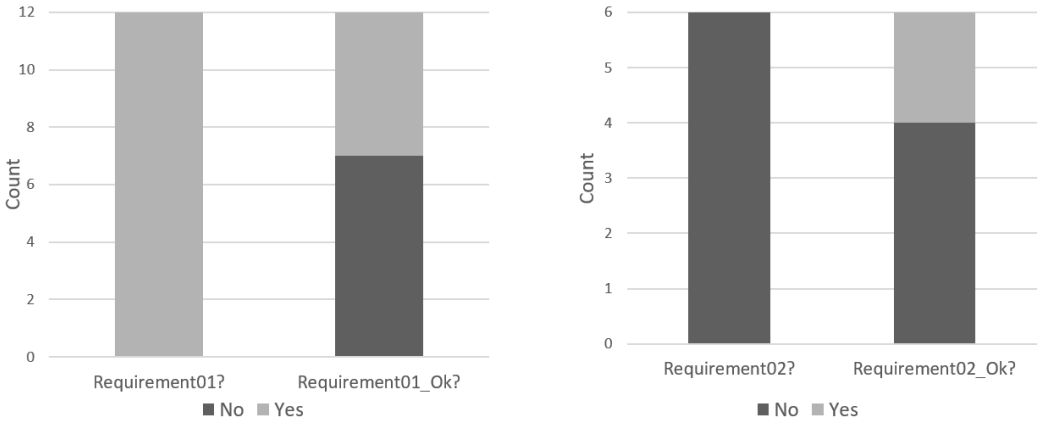

Figure 11. Instantiated model vs. correctly instantiated.

However, only five of them implemented it successfully. The analyses of the work conducted by the seven modelers who did not implement the requirement 01 successfully showed that all of them failed in the same task: the marking of the `Ok_B` place (`Verify Battery` submodule) was not changed from `2'1` to `3'1`. Six modelers attempted to implement requirement 02. However, only two of them achieved the correct answer. The results of the four modelers who did not achieve the correct implementation showed different mistakes, all related to the usage of CPN/ML.

To analyze the time spent to conduct the second experiment, we considered the group related to each modeler. Figure 12 shows the time collected for the control and treatment group. Only one of the modelers successfully implemented the reference model's first and second requirements within 25 and 38 min, respectively. The modelers scaled the first requirement as a low-complexity task while disagreeing when asked about the complexity of the second requirement: the control group's modeler scaled as a medium-complex task. In contrast, the modeler of the treatment group scaled it as a low-complex task.

Analyzing understandability and adaptability from the results presented in Figures 11 and 12 does not enable us to conclude that the reference model is fully reusable, requiring the analysis of results presented in Figure 10. Most modelers did not fully understand each component of the reference model's roles and functioning. The high number of modules required by the complexity of insulin infusion pump systems negatively impacted adapting the model adding new requirements.

Figure 13 supports this claim showing modelers' answers for the complexity of each requirement added to the reference model in the second experiment. Most of the modelers stated the requirements 01 being of low (75%) or medium (25%) complexity, while stated medium or high complexity (75%) for requirement 02.

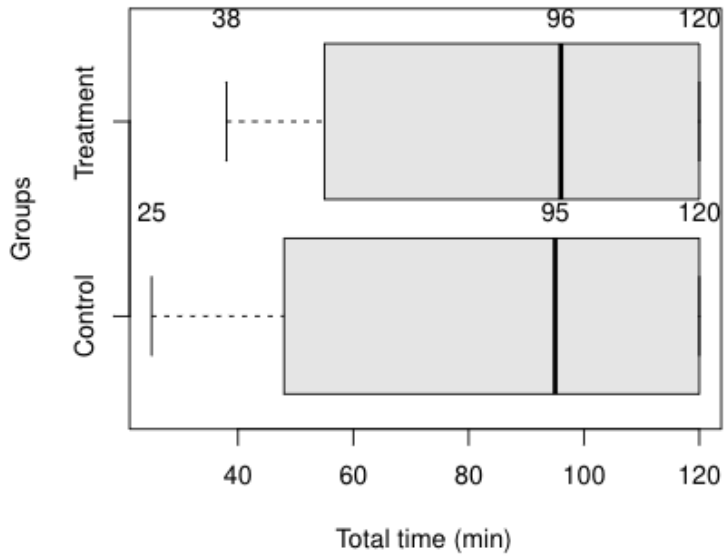

Figure 12. Time collected for the control and treatment group.

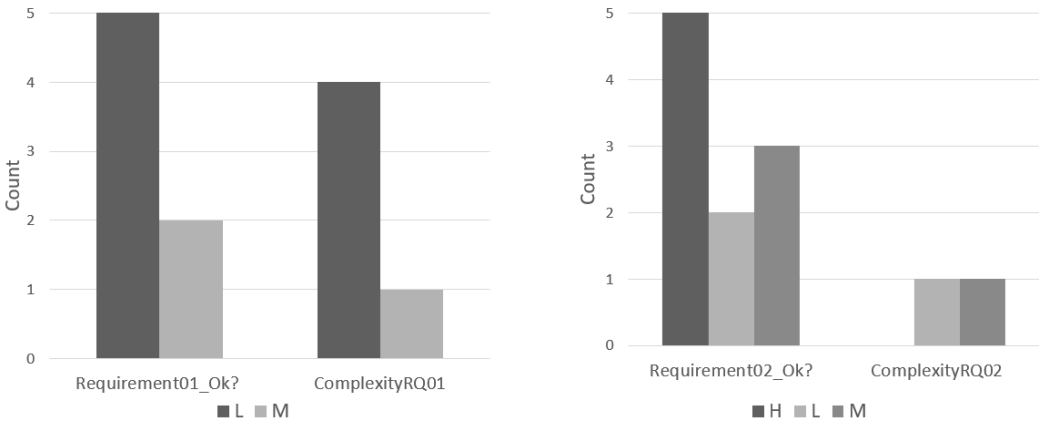

Figure 13. Answers distribution considering complexity of functionalities.

The evaluation of RQ2 based on the questioning presented to the modelers showed that H0-2-1 was refuted due to the mean classification of the understandability factor (approximately 3.17). For H0-2-2, the mean classification of the adaptability factor presented approximately 3.58, refuting the null hypothesis. Thus, the reference model presented acceptable levels of understandability and adaptability.

7. Web-Based Application

Based on the empirical evaluation results, we used the Access/CPN framework and web services to implement a web-based application, aiming to assist modelers to re-use our approach by simulation-based training. The usage of Access/CPN enabled us to embed the reference models in web services to conduct simulations without the GUI of the CPN/Tools, improving the *MBA/CPN*. Thus, we provide web services as consumers of Access/CPN components (Figure 14).

As an API RESTful, the availability of services enables modelers to easily re-use Access/CPN functionalities with no concerns related to development platforms. Other developers, besides our application, can re-use such web API to embed CPN models in different scenarios. The application provides feedback about model components meaning during background simulations (i.e., without the CPN/Tools) to improve productivity and understandability, guided by an easy-to-use GUI.

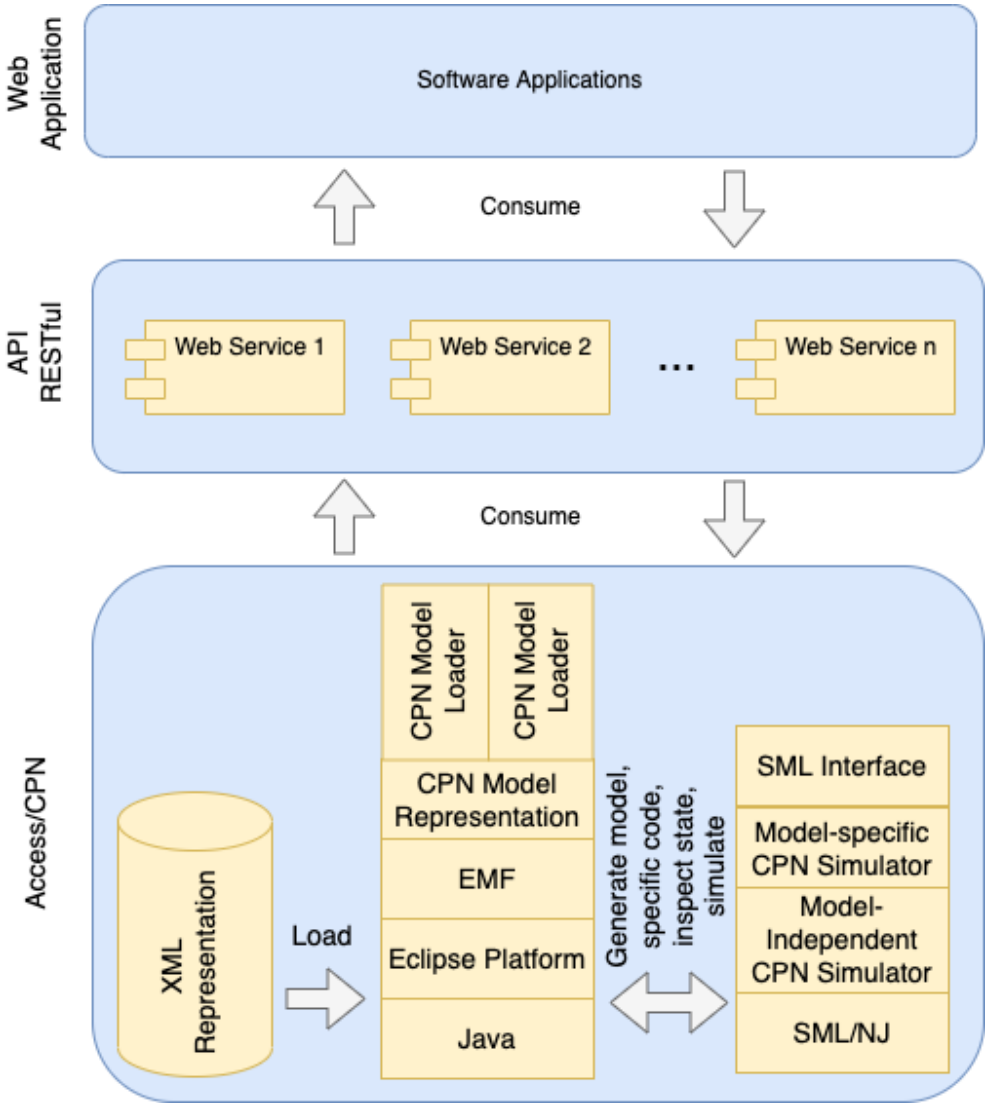

Figure 14. Web application and API implemented as consumers of Access/CPN components [16].

Besides the available public interfaces of Access/CPN, our API RESTful provides new public interfaces to further simplify the background simulation of CPN models. For instance, we provide web services to assist modelers to stop the simulation of the model when certain stop conditions are satisfied, such as reaching specific desired transitions. Such web services are relevant to assist modelers in analyzing specific system requirements during the simulations.

Figures 15 and 16 present GUI samples of the web-based application. For instance, the application enables modelers to simulate the configuration of the pump, showing each simulation step (e.g., along with information related to places and transitions). The web services used to implement such software can be easily adapted to handle a different reference model version. We implemented the application using Java programming language and the JavaScript React library. We also carefully tested the web-based application using the API development environment Postman. Thus, we conducted background CPN model simulations using the web-based application for each insulin infusion pump system requirement.

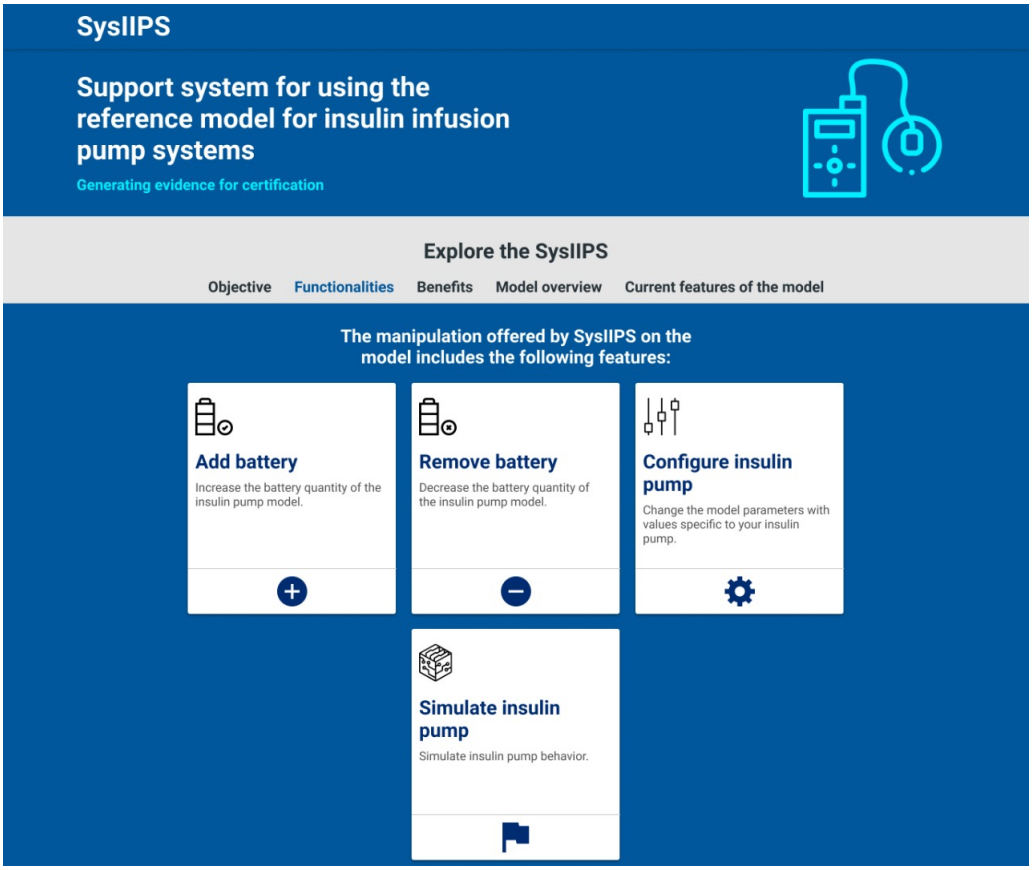

Figure 15. GUI sample of the web-based application presenting main functionalities.

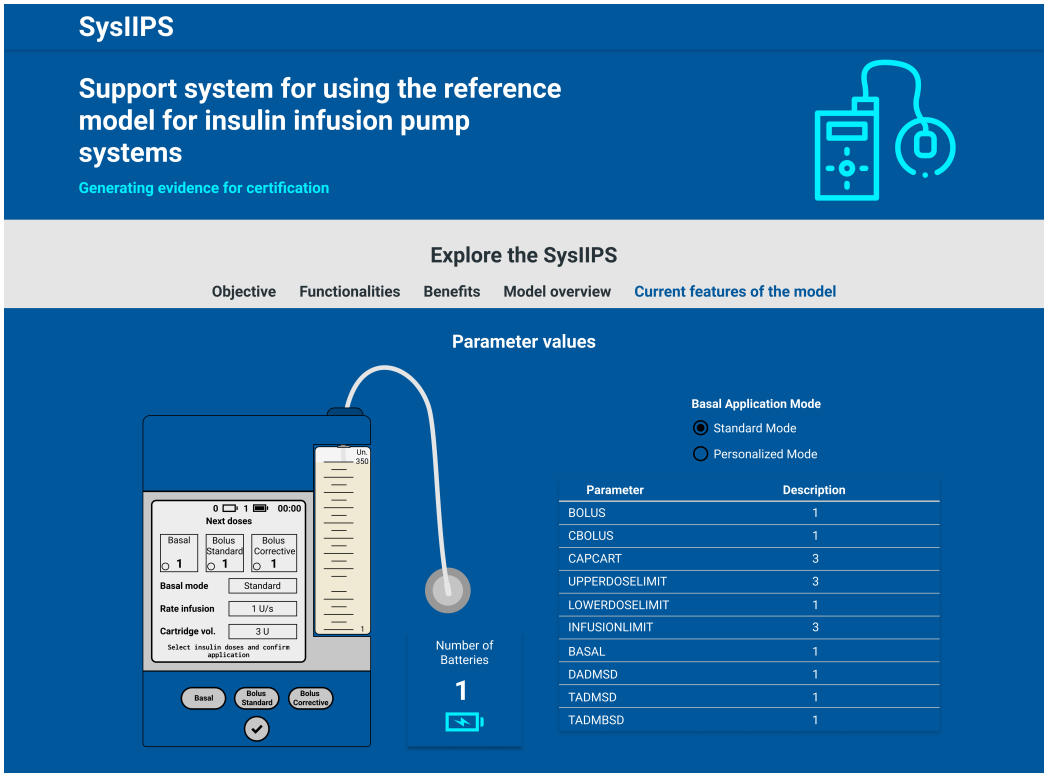

Figure 16. GUI sample of the web-based application for insulin infusion pump simulation.

The modeler can use the web-based application to handle the CPN model of insulin infusion pump systems using four main features: (1) add a battery to the system; (2) remove

a battery to the system; (3) configure the insulin infusion pump; and (4) simulate the insulin infusion pump. Features (1) and (2) allow the user to change the structure of the CPN model by adding/removing an instance of a module that represents the battery of the insulin infusion pump. Such changes are allowed due to the modularization capabilities of hierarchical CPN. Feature (3) allows modelers to change the values of the parameters of the reference model. Finally, Feature (4) allows modelers to simulate the CPN model using the GUI of the web-based application, without using CPN/Tools, to improve the understanding of the pump's behavior without the user having access to the internal structure of the model.

8. Discussion

The *MBA/CPN* has characteristics that were not entirely considered by previous studies, such as assurance case documentation for goal-oriented requirements engineering (ACES documents), configuration (model instantiated for different systems), scheduling (constraints of infusion systems), hierarchy (management of specification complexity), and execution (simulation of behaviors). The documentation of requirements using ACES-based assurance cases helps manufacturers provide arguments and evidence for certification, reusing the same document to manage requirements. This is also relevant to decrease development time, considering that government regulatory agencies (e.g., the FDA) recommend submitting assurance cases of insulin infusion pump systems during the certification process. Manufacturers and regulatory agencies can exchange the ACES document to improve the quality of insulin infusion pump systems.

Besides, to apply the *MBA/CPN* at a real scenario using a specific system, manufacturers only need to configure input parameters of the reference model to represent the basic required functionalities of insulin infusion pump systems. If new functionality is required for the system, the reference model's module structure provides an easy mechanism for adaptations. The reference model's re-use is the basis for developing a new system or conducting quality assessments of existing ones, comparing the insulin infusion pump systems under evaluation and the model. Although we manually specified the CPN models due to our focus on providing a verified and validated basis for further extensions, it is also possible to automatically generate models with a higher level of abstraction (first system refinement - abstract reference model of Figure 2) by inferring them from the ACES specification. The automatic generation of CPN models may be relevant if manufacturers wish to include new goals for the insulin infusion pump systems under development or wish to re-use our approach for other critical systems.

The formal model is a project artifact used to conduct verification and validation activities to increase confidence in insulin infusion pump systems. The verification is relevant to conducting quality assessments of quality attributes during a certification process. The quality assessment may be conducted by comparing each module of the verified reference model with each module of the insulin infusion pump systems under evaluation. The validation is relevant to conducting quality assessments of both safety and effectiveness of insulin infusion pump systems. For example, when a recall is reported, the reference model may assist manufacturers in evaluating the system by comparing outputs with model simulation results. This can decrease cost and development time by promptly correcting the defects/failures.

Considering the FDA guideline related to the life cycle of infusion pumps [11], the reference model complies with the most common reported recalls of infusion pumps, including software error and over infusion, and under infusion. Additionally, based on the same FDA guideline, the model complies with the basic requirements of infusion pump systems: infusion delivery mechanism, drug reservoir, bolus mechanism, real-time clock, pump log, and selection of therapy.

Therefore, manufacturers can re-use the *MBA/CPN* to develop insulin infusion pump systems that comply with the main FDA requirements, correct known problems, and identify and correct unknown problems reported by regulatory agencies as recalls. However, it

is essential to highlight those other relevant characteristics of infusion pumps that are not in the scope of *MBA/CPN*, such as hardware failures, hardware degradation, communication interfaces, power supply, and electrical requirements.

During the empirical evaluation, the task of representing a specific system, the ACCU-CHEK Spirit, lasted only for a few minutes, indicating that extending the modeling is not time-consuming. This finding helps insulin infusion pump systems manufacturers deal with increased development time and project costs when applying formal methods. The questionnaire responses for the two experiments reflect the favorable opinion of the interviewed modelers concerning the re-use of the reference model. All modelers reported an outstanding or optimum achievement of the training phase, showing proper knowledge about the study (approximately 91.7%) and the CPN (approximately 83.3%). The results indicate a positive response to the main RQ. However, the modelers' performance was low when conducting modifications in the reference model, presenting a usability problem. A hypothesis of a solution to address this problem is the usage of GUI to assist modelers in understanding and modifying model components as a learning mechanism. Such a hypothesis motivated the web-based application's implementation to assist the simulation-based training of modelers.

Implementing the application using the Access/CPN framework to conduct the background execution of CPN models enabled the improvement of *MBA/CPN* because it does not require knowledge about all model components to conduct simulations. The application provides feedback about the meaning of such model components to improve the understanding of modelers. Besides, during the implementation, we were also able to provide an API RESTful (Figure 14) that can be re-used by other developers of safety-critical systems (e.g., to perform other simulation-based training). The simulation-based training using the software application may also be relevant to improving systems' GUI design. As stated in the introduction, one of the problems faced by insulin infusion pump systems developers is the provision of inadequate GUI [10].

However, there are some limitations to validity. Two refinements of the reference model enabled the model checking and validation by simulations. With the generation of the strongly connected components graph, this approach prevented the state space explosion problem instead of using a more robust technique to reduce the more detailed version's state space, e.g., equivalence method. This may have made the model harder to handle, considering that manufacturers need to split the verification and validation activities into different specification versions. The empirical evaluation showed that modelers reported the model as an easy-to-understand project artifact. In addition, we used two properties of insulin infusion pump systems to conduct the model checking. This may have limited the verification step because other property regulatory agencies may consider relevant in the certification process. The simulations of the model by hypothetical insulin dosages may also limit the validation compared to using real data from a medical prescription. However, we used verification and validation to illustrate quality assessment scenarios rather than fully validate an instantiated system's behaviors.

9. Conclusions and Future Work

We presented an MBA focused on CPN reference models of insulin infusion pump systems, aiming to assist manufacturers in assessing quality. We also described a case study on a commercial insulin infusion pump system (i.e., ACCU-CHEK Spirit). Reference models are relevant because insulin infusion pumps are safety-critical systems used to monitor and treat diabetes patients. The executable, parametric, modular, and timed CPN model included the essential features of insulin infusion pump systems to ensure correct functioning. The case study was relevant to extending the model to verify and validate two model refinements as quality assessment scenarios. Using the model checking technique, we considered the formal verification of two safety properties for the first refinement. We validated the second refinement using simulations to analyze the model concerning the commercial system's technical specifications. Finally, the empirical evaluation and the

implementation of the web-based application for simulation-based training demonstrated that the *MBA/CPN* is reusable for the development and certification process of insulin infusion pump systems. Thus, the usage of *MBA/CPN* can enable manufacturers to reduce costs and development time when using formal models of insulin infusion pump systems. Manufacturers of other medical devices can also re-use it by defining and evaluating a specific reference model based on the proposed modeling steps.

As there is a lack of studies providing generic, parametric, and timed insulin infusion pump systems models, the methods applied in this study enabled the improvement of available specifications of such systems. Thus, such a research gap also resulted in the lack of studies providing evaluations of productivity and reusability of models when reusing insulin infusion pump systems CPN models. Our study addressed such limitations improving the state-of-the art. During the empirical evaluation, time showed to be a relevant metric to measure productivity (computing the time required by each group to finish the problems that we asked modelers to solve during the evaluation phase). Two factors (understandability and adaptability) also showed to be relevant to measure reusability. Regarding formal techniques (e.g., CPN and model checking), although it is a consensus that the proposal and the use of formal methods to validate the behavior of safety-critical systems is valuable, our proposal also works as a guide, along with reusable project artifacts, for developers of insulin infusion pump systems.

As future work, we envision conducting another empirical study to evaluate the web-based application, integrated with the reference model, to analyze if understandability and adaptability are improved. We also envision investigating the automatic generation of CPN models with a higher level of abstraction from the ACES specification. Such automatically generated models may decrease the efforts of modelers during the specification steps of our approach.

Supplementary Materials: The following are available online at https://bit.ly/2Qyci3k, Additional Figures, Additional Examples, and Source of Models.

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Federal University of Campina Grande.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for supporting this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mertz, L. Automated Insulin Delivery: Taking the Guesswork out of Diabetes Management. *IEEE Pulse* **2018**, *9*, 8–9. [CrossRef] [PubMed]
2. Freckmann, G.; Kamecke, U.; Waldenmaier, D.; Haug, C.; Ziegler, R. Accuracy of Bolus and Basal Rate Delivery of Different Insulin Pump Systems. *Diabetes Technol.* **2019**, *21*, 201–208. [CrossRef] [PubMed]
3. Woodcock, J.; Larsen, P.G.; Bicarregui, J.; Fitzgerald, J. Formal methods: Practice and experience. *ACM Comput. Surv.* **2009**, *41*, 1–36. [CrossRef]
4. Chen, Y.; Lawford, M.; Wang, H.; Wassyng, A. Insulin Pump Software Certification. In Proceedings of the International Symposium on Foundations of Health Informatics Engineering and Systems, Macau, China, 21–23 August 2013; pp. 87–106.
5. Sivakumar, M.S.; Casey, V.; McCaffery, F.; Coleman, G. Improving Verification & Validation in the Medical Device Domain. In Proceedings of the European Conference on Software Process Improvement, Roskilde, Denmark, 27–29 June 2011; pp. 61–71.
6. Jensen, K.; Kristensen, L.M. Colored Petri nets: A graphical language for formal modeling and validation of concurrent systems. *Commun. ACM* **2015**, *58*, 61–70. [CrossRef]

7. Sobrinho, A.; Silva, L.D.; Perkusich, A.; Cunha, P.; Cordeiro, T.; Lima, A.M.N. Formal modeling of biomedical signal acquisition systems: Source of evidence for certification. *Softw. Syst. Model.* **2019**, *18*, 1467–1485. [CrossRef]
8. Costa, T.F.; Sobrinho, A.; Silva, L.C.; e Silva, L.D.; Perkusich, A. A Coloured Petri Nets Reference Model of Insulin Infusion Pump Control Systems: Assisting the Certification Process. In Proceedings of the 45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 14–17 October 2019; pp. 2871–2876.
9. Medical Device Classification Procedures: Incorporating FDA Safety and Innovation Act Procedures (Final Rule) Regulatory Impact Analysis. Available online: https://bit.ly/3oaqDnN (accessed on 25 January 2022).
10. Infusion Pumps Total Product Life Cycle: Guidance for Industry and FDA Staff. Available online: https://bit.ly/2QAHe2V (accessed on 25 January 2022).
11. Infusion Pumps Total Product Life Cycle. Available online: https://bit.ly/2EMOXrW (accessed on 25 January 2022).
12. Rathore, H.; Wenzel, L.; Al-Ali, A.K.; Mohamed, A.; Du, X.; Guizani, M. Multi-Layer Perceptron Model on Chip for Secure Diabetic Treatment. *IEEE Access* **2018**, *6*, 44718–44730. [CrossRef]
13. Gao, X.; Wen, Q.; Duan, X.; Jin, W.; Tang, X.; Zhong, L.; Xia, S.; Feng, H.; Zhong, D. A Hazard Analysis of Class I Recalls of Infusion Pumps. *JMIR Hum. Factors* **2019**, *6*, 10366. [CrossRef]
14. Accu-Chek Spirit: Pump User Guide. Available online: https://bit.ly/2QzJMOT (accessed on 25 January 2022).
15. Goal Structuring Notation. Available online: https://bit.ly/3z0HbTD (accessed on 25 January 2022).
16. Westergaard, M. Access/CPN 2.0: A High-Level Interface to Coloured Petri Net Models. In Proceedings of the International Conference on Application and Theory of Petri Nets and Concurrency, Newcastle, UK, 20–24 June 2011; pp. 328–337.
17. Majma, N.; Babamir, S.M. Model-Based Monitoring and Adaptation of Pacemaker Behavior Using Hierarchical Fuzzy Colored Petri-Nets. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 3344–3357. [CrossRef]
18. Mian, Z.; Bottaci, L.; Papadopoulos, Y.; Mahmud, N. Model transformation for analyzing dependability of AADL model by using HiP-HOPS. *J. Syst. Softw.* **2019**, *151*, 258–282. [CrossRef]
19. Entezari-Maleki, R.; Etesami, S.E.; Ghorbani, N.; Niaki, A.A.; Sousa, L.; Movaghar, A. Modeling and Evaluation of Service Composition in Commercial Multiclouds Using Timed Colored Petri Nets. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 947–961. [CrossRef]
20. Garcìa-Valls, M.; Perez-Palacin, D.; Mirandola, R. Pragmatic cyber physical systems design based on parametric models. *J. Syst. Softw.* **2018**, *144*, 559–572. [CrossRef]
21. Montecchi, L.; Lollini, P.; Bondavalli, A. A Template-Based Methodology for the Specification and Automated Composition of Performability Models. *IEEE Trans. Reliab.* **2020**, *69*, 293–309. [CrossRef]
22. Kanoun, K.; Ortalo-Borrel, M. Fault-tolerant system dependability-explicit modeling of hardware and software component-interactions. *IEEE Trans. Reliab.* **2020**, *49*, 363–376. [CrossRef]
23. Nencioni, G.; Helvik, B.E.; Heegaard, P.E. Including Failure Correlation in Availability Modeling of a Software-Defined Backbone Network. *IEEE Trans. Netw. Serv. Manag.* **2017**, *14*, 1032–1045. [CrossRef]
24. Rabah, M.; Kanoun, K. Performability evaluation of multipurpose multiprocessor systems: The "separation of concerns" approach. *IEEE Trans. Comput.* **2003**, *52*, 223–236. [CrossRef]
25. Silva, L.C.; Almeida, H.O.; Perkusich, A.; Perkusich, M. A Model-Based Approach to Support Validation of Medical Cyber-Physical Systems. *Sensors* **2015**, *15*, 27625–27670. [CrossRef]
26. Zhang, Y.; Jones, P.L.; Jetley, R. A Hazard Analysis for a Generic Insulin Infusion Pump. *J. Diabetes Sci. Technol.* **2010**, *4*, 263–283. [CrossRef]
27. Hatcliff, J.; Larson, B.; Carpenter, T.; Jones, P.; Zhang, Y.; Jorgens, J. The Open PCA Pump Project: An Exemplar Open Source Medical Device as a Community Resource. *ACM Sigbed Rev.* **2019**, *16*, 8–13. [CrossRef]
28. MBA/CPN: Model-Based Approach (MBA) with Coloured Petri Nets (CPN). Available online: https://bit.ly/2Qyci3k (accessed on 25 January 2022).
29. Wang, R.; Kristensen, L.M.; Meling, H.; Stolz, V. Automated test case generation for the Paxos single-decree protocol using a coloured Petri net model. *J. Log. Algebr. Methods Program.* **2019**, *104*, 254–273. [CrossRef]
30. Basili, V.R.; Rombach, H.D. The TAME project: Towards improvement-oriented software environments. *IEEE Trans. Softw. Eng.* **1988**, *14*, 758–773. [CrossRef]
31. Kitchenham, B.; Pickard, L.; Pfleeger, S.L. Case studies for method and tool evaluation. *IEEE Softw.* **1995**, *12*, 52–62. [CrossRef]
32. Washizaki, H.; Yamamoto, H.; Fukazawa, Y. A metrics suite for measuring reusability of software components. In Proceedings of the 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry, Sydney, Australia, 3–5 September 2003; pp. 211–223.