WEBTAMP: A WEB TOOL TO PERFORM STPA ANALYSIS

Celso Massaki Hirata, ITA

hirata@ita.br

Project Team

The project is a collaborative effort of many voluntary contributors. They include:

- Celso Massaki Hirata (ITA) (project coordinator)
  emails: hirata@ita.br hiratacm@gmail.com
- Fellipe Guilherme Rey de Souza (ITA)
- Rodrigo Martins Pagliares (UNIFAL-MG)
- Juliana de Melo Bezerra (ITA)
- Filipe Parisoto Ribeiro (UNIFAL-MG)
- João Hugo Marinho Maimone (UNIFAL-MG)
- and many others.

Agenda

- About WebSTAMP
- System description – Insulin pump with continuous glucose monitor
- Define purpose of the analysis
- Model the control structure
- Identify Unsafe Control Actions
- Identify Loss Scenarios
- Concluding Remarks

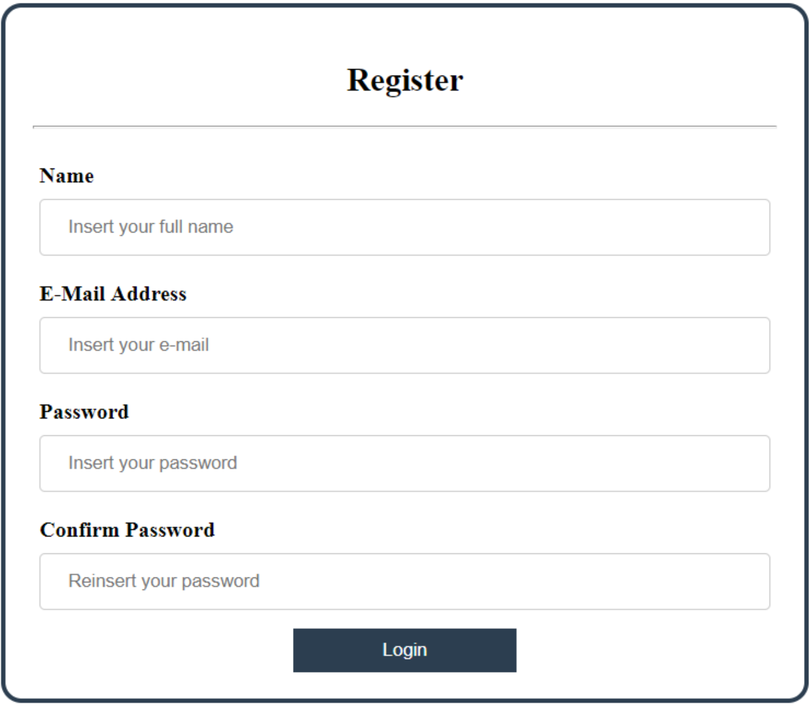
WebSTAMP

- WebSTAMP is a **web** application to perform basic Systems-Theoretic Process Analysis (STPA).
- You can access WebSTAMP – available in the link: http://webstamp.herokuapp.com.
- We will conduct an analysis of the system “Insulin Pump with continuous glucose monitor”.

WebSTAMP

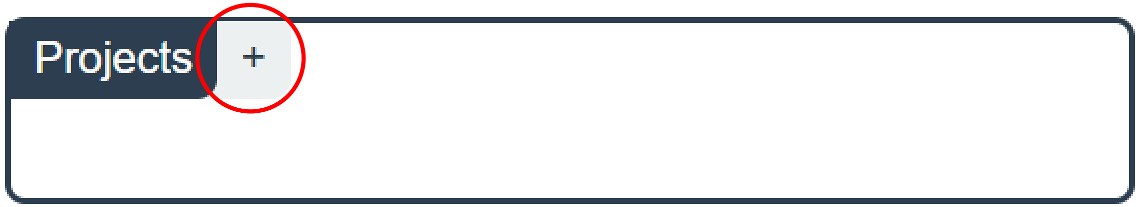
- After accessing WebSTAMP, find and click in the top right menu the option “Register”. Insert your full name, e-mail address, and password to continue.

- Once you have completed your registration (no need of e-mail confirmation), WebSTAMP will redirect you to the “Projects” page.



WebSTAMP

- The next step is create a new Project. Click in the “**+**” button near the label “Projects” to add a new Project.



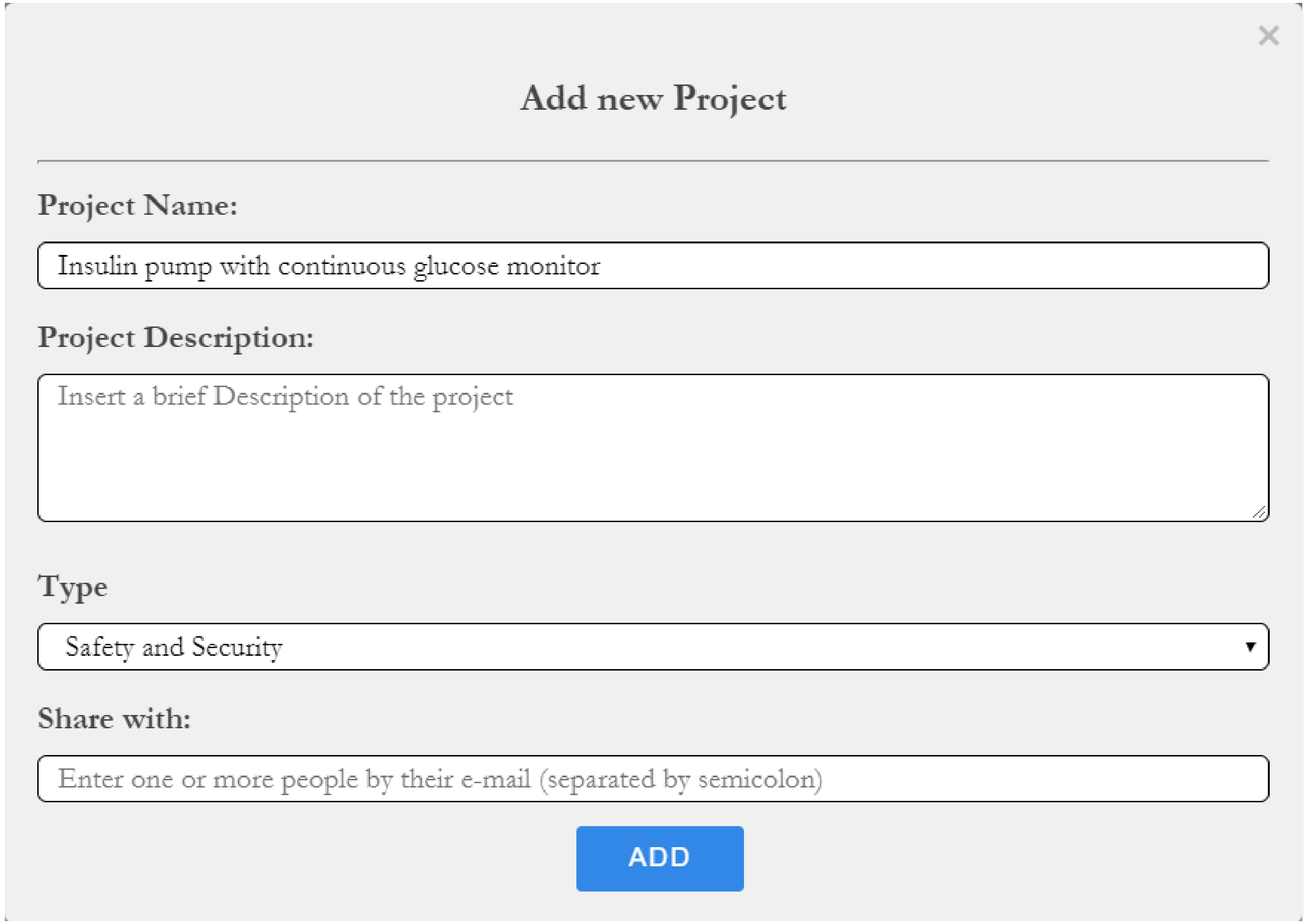
- Insert the information:
  - **Project Name**: Insulin pump with continuous glucose monitor
  - **Type**: Safety and Security
  - **System Description** and **Shared With** are optional. There is no need to assign any value.

WebSTAMP

Okay, but what is an **Insulin pump with continuous glucose monitor**?

What are the safety and security concerns?

The next slides are going to introduce you to the step “Define purpose of the analysis”.



Add new Project

Project Name:

Insulin pump with continuous glucose monitor

Project Description:

Insert a brief Description of the project

Type

Safety and Security ▾

Share with:

Enter one or more people by their e-mail (separated by semicolon)

Motivation for the system

- People with Diabetes may take 1-2 insulin injections of a long-acting insulin every day and three or more injections of rapid-acting insulin for meals and snacks.
- The typical person with Type 1 Diabetes can take 4-7 injections a day. Many people currently receive insulin through an insulin pen or a syringe.
- An insulin pump delivers rapid acting insulin in two ways.

Motivation for the system

- **Basal** is the insulin that a person needs even in the absence of food. The basal rate replaces the long acting injection that a person takes. The pump is programmed to give insulin every hour.
- **Bolus** is the insulin that a person takes for food or to correct a high blood sugar.
- Once a person is on a pump, all insulin is delivered through the pump and shots are no longer necessary.

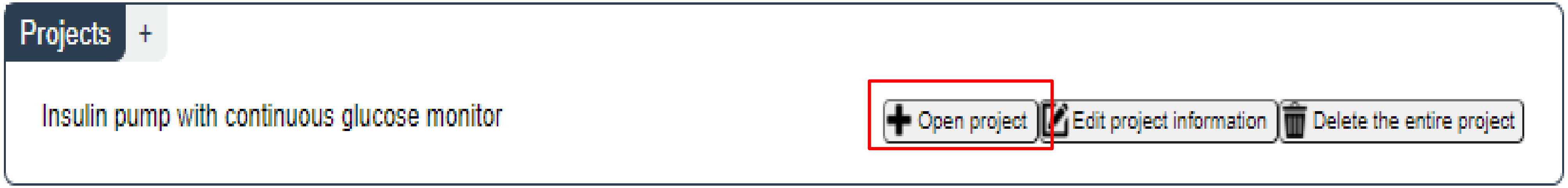
The system

- The system is comprised of a device named **Insulin Pump** that pumps insulin in the **Patient's Body** and a **Mobile device with an app** to control the amount and frequency of insulin injected in the patient.
- The **Patient** is responsible to determine the amount and the frequency of insulin to be injected.
- Why worry about cybersecurity?
    - In August 2011, Radcliffe [1] demonstrated a security flaw in insulin pumps. It was possible to remotely hack the wireless interface used to control the pump.

WebSTAMP

- After creating a Project, it is time to work on it.

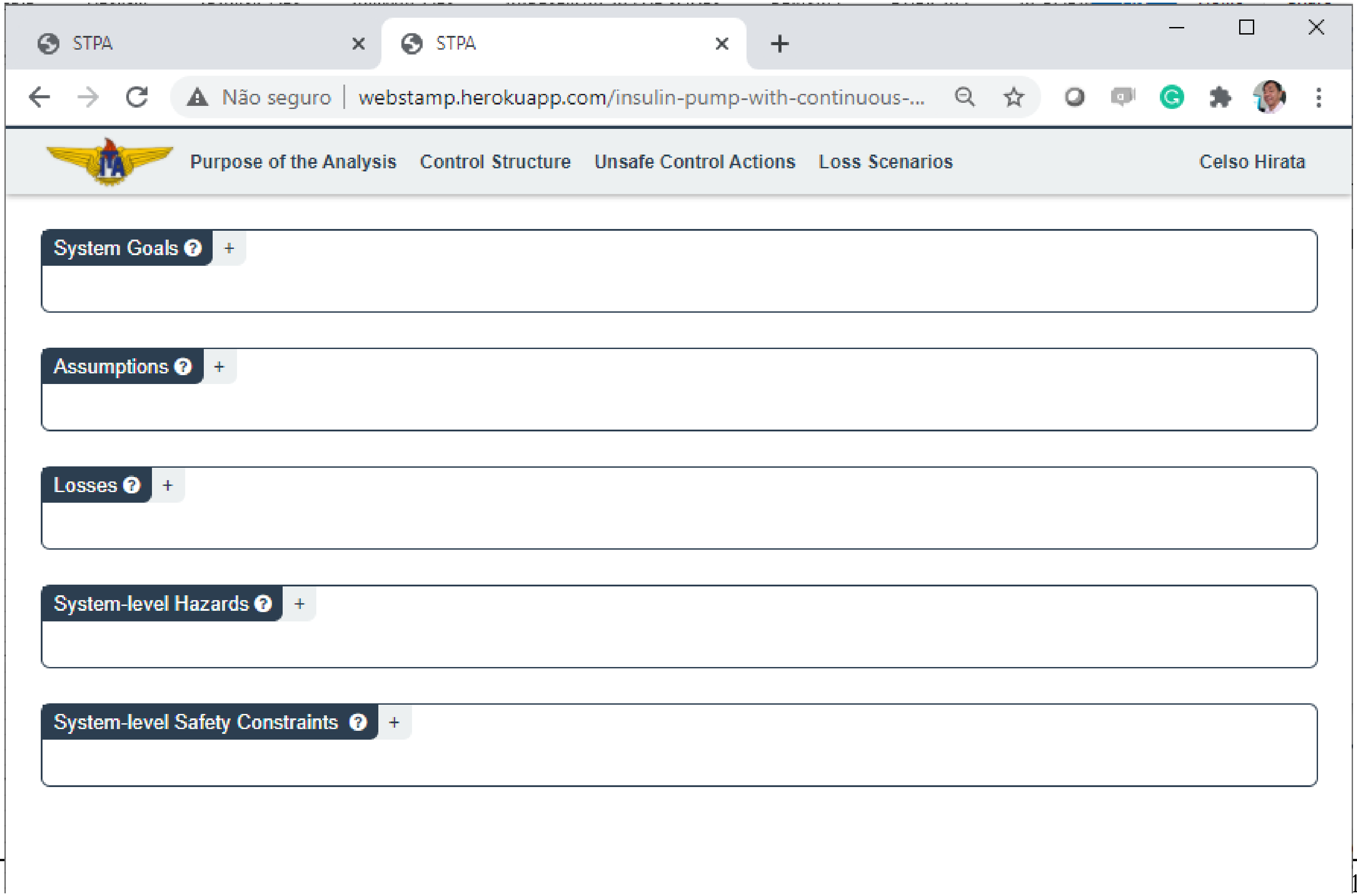
- Click on the “Select Project” button to be redirect to the tab “**Purpose of the Analysis**”.



Insulin pump with continuous glucose monitor

WebSTAMP

- Has four tabs that are related to the STPA steps.
- We are going to work on tab “**Purpose of the Analysis**”.



Purpose of the analysis

- This step has five parts: System Goals, Assumptions, Losses, System-Level Hazards, and System-level Safety Constraints.
- In the next slides we provide the information that you need to complete this step.
- You have to enter the information **sequentially**: System Goals, Assumptions, Losses, System-Level Hazards, and System-level Safety Constraints.
- In case of error, you should undo (exclude) up to the correct information and resume. WebSTAMP still has bugs. So, be careful.

Purpose of the analysis

The goals of the system are:

G-1: monitor the patient's glucose level,

G-2: control the injection of insulin, and

G-3: provide alerts about the system's operation.

Assumptions

The list of assumptions for Insulin pump with continuous glucose monitor are (it is not necessary to put “A-1”, “A-2”, etc. WebSTAMP creates the indexes automatically):

A-1: The system operates with a smartphone with Internet connection..

A-2: The smartphone has an app to aid the control.

Losses

The losses for Insulin pump with continuous glucose monitor are:

L-1: Patient is injured or killed from overdose or underdose.

L-2: Loss of the manufacturer's credibility.

L-3: Loss of personal information (e.g. level of glucose, amount of glucose, and etc.).

Hazards

To enter with the **Hazards** information, you need to enter previously the **Losses** information.

The hazards and **the associated losses** are:

H-1: Pumping insulin when glucose level is going down – hypoglycemia **[L-1]**

H-2: Not pumping insulin when glucose level is going up – hyperglycemia **[L-1]**

H-3: System in operation with battery or reservoir level below recommended values. **[L-1]**

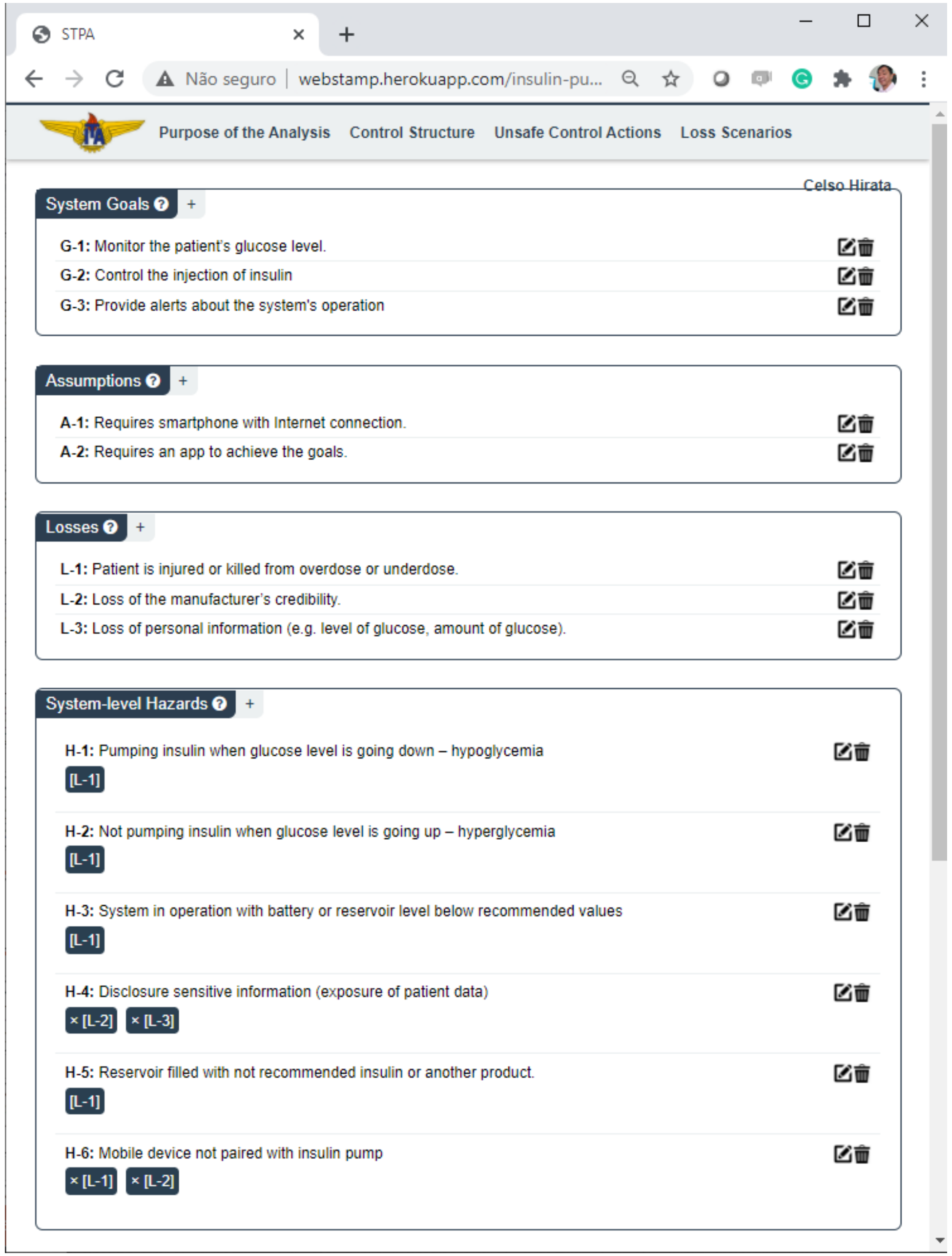
H-4: Disclosure sensitive information (exposure of patient data) **[L-2] [L-3]**

H-5: Reservoir filled with not recommended insulin or another product. **[L-1]**

H-6: Mobile device not paired with insulin pump. **[L-1] [L-2]**

After entering the Hazards information, you should have the following window.

In case of error, you should undo (exclude) up to the correct information and resume. So, be careful.



System Goals +

G-1: Monitor the patient's glucose level.

G-2: Control the injection of insulin

G-3: Provide alerts about the system's operation

Assumptions +

A-1: Requires smartphone with Internet connection.

A-2: Requires an app to achieve the goals.

Losses +

L-1: Patient is injured or killed from overdose or underdose.

L-2: Loss of the manufacturer's credibility.

L-3: Loss of personal information (e.g. level of glucose, amount of glucose).

System-level Hazards +

H-1: Pumping insulin when glucose level is going down – hypoglycemia
[L-1]

H-2: Not pumping insulin when glucose level is going up – hyperglycemia
[L-1]

H-3: System in operation with battery or reservoir level below recommended values
[L-1]

H-4: Disclosure sensitive information (exposure of patient data)
× [L-2] × [L-3]

H-5: Reservoir filled with not recommended insulin or another product.
[L-1]

H-6: Mobile device not paired with insulin pump
× [L-1] × [L-2]

WebSTAMP

Insulin Pump

→ Purpose of the Analysis

Control Structure

Unsafe Control Actions

Loss Scenarios

System-level safety constraints

To enter with the **Constraints** information, you need to enter previously the **Hazards** information.

The list of System-level Safety Constraints and **the associated hazards** are:

SSC-1: The pumping of insulin must be stopped when the glucose level goes below a configurable minimum level (for both Bolus and Basal). **[H-1]**

SSC-2: The system must automatically start pumping insulin after reaching some maximum configurable level **[H-2]**

System-level safety constraints (cont.)

SSC-3: The system must send an alert when detects a battery or reservoir level near the below level **[H-3]**

SSC-4: The system must never exposure patient data without the patient's consent. **[H-4]**

SSC-5: Reservoir must be filled only with the recommended insulin. **[H-5]**

SSC-6: Mobile device always must be paired with insulin pump. **[H-6]**

Model the control structure

- Unfortunately, WebSTAMP does not have a user-friendly graphical user interface to model the control structure **yet**. Therefore, the model information is entered in text boxes.
- Components are Actuators, Controlled Process, Controllers or Sensors.
- Click on the tab “Control Structure”

Model the control structure

WebSTAMP

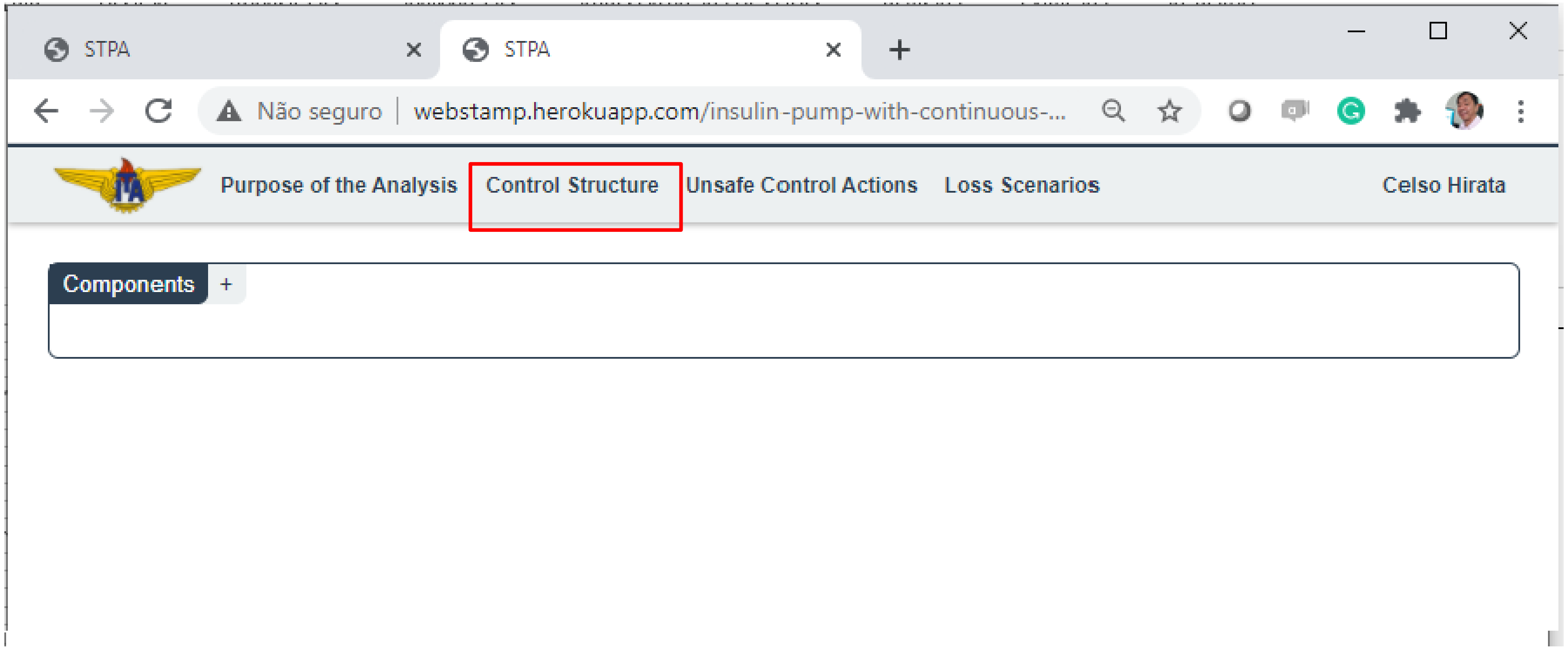
Insulin Pump

Purpose of the Analysis

→ Control Structure

Unsafe Control Actions

Loss Scenarios



Model the control structure

- Before we demonstrate how WebSTAMP deals textually with the functional control structure, we present its graphical version (made manually).
- The “Insulin pump with continuous glucose monitor” has 5 (five) components: Patient, Mobile Device and App, Apple Store/Google Play, Insulin Pump, and Patient’s Body.
- Patient’s Body is a Controlled Process and the other components are Controllers (in fact, Apple Store and Google Play are external systems).

WebSTAMP

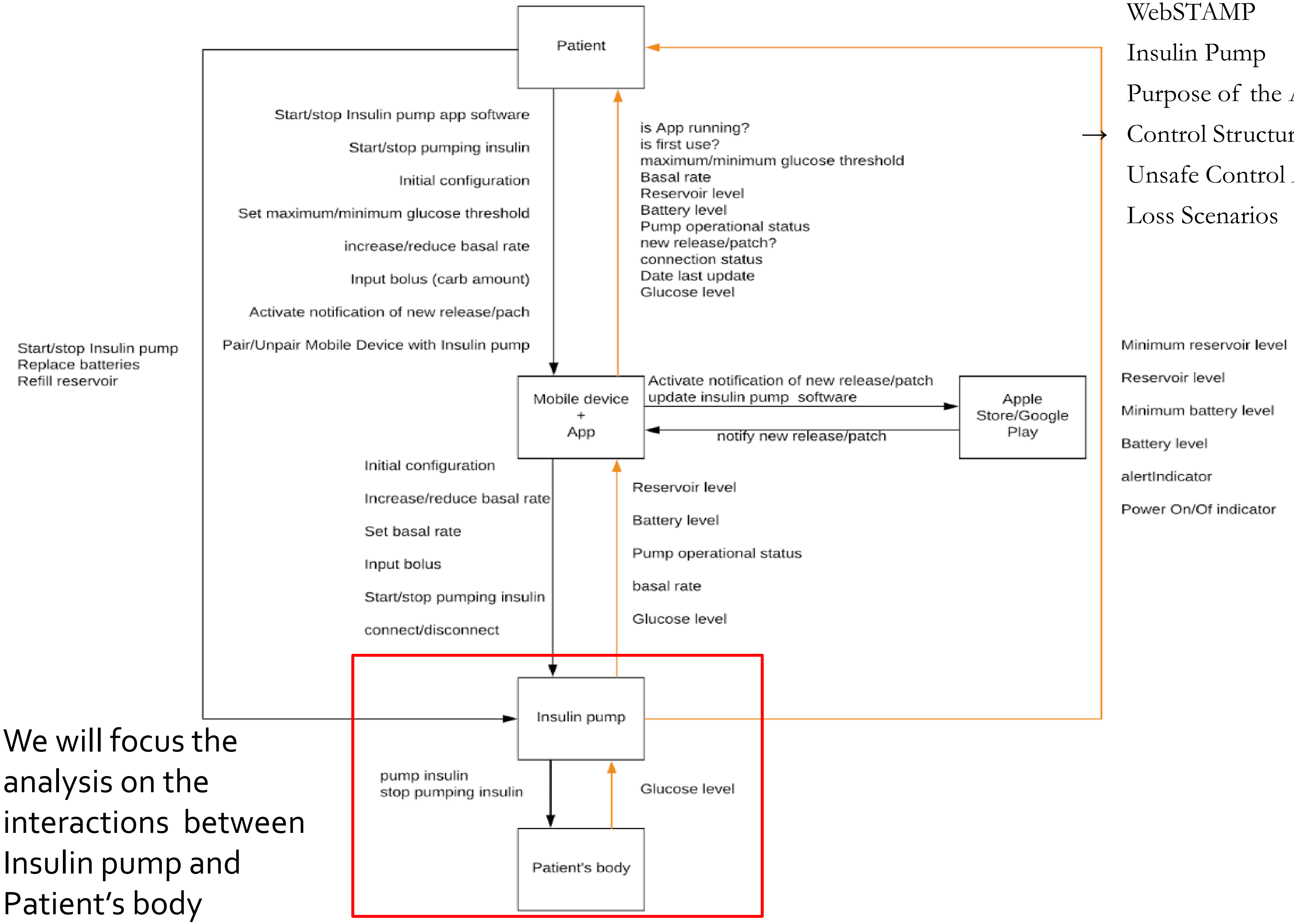
Insulin Pump

Purpose of the Analysis

→ Control Structure

Unsafe Control Actions

Loss Scenarios



Patient

Start/stop Insulin pump app software

Start/stop pumping insulin

Initial configuration

Set maximum/minimum glucose threshold

increase/reduce basal rate

Input bolus (carb amount)

Activate notification of new release/pach

Pair/Unpair Mobile Device with Insulin pump

is App running?
is first use?
maximum/minimum glucose threshold
Basal rate
Reservoir level
Battery level
Pump operational status
new release/patch?
connection status
Date last update
Glucose level

Start/stop Insulin pump
Replace batteries
Refill reservoir

Mobile device
+
App

Activate notification of new release/patch
update insulin pump software

notify new release/patch

Apple
Store/Google
Play

Minimum reservoir level

Reservoir level

Minimum battery level

Battery level

alertIndicator

Power On/Of indicator

Initial configuration

Increase/reduce basal rate

Set basal rate

Input bolus

Start/stop pumping insulin

connect/disconnect

Reservoir level

Battery level

Pump operational status

basal rate

Glucose level

Insulin pump

pump insulin
stop pumping insulin

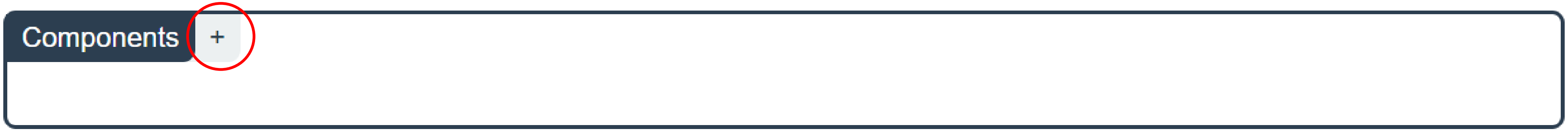
Glucose level

Patient's body

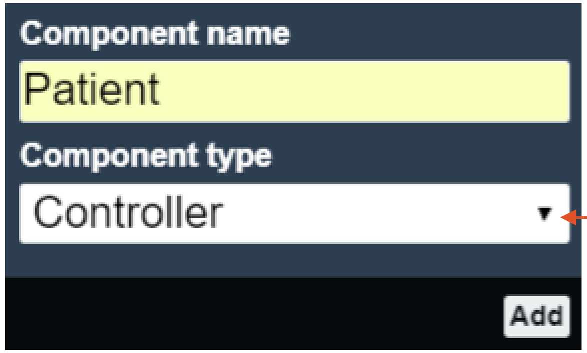
We will focus the analysis on the interactions between Insulin pump and Patient's body

Model the control structure

- To add a new Component in WebSTAMP, click on the "+" button.



- The following window will popup. Just insert the component name and select the type "Controller" (Actuator, Controlled Process, Controller or Sensor).


Component name
Patient
Component type
Controller ▾

Select the Component type here!

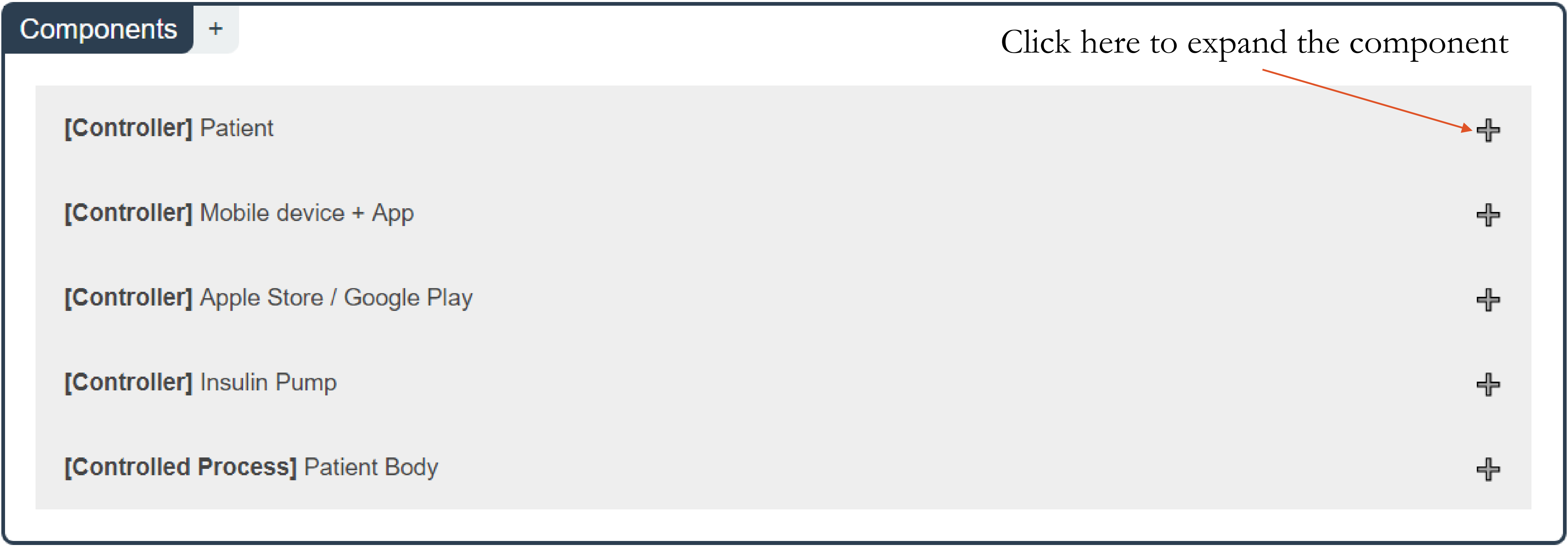
Model the control structure

- Add the Components:

Component Name	Type
Patient	Controller
Mobile Device and App	Controller
Apple Store / Google Play	Controller
Insulin Pump	Controller
Patient’s Body	Controlled Process

Model the control structure

- After adding the components, you will see the window:



Click here to expand the component

[Controller] Patient

[Controller] Mobile device + App

[Controller] Apple Store / Google Play

[Controller] Insulin Pump

[Controlled Process] Patient Body

Model the control structure

- Notice that when you expand a component, all of them have a field called "Connections". Here, you have to define the connections between components i.e. the outgoing arcs.
- You don't need to specify the incoming arcs – after adding all the components, just refresh the page and WebSTAMP will do it for you.

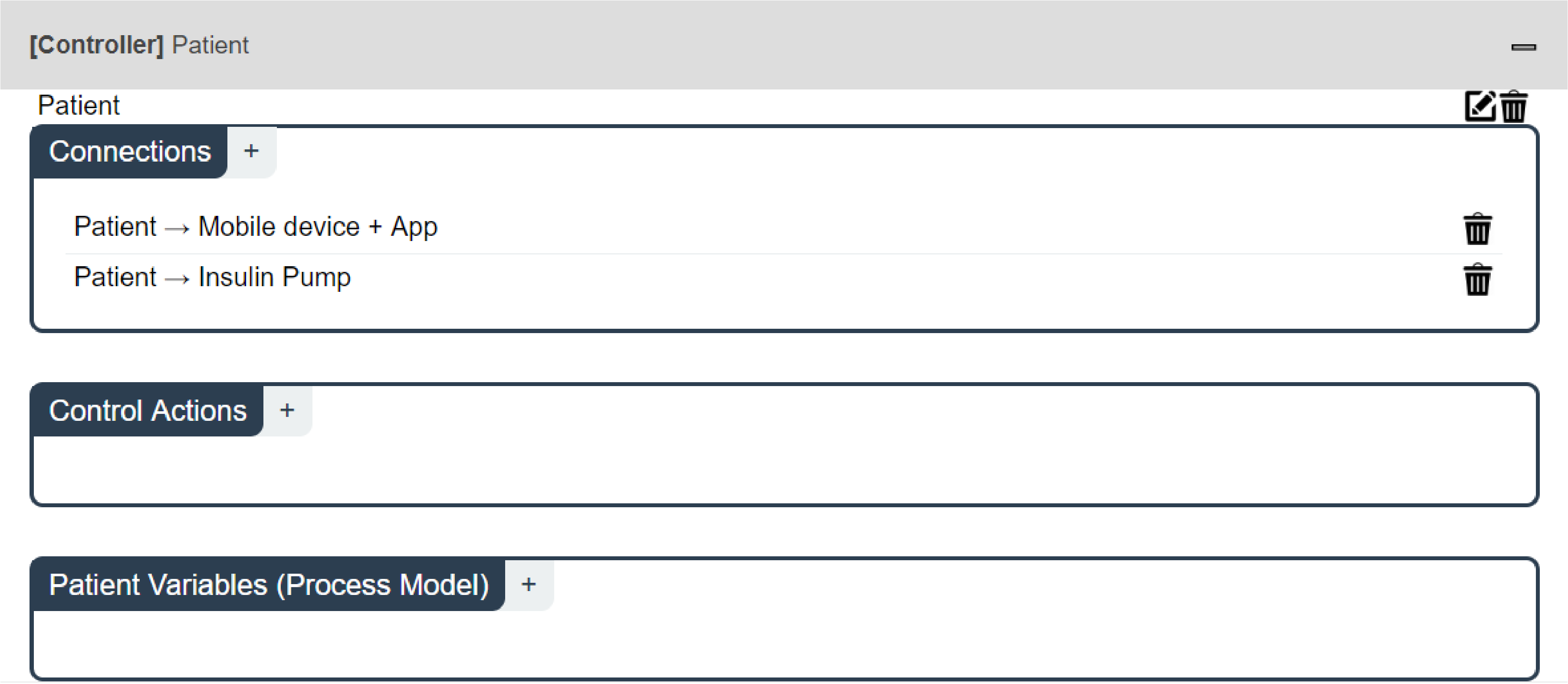
Model the control structure

- Add the connections:

Component Source	Component Destiny
Patient	Mobile Device + App
Patient	Insulin Pump
Mobile Device + App	Apple Store / Google Play
Insulin Pump	Patient’s Body

Model the control structure

- The connections for the “**Patient**” should be:



[Controller] Patient

Patient

Connections +

Patient → Mobile device + App

Patient → Insulin Pump

Control Actions +

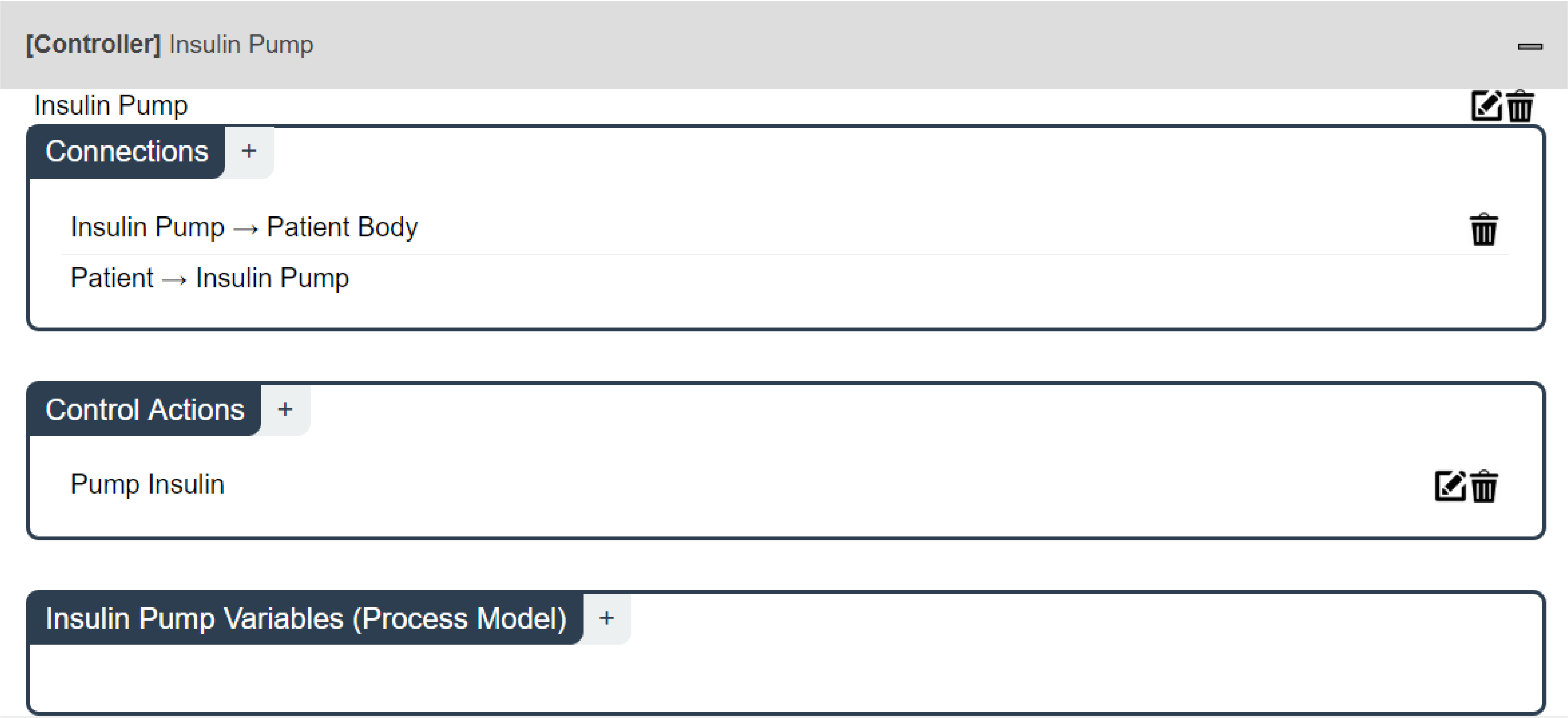
Patient Variables (Process Model) +

WebSTAMP
Insulin Pump
Purpose of the Analysis
→ Control Structure
Unsafe Control Actions
Loss Scenarios

Model the control structure

- Now, we must define the control Actions. For now, we can “ignore” the Controllers: **Patient**, **Mobile Device + App** and **Apple Store / Google Play**.
- The definitions of these components are necessary for a complete STPA analysis. For the time being, we “suppress” the control actions of these components.
- Insert the control action “**Pump Insulin**” for the **“Insulin Pump”** controller.

Model the control structure



[Controller] Insulin Pump

Insulin Pump

Connections +

Insulin Pump → Patient Body

Patient → Insulin Pump

Control Actions +

Pump Insulin

Insulin Pump Variables (Process Model) +

Model the control structure

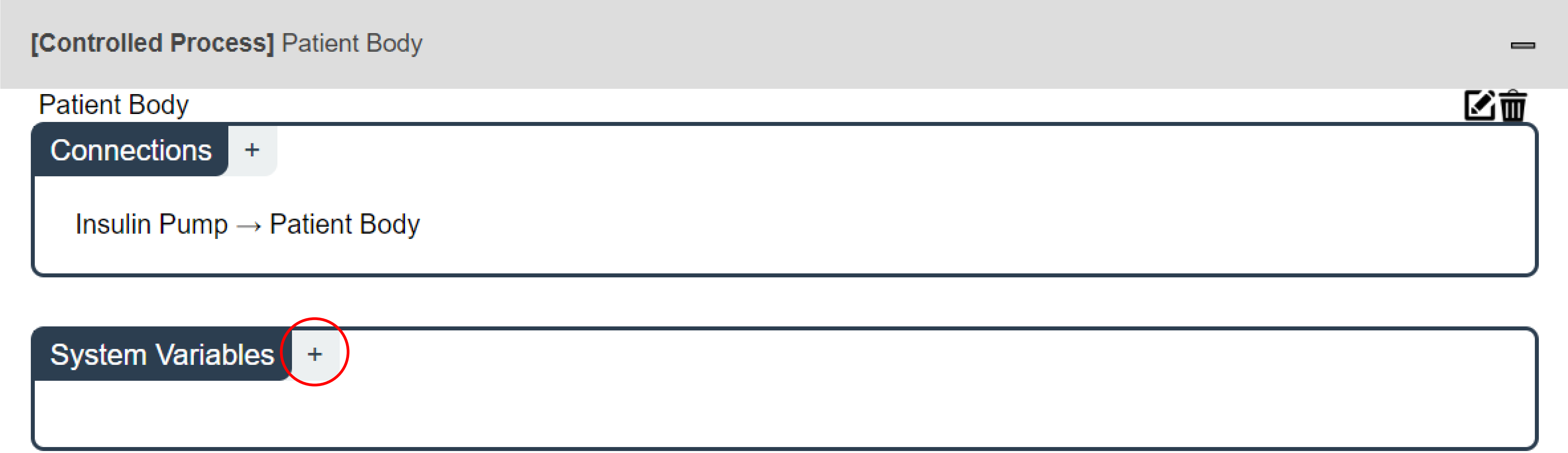
- It is time to define the variables. There are two types of variables:
  - Controlled Process Variables – Variables whose values come from the controlled process (obtained by feedback).
  - Other Variables – Variables whose values come from other entities or environment.
- **We will define the controlled process variables first**. They will be defined in the Patient’s Body controlled process.

Model the control structure

- The controlled process “**Patient’s Body”** has only a variable: “**Glucose level”**.
- Glucose level is a positive integer. For convenience, we can describe glucose level using three values: Below (lower than 50 mg/dL) Normal (Between 50 mg/dL and 140 mg/dL) and Above (higher than 140 mg/dL).
  - These bounds are not the same for everybody. We must take in account if the person is fed or in fasting, the type of diabetes, etc.
  - We will work with ranges of values.

Model the control structure

- To add a new variable, expand the “[Controlled Process] Patient’s Body” and click in the “+” button next to the label “System Variables”.



[Controlled Process] Patient Body

Patient Body

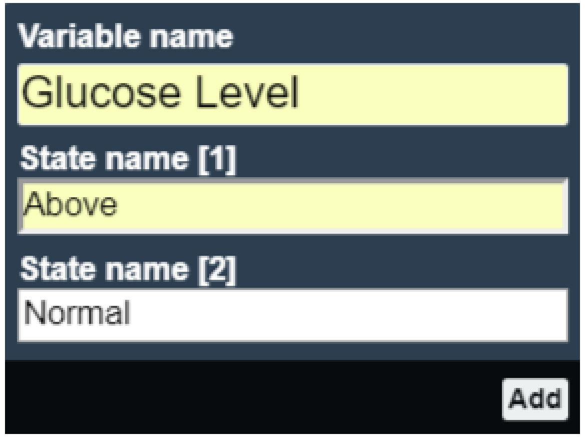
Connections +

Insulin Pump → Patient Body

System Variables +

Model the control structure

- The following window will popup. Note that we define “Glucose level” with three values (or states), but WebSTAMP initially provides interface for only two values. Provide “Above” and “Normal”. We will add “Below” value soon.



Variable name: Glucose Level

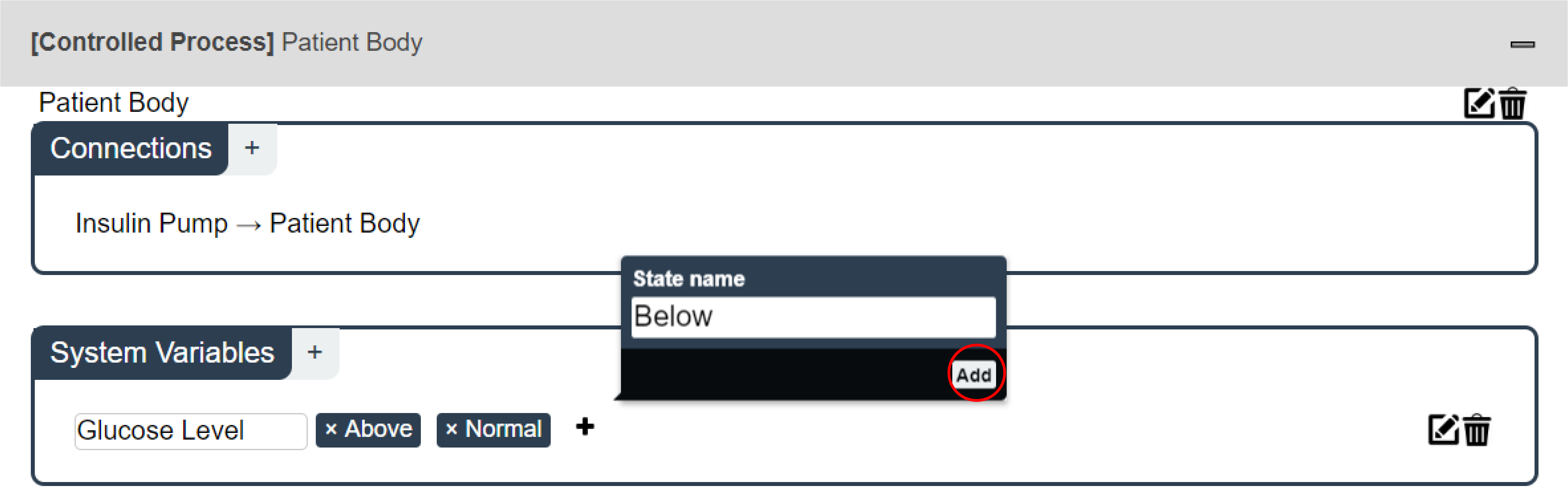
State name [1]: Above

State name [2]: Normal

Model the control structure

WebSTAMP
Insulin Pump
Purpose of the Analysis
→ Control Structure
Unsafe Control Actions
Loss Scenarios

- To add a third (or more) value, follow the steps below:



[Controlled Process] Patient Body

Patient Body

Connections +

Insulin Pump → Patient Body

System Variables +

Glucose Level × Above × Normal +

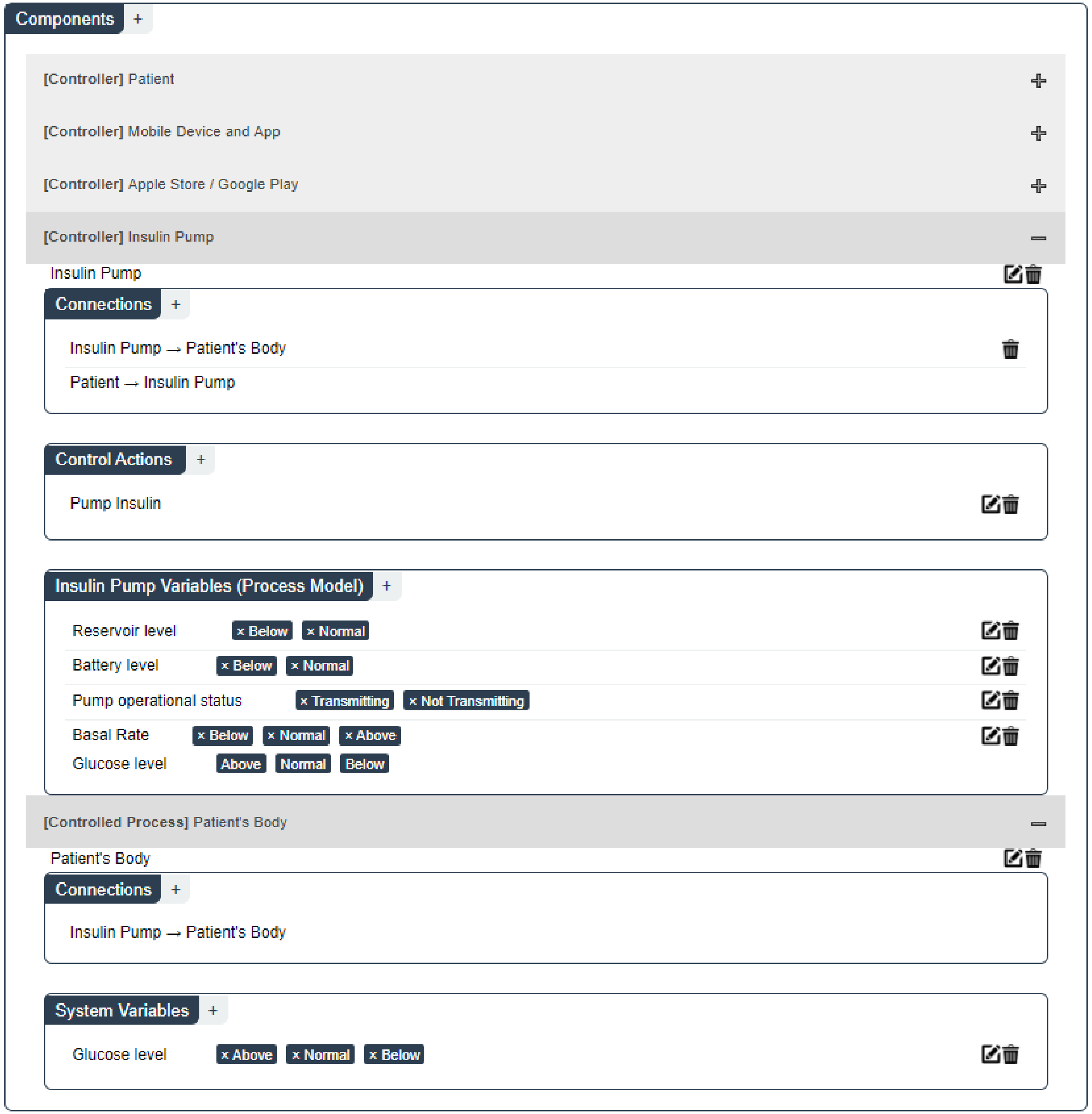
State name: Below

Model the control structure

- In the same way, add the following variables and values for the “**Insulin Pump”** controller:

Variables	Suggestion of States (Values)
Reservoir level	[Below / Normal]
Battery level	[Below / Normal]
Pump operational status	[Transmitting / Not Transmitting]
Basal Rate	[Below / Normal / Above]

The control structure so far …



Components +

[Controller] Patient +

[Controller] Mobile Device and App +

[Controller] Apple Store / Google Play +

[Controller] Insulin Pump —

Insulin Pump

Connections +

Insulin Pump → Patient's Body

Patient → Insulin Pump

Control Actions +

Pump Insulin

Insulin Pump Variables (Process Model) +

Reservoir level	× Below × Normal
Battery level	× Below × Normal
Pump operational status	× Transmitting × Not Transmitting
Basal Rate	× Below × Normal × Above
Glucose level	Above Normal Below

[Controlled Process] Patient's Body —

Patient's Body

Connections +

Insulin Pump → Patient's Body

System Variables +

Glucose level	× Above × Normal × Below

Model the control structure

- For the time being, we will focus on the interactions between "Insulin Pump" and "Patient's Body" so we won't complete the Control Structure step.
- After completing partially the Components (the Model Control Structure step), the next step is to analyze, for each Control Action defined in the Control Structure, which context is unsafe.
- Click on the tab "Identify unsafe control actions" in the top menu.

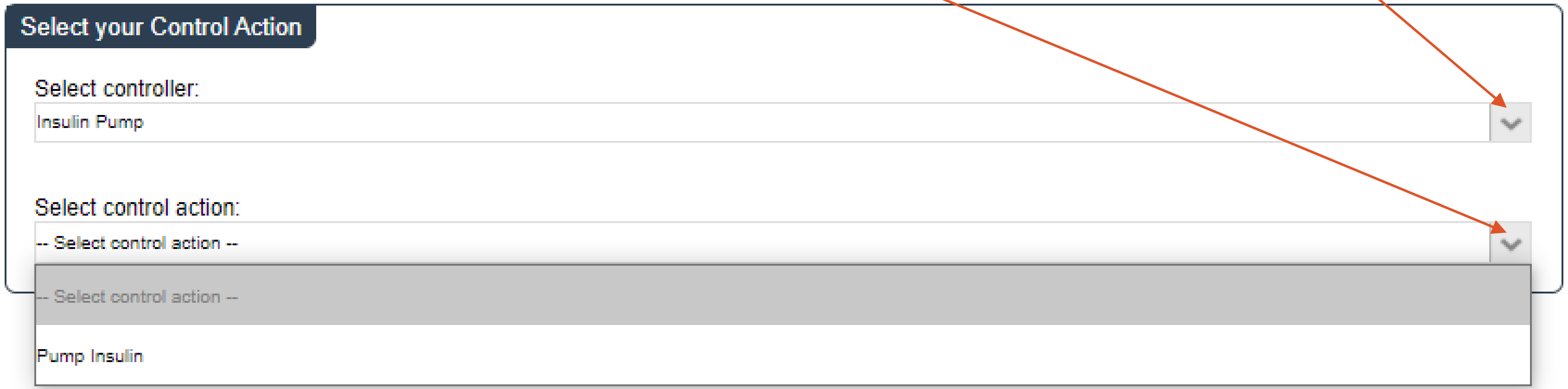
Identify unsafe control actions

- WebSTAMP deals with the step Identify unsafe control actions using the Context Table (Thomas [4]) and the rule-based approach (Gurgel et al. [5]).
- You have to select which control action you want to analyze. As we have only one control action, “Pump Insulin” issued by the “Insulin Pump” controller, it will be selected.

Identify unsafe control actions

1. In the tab “Identify unsafe control action”, you have to select the controller “Insulin Pump”.

2. Afterwards, you select the control action “Pump Insulin”.



Select your Control Action

Select controller: Insulin Pump

Select control action: -- Select control action --

Identify unsafe control actions

- Context table is the combination of all process model variables and their states (values).
- For the control action "Pumping Insulin" of the "Insulin Pump" controller, the table is the combination of all states of the variables: Glucose Level, Reservoir Level, Battery Level, Pump operational status and Basal rate.
- The number of rows of the table is the multiplication of **numbers of values** of Glucose level (3), Reservoir level (2), Battery level (2), Pump operational status (2), and Basal Rate (3), which results in **72 rows**.

WebSTAMP

Insulin Pump

Purpose of the Analysis

Control Structure

→ Unsafe Control Actions

Loss Scenarios

Identify unsafe control actions

Glucose level	Reservoir level	Battery level	Pump operational status	Basal rate
Below	Below	Below	Transmitting	Below
Below	Below	Below	Transmitting	Normal
Below	Below	Below	Transmitting	Above
Below	Below	Below	Not transmitting	Below
Below	Below	Below	Not transmitting	Normal
Below	Below	Below	Not transmitting	Above
Below	Below	Normal	Transmitting	Below
Below	Below	Normal	Transmitting	Normal
Below	Below	Normal	Transmitting	Above

Example of a Context Table – showing the first 9 rows of 72

Identify unsafe control actions

- Context table is a systematic way to find the hazardous control actions, because the analyst can verify all the possibilities.
- It is also a problem – for Pump Insulin control action, the analyst must check 72 contexts for "Pumping Insulin" control action.
- The analyst must verify for each control action and each context, and **each type of provision** of a control action.
- The types of provision are: Provided, Not provided, Provided in wrong order, Provided too early, Provided too late, Stopped too soon, and Applied to long. Seven types of provision for 72 contexts result in **504 situations** to be analyzed!

Identify unsafe control actions

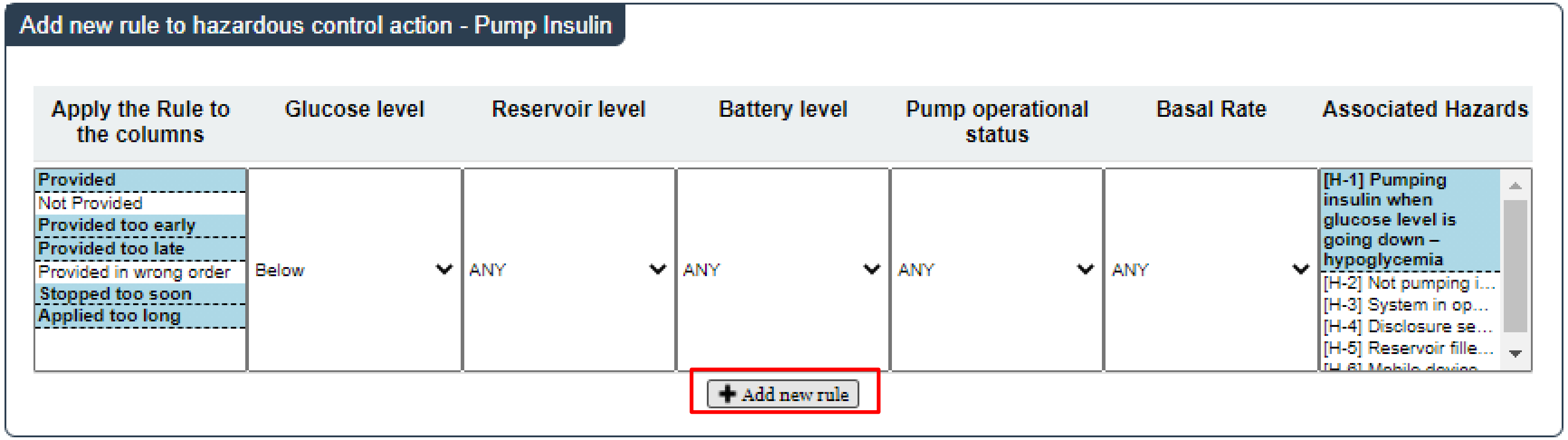
- The analysis requires a large amount of effort, because it is often a repetitive work (contexts are similar to each other!).
- The approach proposed by Gurgel et al. aim to help analysts to find hazardous contexts automatically using rules. The main purpose of creating rules is to automatically assign hazardous contexts in the Context Table.
- The use of rules is not required, but it is useful to save time and effort.

Identify unsafe control actions

- An example of rule is:
  - Whenever the **Glucose Level is below** and **Reservoir Level is ANY** and **Battery level is ANY** and **Pump operational status is ANY** and **Basal Rate is ANY** for the columns: **Provided**, **Provided too early**, **Provided too late**, **Stopped too soon** and **Applied too long**, providing the control Action Pump Insulin is hazardous.
    - The **ANY** keyword is used to specify that any value (state) of that variable is considered (e.g., the variables Reservoir level, Battery level, Pump operational status and Basal rate can have ANY value, but Glucose level must be Below to be hazardous).

Identify unsafe control actions

- Note that the mentioned rule can be created in the following way. You have to select the associated hazards



Add new rule to hazardous control action - Pump Insulin

Apply the Rule to the columns	Glucose level	Reservoir level	Battery level	Pump operational status	Basal Rate	Associated Hazards
Provided Not Provided Provided too early Provided too late Provided in wrong order Stopped too soon Applied too long	Below	ANY	ANY	ANY	ANY	[H-1] Pumping insulin when glucose level is going down – hypoglycemia [H-2] Not pumping i... [H-3] System in op... [H-4] Disclosure se... [H-5] Reservoir fille... [H-6] Mobile device

- Hint: For "Apply the Rule to the columns", you can press CTRL + mouse left click to select more than one column.

WebSTAMP
Insulin Pump
Purpose of the Analysis
Control Structure
→ Unsafe Control Actions
Loss Scenarios

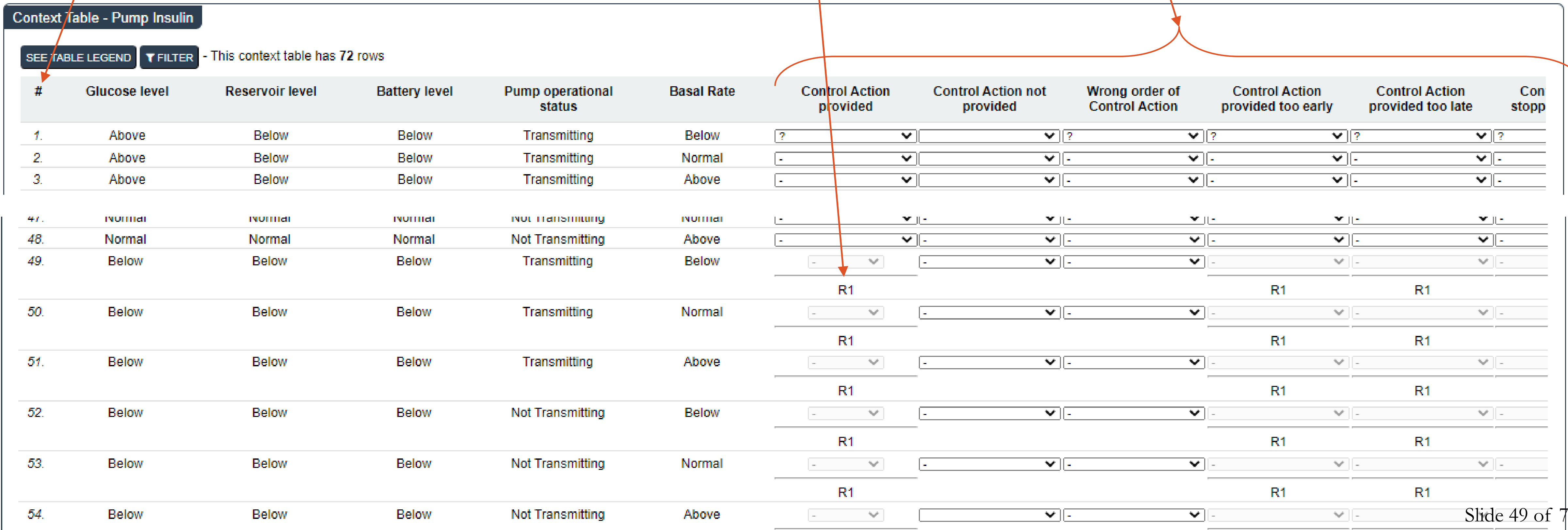
Identify unsafe control actions

The number of the context/row (1 to 72)

Variables of the process model

Index of the created rule (R1, R2, …, R*n*)

Types of provision



Context Table - Pump Insulin

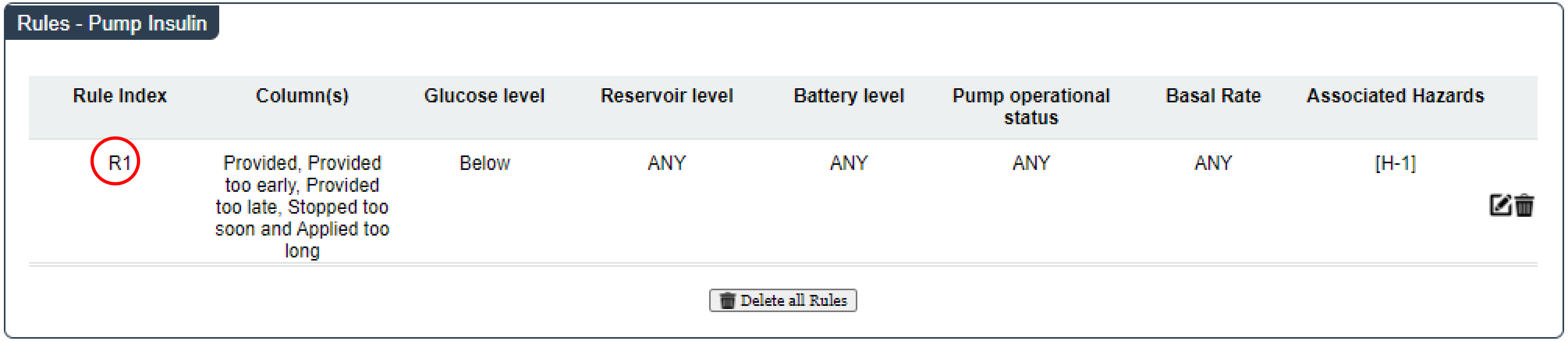
SEE TABLE LEGEND | FILTER - This context table has **72** rows

#	Glucose level	Reservoir level	Battery level	Pump operational status	Basal Rate	Control Action provided	Control Action not provided	Wrong order of Control Action	Control Action provided too early	Control Action provided too late	Con stopp
1.	Above	Below	Below	Transmitting	Below	?		?	?	?	?
2.	Above	Below	Below	Transmitting	Normal	-		-	-	-	-
3.	Above	Below	Below	Transmitting	Above	-		-	-	-	-
47.	Normal	Normal	Normal	Not Transmitting	Normal	-	-	-	-	-	-
48.	Normal	Normal	Normal	Not Transmitting	Above	-	-	-	-	-	-
49.	Below	Below	Below	Transmitting	Below	- R1	-	-	- R1	- R1	-
50.	Below	Below	Below	Transmitting	Normal	- R1	-	-	- R1	- R1	-
51.	Below	Below	Below	Transmitting	Above	- R1	-	-	- R1	- R1	-
52.	Below	Below	Below	Not Transmitting	Below	- R1	-	-	- R1	- R1	-
53.	Below	Below	Below	Not Transmitting	Normal	- R1	-	-	- R1	- R1	-
54.	Below	Below	Below	Not Transmitting	Above	-		-	-	-	

WebSTAMP
Insulin Pump
Purpose of the Analysis
Control Structure
→ Unsafe Control Actions
Loss Scenarios

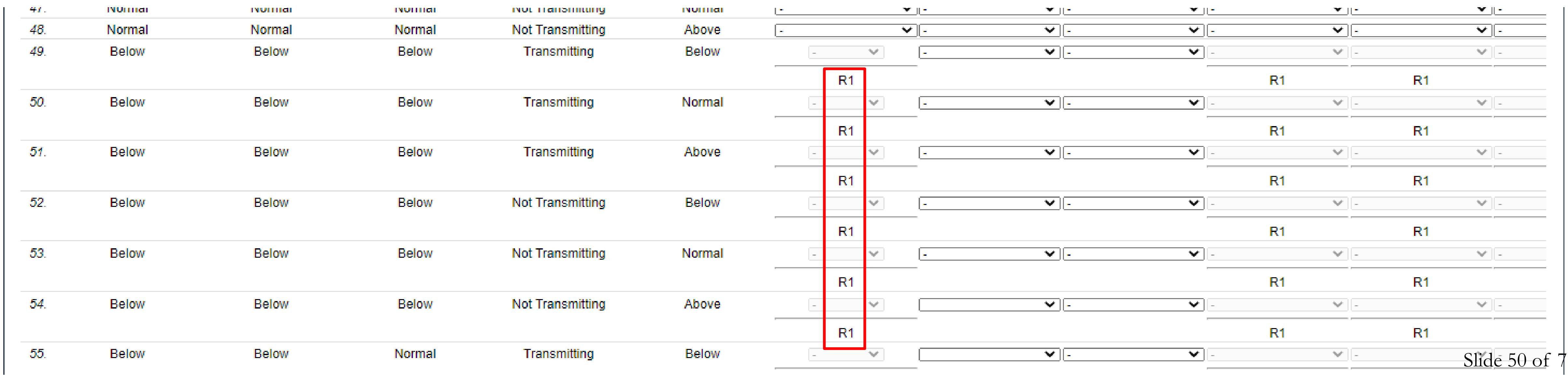
Identify unsafe control actions

- Example of rule “R1” applied in Context Table.



Rules - Pump Insulin

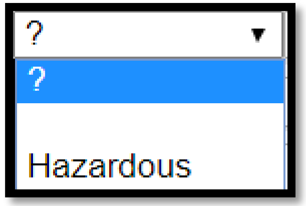
Rule Index	Column(s)	Glucose level	Reservoir level	Battery level	Pump operational status	Basal Rate	Associated Hazards
R1	Provided, Provided too early, Provided too late, Stopped too soon and Applied too long	Below	ANY	ANY	ANY	ANY	[H-1]



47.	Normal	Normal	Normal	Not Transmitting	Normal	-	-	-	-	-	-
48.	Normal	Normal	Normal	Not Transmitting	Above	-	-	-	-	-	-
49.	Below	Below	Below	Transmitting	Below	- R1	-	-	- R1	- R1	-
50.	Below	Below	Below	Transmitting	Normal	- R1	-	-	- R1	- R1	-
51.	Below	Below	Below	Transmitting	Above	- R1	-	-	- R1	- R1	-
52.	Below	Below	Below	Not Transmitting	Below	- R1	-	-	- R1	- R1	-
53.	Below	Below	Below	Not Transmitting	Normal	- R1	-	-	- R1	- R1	-
54.	Below	Below	Below	Not Transmitting	Above	- R1		-	- R1	- R1	-
55.	Below	Below	Normal	Transmitting	Below	-		-	-	-	

Identify unsafe control actions

- We must complete the Context Table to discover the hazardous control actions through hazardous scenarios. Each element in context table has three values:
  - "**?**", that means the context for that type of provision was not analyzed yet.
  - " " (*blank*), that means the context for that type of provision is not hazardous.
  - "**Hazardous**", that means the context for that type of provision is hazardous.



Identify unsafe control actions

- After finishing the analysis of the context table (recommended, but not required), it is time to define the hazardous (unsafe) control actions.
- There are three ways to define a hazardous control action:
  1. When you create a rule.
  2. When you mark a context with "Hazardous" in a cell (to be implemented).
  3. Defining manually the context of hazardous control action.

WebSTAMP

Insulin Pump

Purpose of the Analysis

Control Structure

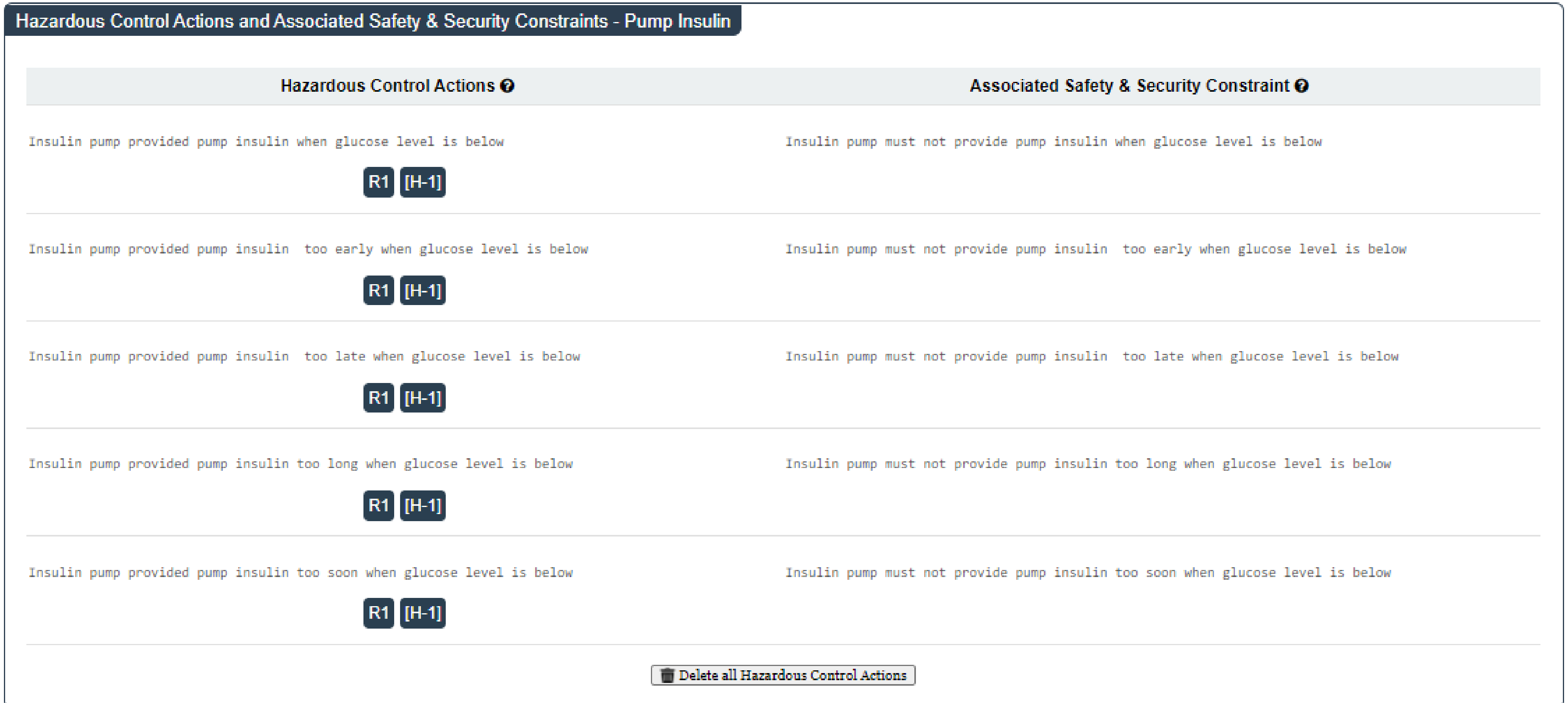
→ Unsafe Control Actions

Loss Scenarios

Identify unsafe control actions

1. When you create a rule

- Notice that for the Rule “R1”, five hazardous control actions were automatically defined.



Hazardous Control Actions and Associated Safety & Security Constraints - Pump Insulin

Hazardous Control Actions	Associated Safety & Security Constraint
Insulin pump provided pump insulin when glucose level is below R1 [H-1]	Insulin pump must not provide pump insulin when glucose level is below
Insulin pump provided pump insulin too early when glucose level is below R1 [H-1]	Insulin pump must not provide pump insulin too early when glucose level is below
Insulin pump provided pump insulin too late when glucose level is below R1 [H-1]	Insulin pump must not provide pump insulin too late when glucose level is below
Insulin pump provided pump insulin too long when glucose level is below R1 [H-1]	Insulin pump must not provide pump insulin too long when glucose level is below
Insulin pump provided pump insulin too soon when glucose level is below R1 [H-1]	Insulin pump must not provide pump insulin too soon when glucose level is below

WebSTAMP

Insulin Pump

Purpose of the Analysis

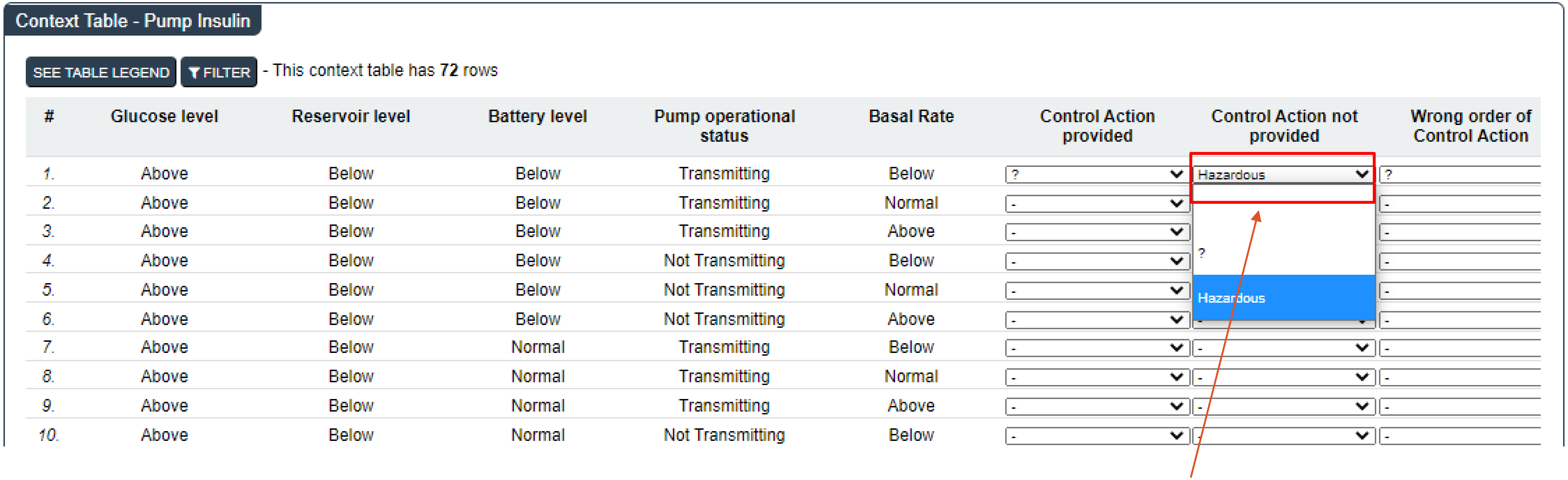
Control Structure

→ Unsafe Control Actions

Loss Scenarios

Identify unsafe control actions

2. When you mark a context with “Hazardous”



Context Table - Pump Insulin

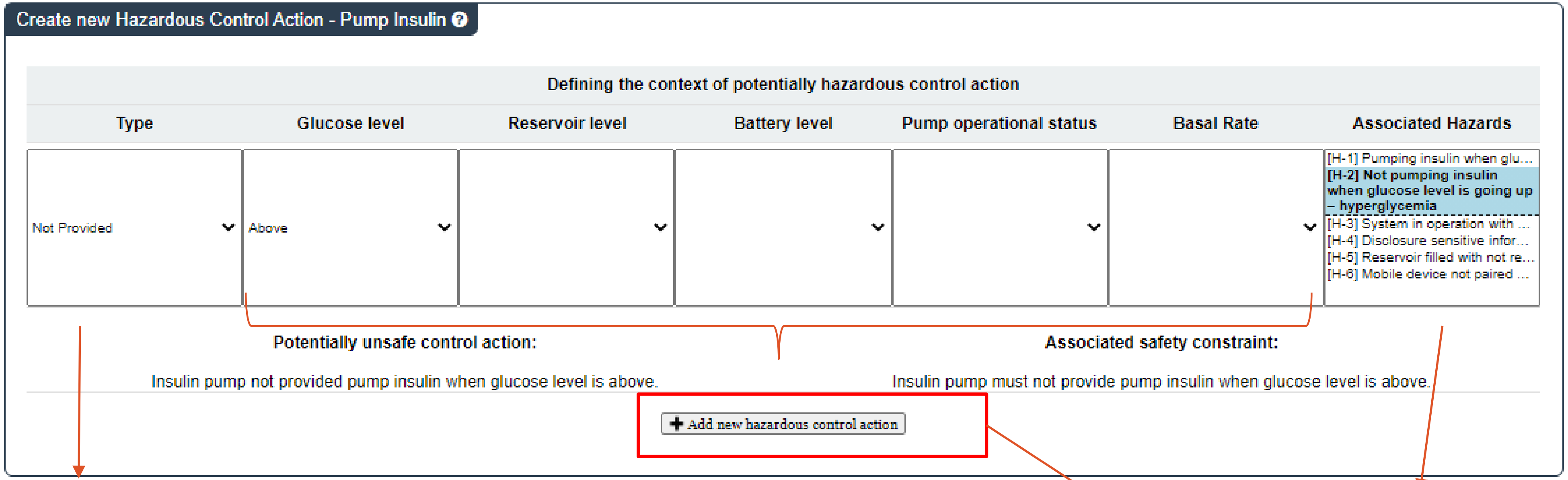
SEE TABLE LEGEND | FILTER - This context table has **72** rows

#	Glucose level	Reservoir level	Battery level	Pump operational status	Basal Rate	Control Action provided	Control Action not provided	Wrong order of Control Action
1.	Above	Below	Below	Transmitting	Below	?	Hazardous	?
2.	Above	Below	Below	Transmitting	Normal	-		-
3.	Above	Below	Below	Transmitting	Above	-		-
4.	Above	Below	Below	Not Transmitting	Below	-		-
5.	Above	Below	Below	Not Transmitting	Normal	-		-
6.	Above	Below	Below	Not Transmitting	Above	-	-	-
7.	Above	Below	Normal	Transmitting	Below	-	-	-
8.	Above	Below	Normal	Transmitting	Normal	-	-	-
9.	Above	Below	Normal	Transmitting	Above	-	-	-
10.	Above	Below	Normal	Not Transmitting	Below	-	-	-

Cell of the Context Table manually assigned!

Identify unsafe control actions

3. Defining manually for the control action pump insulin



Create new Hazardous Control Action - Pump Insulin ?

Defining the context of potentially hazardous control action

Type	Glucose level	Reservoir level	Battery level	Pump operational status	Basal Rate	Associated Hazards
Not Provided	Above					[H-1] Pumping insulin when glu... **[H-2] Not pumping insulin when glucose level is going up – hyperglycemia** [H-3] System in operation with ... [H-4] Disclosure sensitive infor... [H-5] Reservoir filled with not re... [H-6] Mobile device not paired ...

Potentially unsafe control action:

Insulin pump not provided pump insulin when glucose level is above.

Associated safety constraint:

Insulin pump must not provide pump insulin when glucose level is above.

1. Select the type of Provision (Provided, Not provided, too early, etc.)

2. Select your context (when you select one or more variable's values, a text are generated automatically

3. Select the associated hazards.

4. Add the hazardous control action.

WebSTAMP

Insulin Pump

Purpose of the Analysis

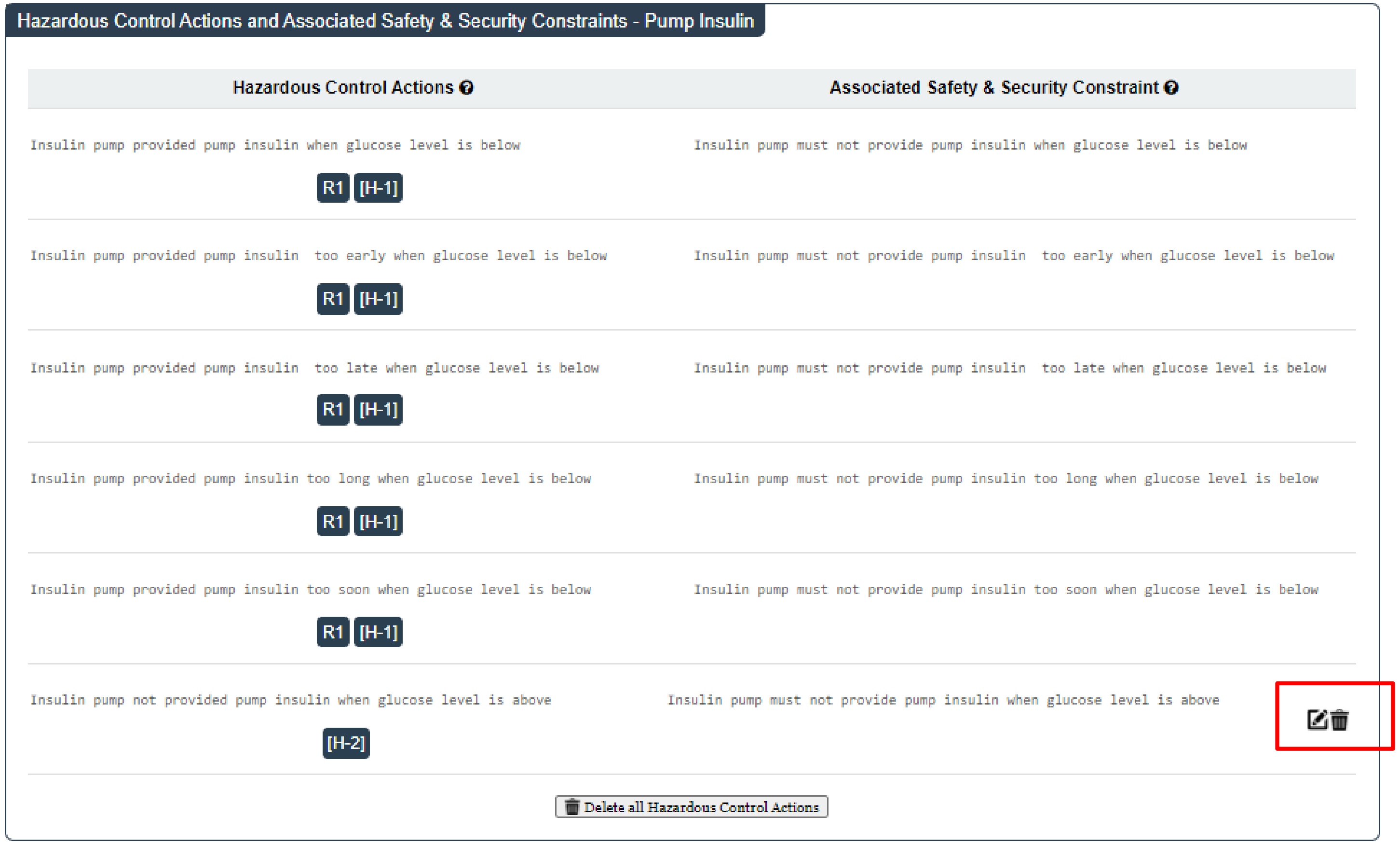
Control Structure

→ Unsafe Control Actions

Loss Scenarios

Identify unsafe control actions

3. Defining manually the hazardous control action (very similar to rule)



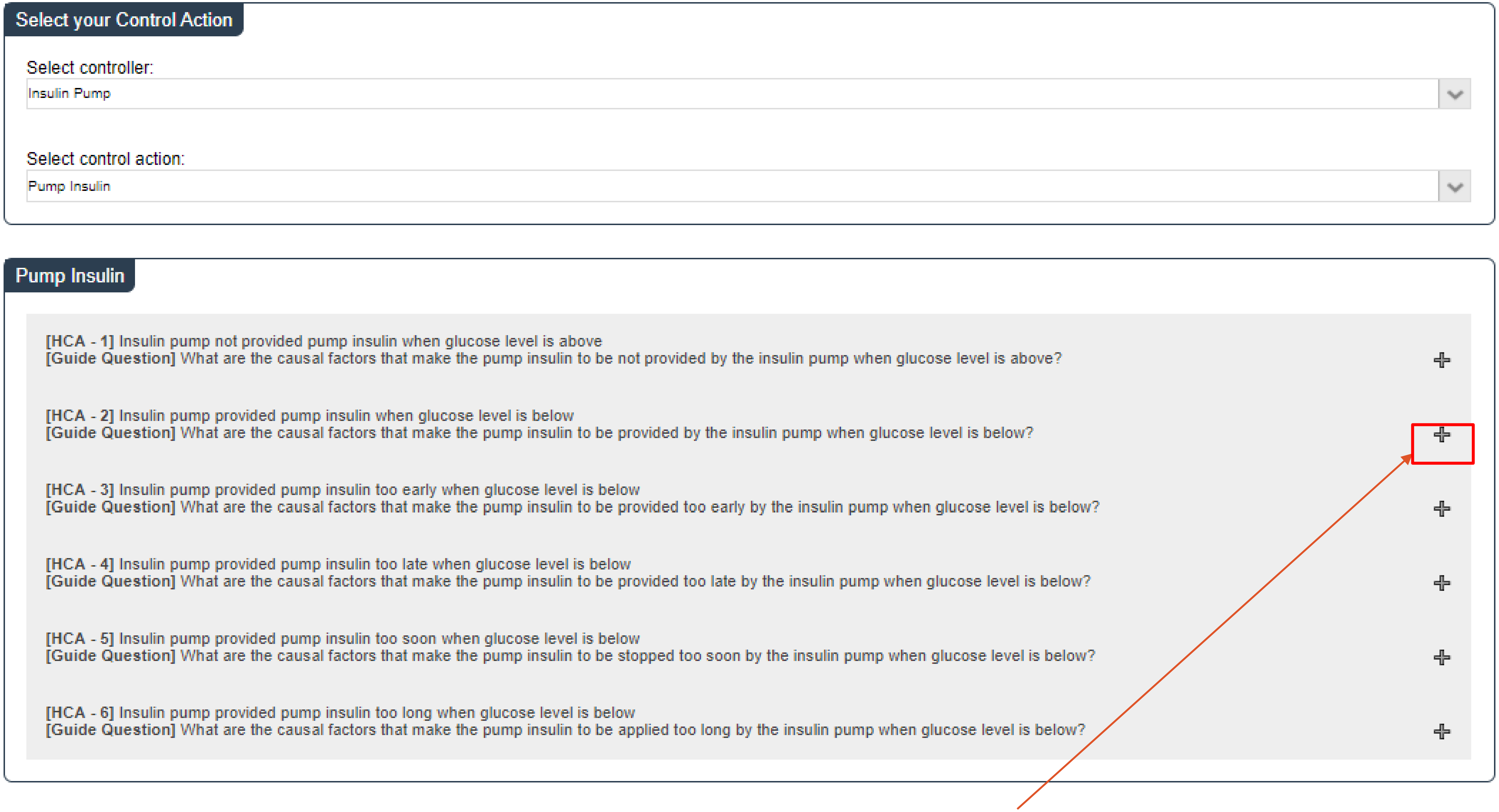
Hazardous Control Actions and Associated Safety & Security Constraints - Pump Insulin

Hazardous Control Actions	Associated Safety & Security Constraint	
Insulin pump provided pump insulin when glucose level is below R1 [H-1]	Insulin pump must not provide pump insulin when glucose level is below	
Insulin pump provided pump insulin too early when glucose level is below R1 [H-1]	Insulin pump must not provide pump insulin too early when glucose level is below	
Insulin pump provided pump insulin too late when glucose level is below R1 [H-1]	Insulin pump must not provide pump insulin too late when glucose level is below	
Insulin pump provided pump insulin too long when glucose level is below R1 [H-1]	Insulin pump must not provide pump insulin too long when glucose level is below	
Insulin pump provided pump insulin too soon when glucose level is below R1 [H-1]	Insulin pump must not provide pump insulin too soon when glucose level is below	
Insulin pump not provided pump insulin when glucose level is above [H-2]	Insulin pump must not provide pump insulin when glucose level is above	✎ 🗑

The manually added hazardous control action can be edited or deleted.

Identify loss scenarios

- The first task in the task Identify loss scenarios is to choose which control (and controller) action will be analyzed.
- When you select a controller and a control action, all the unsafe control actions identified in the previous step (Unsafe Control Actions and Associated Safety Constraints list) are listed.
- Select the controller ‘Insulin pump” and the control action “Pump Insulin”.



Select your Control Action

Select controller:

Insulin Pump

Select control action:

Pump Insulin

Pump Insulin

[HCA - 1] Insulin pump not provided pump insulin when glucose level is above
[Guide Question] What are the causal factors that make the pump insulin to be not provided by the insulin pump when glucose level is above?

[HCA - 2] Insulin pump provided pump insulin when glucose level is below
[Guide Question] What are the causal factors that make the pump insulin to be provided by the insulin pump when glucose level is below?

[HCA - 3] Insulin pump provided pump insulin too early when glucose level is below
[Guide Question] What are the causal factors that make the pump insulin to be provided too early by the insulin pump when glucose level is below?

[HCA - 4] Insulin pump provided pump insulin too late when glucose level is below
[Guide Question] What are the causal factors that make the pump insulin to be provided too late by the insulin pump when glucose level is below?

[HCA - 5] Insulin pump provided pump insulin too soon when glucose level is below
[Guide Question] What are the causal factors that make the pump insulin to be stopped too soon by the insulin pump when glucose level is below?

[HCA - 6] Insulin pump provided pump insulin too long when glucose level is below
[Guide Question] What are the causal factors that make the pump insulin to be applied too long by the insulin pump when glucose level is below?

Select the HCA-2

Identify loss scenarios

- WebSTAMP aids identifying loss scenarios, which are seen as 4-tuples of Scenario, Associated Causal Factor, Recommendation, and Rationale.
- For each hazardous control action, WebSTAMP automatically creates a guide question to help the user to think of possible causal factors.
- Initially, for each hazardous control action, the analyst can choose between two buttons: “Checklist” and “Add new 4-tuple”.

WebSTAMP

Insulin Pump

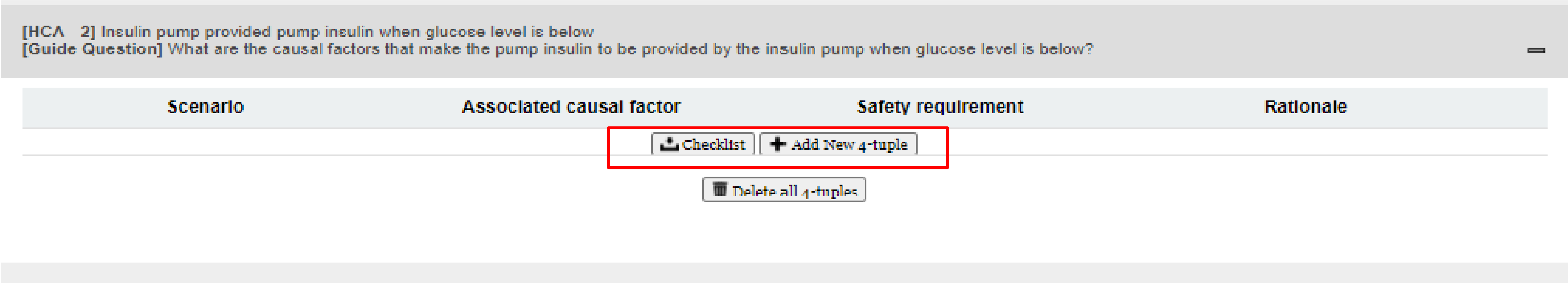
Purpose of the Analysis

Control Structure

Unsafe Control Actions

→ Loss Scenarios

Identify loss scenarios



[HCA 2] Insulin pump provided pump insulin when glucose level is below
[Guide Question] What are the causal factors that make the pump insulin to be provided by the insulin pump when glucose level is below?

Scenario	Associated causal factor	Safety requirement	Rationale

Identify loss scenarios

- The button "**Checklist**" contains a set of generic 4-tuples (we call as 4-tuple the set of elements: **Scenario**, **Associated Causal Factor**, **Recommendation** and **Rationale**) that was previously stored. In general, the tuples are as generic as possible, **therefore, you must tailor them**.
- The tuples of "**Checklist"** represent the most common causal factors in the generic control loop – problems in Control Algorithm, Process Model, Inadequate Operation of Actuator or Sensor, and so on.

WebSTAMP

Insulin Pump

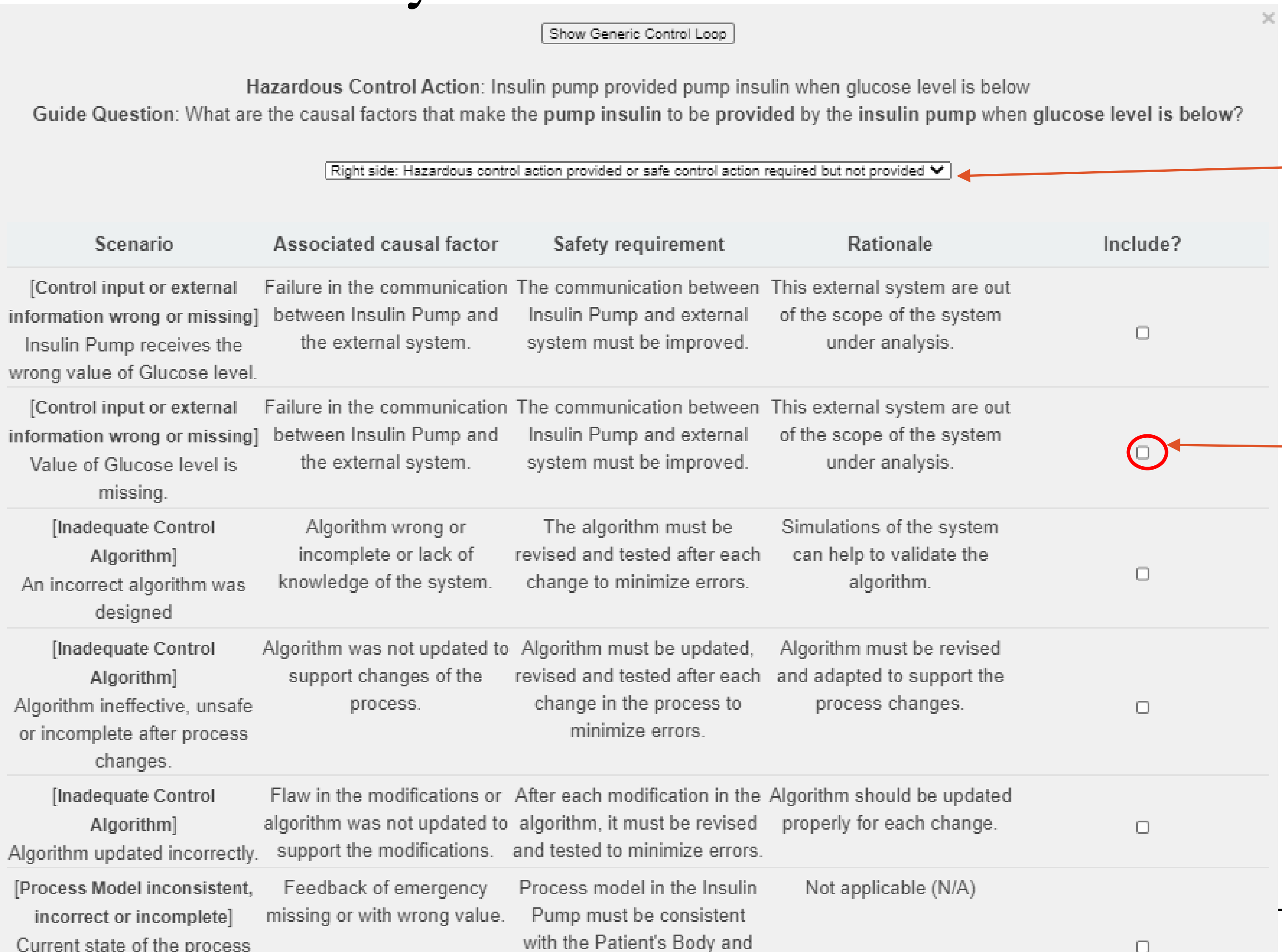
Purpose of the Analysis

Control Structure

Unsafe Control Actions

→ Loss Scenarios

Identify loss scenarios



Hazardous Control Action: Insulin pump provided pump insulin when glucose level is below

Guide Question: What are the causal factors that make the **pump insulin** to be **provided** by the **insulin pump** when **glucose level is below**?

Right side: Hazardous control action provided or safe control action required but not provided

Scenario	Associated causal factor	Safety requirement	Rationale	Include?
[Control input or external information wrong or missing] Insulin Pump receives the wrong value of Glucose level.	Failure in the communication between Insulin Pump and the external system.	The communication between Insulin Pump and external system must be improved.	This external system are out of the scope of the system under analysis.	□
[Control input or external information wrong or missing] Value of Glucose level is missing.	Failure in the communication between Insulin Pump and the external system.	The communication between Insulin Pump and external system must be improved.	This external system are out of the scope of the system under analysis.	□
[Inadequate Control Algorithm] An incorrect algorithm was designed	Algorithm wrong or incomplete or lack of knowledge of the system.	The algorithm must be revised and tested after each change to minimize errors.	Simulations of the system can help to validate the algorithm.	□
[Inadequate Control Algorithm] Algorithm ineffective, unsafe or incomplete after process changes.	Algorithm was not updated to support changes of the process.	Algorithm must be updated, revised and tested after each change in the process to minimize errors.	Algorithm must be revised and adapted to support the process changes.	□
[Inadequate Control Algorithm] Algorithm updated incorrectly.	Flaw in the modifications or algorithm was not updated to support the modifications.	After each modification in the algorithm, it must be revised and tested to minimize errors.	Algorithm should be updated properly for each change.	□
[Process Model inconsistent, incorrect or incomplete] Current state of the process	Feedback of emergency missing or with wrong value.	Process model in the Insulin Pump must be consistent with the Patient's Body and	Not applicable (N/A)	□

Example of the checklist for "Right Side". You can see more tuples scrolling down your page or changing the checklist for Left side by clicking here.

To select, just click here. After selecting, you have to add the tuples using the button in the bottom of the page)

Identify loss scenarios

- Another option is to click on the button “Add new 4-tuple”. In this case, the analyst must define (edit) the set of Scenario, Associated Causal Factors, Recommendations and Rationale and assign a Guideword for that tuple.
- This option is the way to include new scenarios and causal factors in this step.

Identify loss scenarios

WebSTAMP

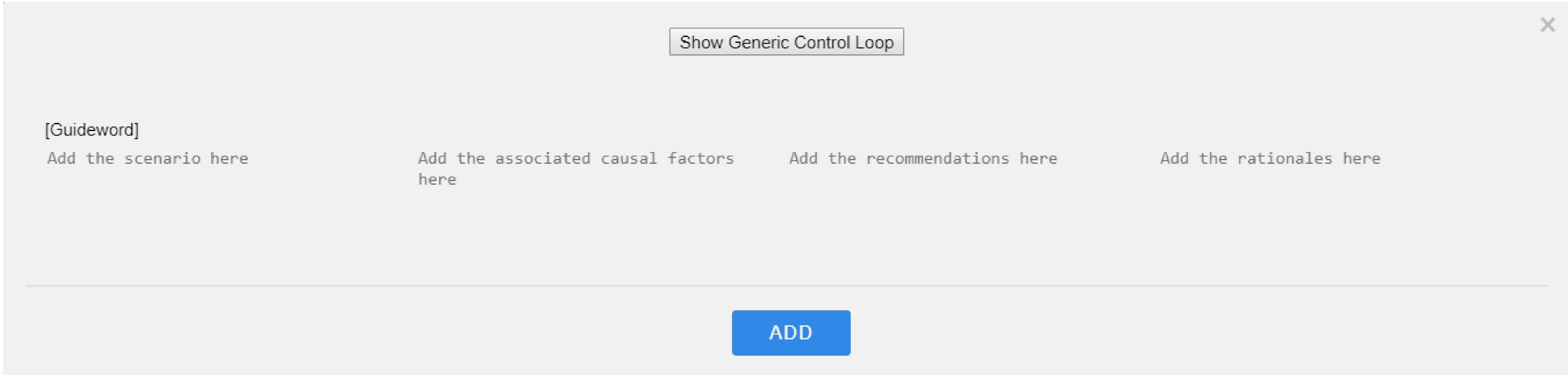
Insulin Pump

Purpose of the Analysis

Control Structure

Unsafe Control Actions

→ Loss Scenarios



[Guideword]

Add the scenario here

Add the associated causal factors here

Add the recommendations here

Add the rationales here

Exercise

- If you finished the analysis for "Pump Insulin" control action, go to the step Model Control Structure and add a new control action for the "Insulin Pump" controller: "Stop pumping insulin".
- And perform the analysis with the added control action.

References

1. Radcliffe, J. (2011). Haking Medical Devices for Fun and Insulin: Breaking the Human SCADA System. Blackhat USA, 2011. Available at https://media.blackhat.com/bh-us-11/Radcliffe/BH_US_11_Radcliffe_Hacking_Medical_Devices_WP.pdf

2. Young, W., Porada, R. (2017). System-Theoretic Process Analysis for Security (STPA-SEC): Cyber Security and STPA. In: 2017 STAMP Workshop. Available at http://psas.scripts.mit.edu/home/2017-stamp-presentations/.

3. Thomas, J. (2013). Extending and automating a systems-theoretic hazard analysis for requirements generation and analysis (Doctoral dissertation, Massachusetts Institute of Technology).

4. Gurgel, D. L., Hirata, C. M., Bezerra, J. D. M. (2015). A rule-based approach for safety analysis using STAMP/STPA. In 2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC) (pp. 7B2-1). IEEE.

Concluding Remarks

- WebSTAMP is an ongoing development project. It has bugs yet.
- Many features are still being implemented. They include GUI for editing control structure, import/export of analysis, report generation (certification), traceability, collaborative analysis, consistency check, and STPA extensions, such as the inclusion of threat model for security and ontology.
- Interests from aeronautical and railroad industries.

WEBTAMP: A WEB TOOL TO PERFORM STPA ANALYSIS

Celso Massaki Hirata, ITA

hirata@ita.br