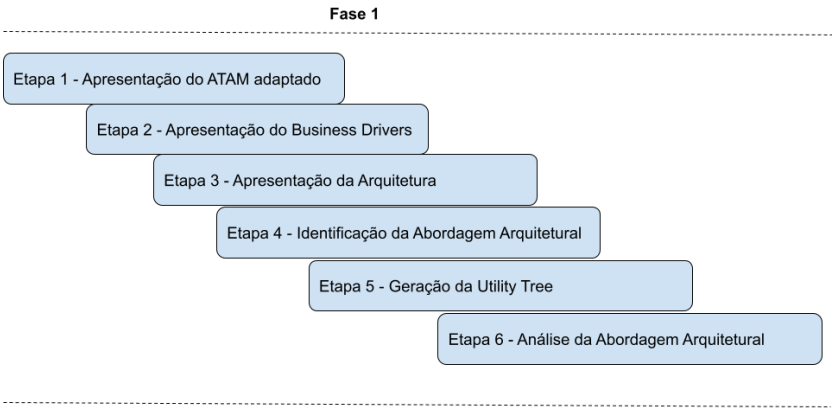
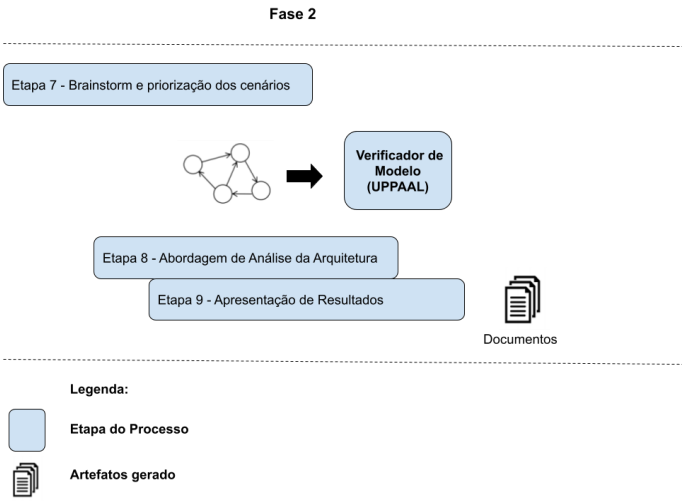
Universidade Federal da Bahia (UFBA) - BA – Brasil
Orientador: Ivan Machado
Discente: Caiza Almeida Fortunato

Preliminary
1.1 ATAM-4SAS

This document aims to present the preliminary concepts that involve the realization of this focus group. It contains information on the method of analysis to be addressed during this study. Likewise, the MAPE-K model and the UPPAAL Statistic Model Checking will be presented. Then, an application of the ATAM-4SAS model is described, containing 9 phases distributed in two stages.

ATAM-4SAS is an adaptation of ATAM (ATAM - Architecture Tradeoff Analysis Method) which is a modified architecture Tradeoff analysis method to be used in self-adapted systems. A self-adaptive system is a closed-loop system capable of monitoring itself, its context, detecting significant changes, deciding how to react and taking action to execute those decisions, with as little human interference as possible. The method consists of 2 phases and 9 steps in total. Steps 1 to 6 make up phase 1 and steps 7 to 9 are in phase 2. Each step includes the preparation of documents necessary for the Tradeoff analysis of an ATAM-based architecture.
The first phase is architecture-centric and focuses on extracting information and analyzing it; the second phase focuses on getting stakeholder views and verifying the results of the first phase.
To help ATAM deal with trade-offs between quality attributes of self-adaptive systems, our proposal includes carrying out the MAPE-K modeling, in step 7, using the UPPAAL SMC tool, as illustrated in the figure. below.





1.2 MAPE-K model

In the MAPE-K loop, Monitor collects and filters a large amount of data from managed resources by a touchpoint sensor interface. It correlates them into a symptom that could be examined. Analyze observes and examines the situation and decides whether some change needs to be made. Whether a change is required, Analyze requests a change to Plan. This change request describes the desirable modification. In its turn, Plan creates or selects a procedure to enable a desired adjustment in the managed resource. Then, it generates a change plan to Execute, which performs the procedure that was generated via Plan through a set of actions. These actions are carried out using the touchpoint effector interface of managed resources. Finally, Knowledge is an implementation of a registry, dictionary, database, or another repository that provides access to knowledge according to the interfaces prescribed by the architecture.

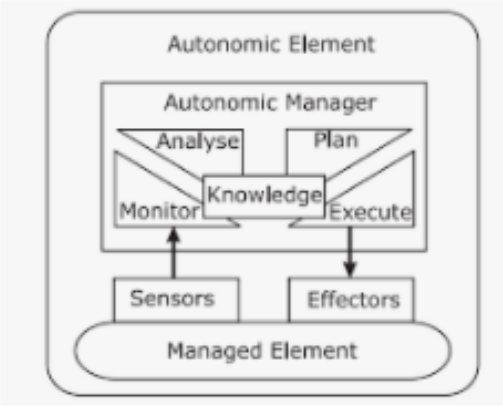


Fig 1: Monitor-Analyse-Plan-Execute Knowledge (MAPE-K)

1.3 UPPAAL Statistical Model Checking (SMC)

The UPPAAL model-checker is based on the theory of timed automata (TA) and its modeling language offers additional features such as bounded integer variables.

Besides, UPPAAL is an integrated tool environment for modeling, validation, and verification of real-time systems modeled as networks of timed automata extended with data types". Within UPPAAL, a system is modeled as a network of several timed automata in parallel .

2. Applying the ATAM-4SAS

Next, we will describe the application of each stage of the adapted model through its application being applied in a smart home project, the SHE. The objective of this project is to develop a self-adaptive system that works by collecting data from sensors such as: temperature, water, light, presence, distance, soil moisture, slope and gas. The SHE must be able to. It has the ability to automatically recognize new contexts and make the necessary adaptations at runtime.

Phase 1

Step 1 - Present the ATAM: A 30-minute presentation was held to present an overview of the adapted ATAM to apply to the SHE platform. The list of expected outputs was also introduced.

Step 2 - Present business drivers: The architect carried out a presentation about the SHE and the business needs driven by QA goals at a high level of abstraction, e.g., driving architectural requirements, and the set of functional requirements the system should meet. As an output, we created a document containing all the information related to the business context collected in this step. Table \ref{fig:business} illustrates this output.

Output Artifact - Step 2: Business Context / Motivators Presentation document

Business Context/Drivers Presentation
1. Description of the business environment.
SHE was designed aimed to reconfigure the system at runtime, according to the environment where the software is running. To achieve this goal, the system needs to be aware of the context to which it is embedded. This context awareness is obtained from the sensors used by the user and the process of installing and uninstalling the features that compose it. The system adapts according to the set of sensors that were connected by the user. For this adaptation, the system constructs a screen of presentation based on sensors that are being used in a certain moment.
2. Description of the technical constraints.
The Context Sensors were developed as a SPL developed in C++, where each sensor was developed using the same core. Moreover, the core was developed in Java.
3. Quality attributes desired and what business needs these are derived from.
The set of desired quality attributes and business needs. When either adding or removing the sensors, we want that SHE follows the QA: performance, availability, cost, scalability, interoperability, robustness, portability, maintainability, and reliability.

Business Context/Drivers Presentation based on the ATAM template.

Step 3 - Present the architecture.
In this step, we could elaborate on an architectural presentation, as Table shows. Figure shows an activity diagram representing the activities of the proposed approach, emphasizing the message exchanges among each architectural element. The diagram comprises a set of activities, numbered in the order they are executed. It illustrates a scenario in which a new physical sensor is connected to the system, as explained next:

Output Artifact - Step 3: Architecture Presentation document

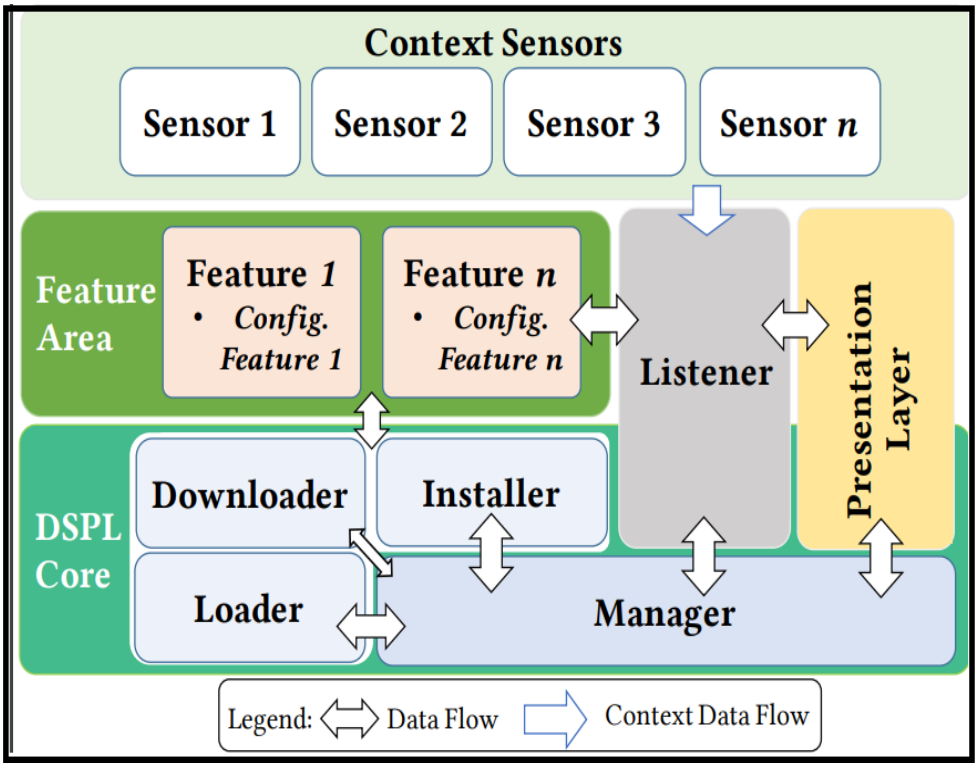
Architecture Presentation
1. Driving architectural requirements.
Driving architectural requirements (performance, availability, cost, scalability, and interoperability) uses the MAPE-K model to deal with the problem of automatic reconfiguration of the software system considering its context at runtime.
2. Description of the business constraints.
The system is able to detect 8 sensors (water/humidity, fire, gas, presence luminosity, inclination, temperature, and soil moisture). When added to the SHE system, it detects if the feature has already been installed, otherwise, the system performs the feature download, installs and displays the data on the system screen. If the system has the feature installed, it only executes the feature. The moment any sensor is removed or some failure occurs, the system detects and removes the corresponding data from the feature out of the screen.
3. Process/thread.
The core of the system is composed of a set of threads. Listener, Manager, and presentation layer components are threads. Data synchronization between data collected from sensors developed using Arduino and the core installed on a Raspberry Pi.
4. Hardware.
The prototype hardware uses low-cost devices such as Arduino and Raspberry Pi. The former encompasses the Context Sensors and the latter executes both the core and the Feature Area.
5. Architectural approaches or styles employed.
SHE uses the MAPE-K model to support the required information throughout the activities. The proposed architecture model employs the styles of the component-and-connector view-type called Publish-Subscribe to centralize the communication process between the internal and external environments of the system. Such an architectural style enables the creation of objects that react to events generated by their environment, and in turn, it may impact other components as a side-effect of their event announcements. The Publish-Subscribe architecture style capability to completely dissociate participants from communication, allowing the development of more tolerant asynchronous applications. It deals with reliability because if the message producer fails the system does not stop. SHE uses object-oriented design patterns such as polymorphism and inheritance that are recurring design solutions for object-oriented systems which can improve reusability and maintainability.

Architecture Presentation based on the ATAM template.

Step 4 - Identify architecture. In this step, the focus is on the analysis of architecture by understanding the approaches that are used in it. Then, the architect identified the set of details of the SHE architecture. As a result, we could provide the DSPL architecture model of the SHE.

The SHE Architecture Model is based on the MAPE-K, and it was designed to support DSPL engineering. The proposed architecture handles changes in context and enables dynamic adaptations of the system behavior at runtime. The SHE Architecture Model comprises a set of features: (i) Context Sensors, (ii) DSPL Core, and (iii) Feature Area. Figure shows the architecture overview, which is detailed next.

Output Artifact - Step 4: Smart home project architecture model



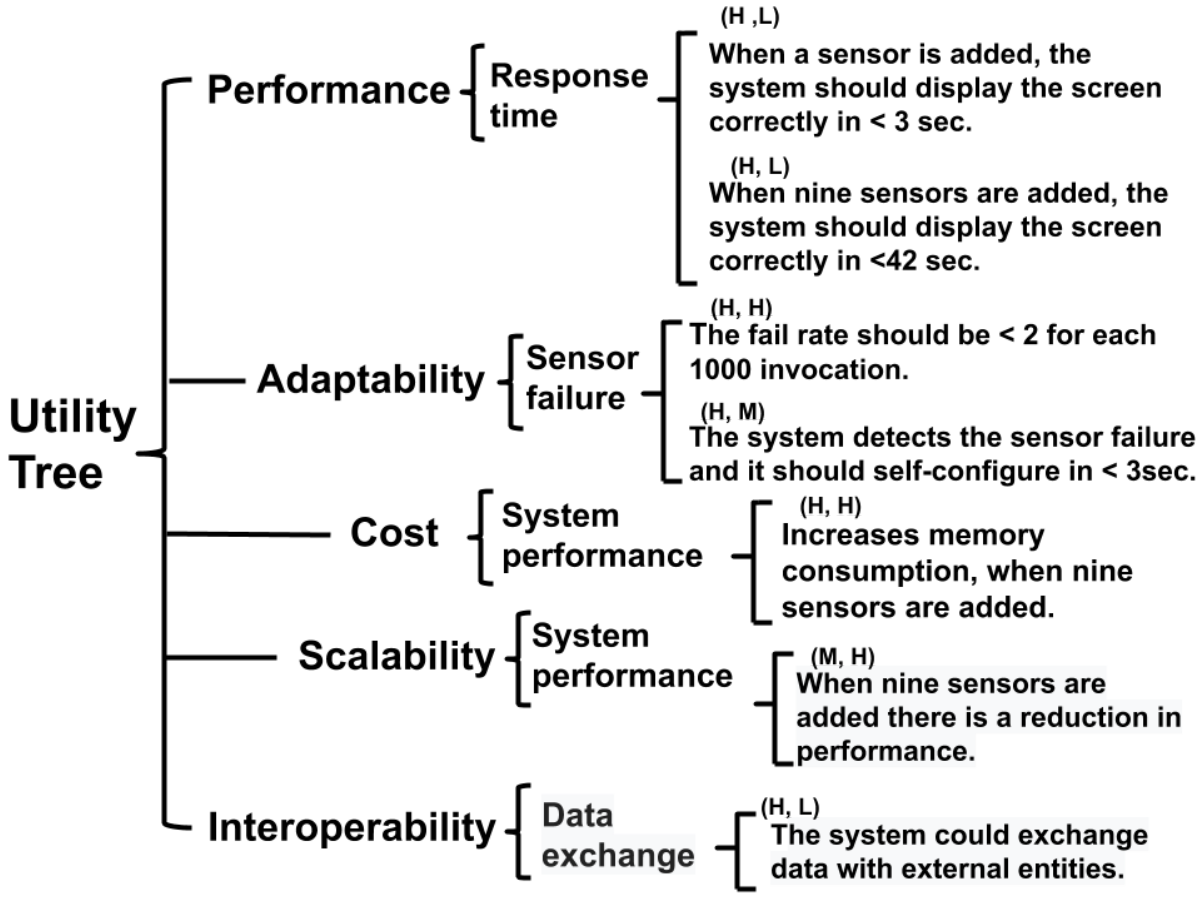
Step 5 - Generate and prioritize the QA scenarios:} Alike in a brainstorming session, we elicited and documented the scenarios for each function of the SHE, and associated them with a proper QA. After that, we voted the priority of each scenario. The index establishes a priority value ranging from \textit{Low} to High.

Based on our stated concerns and on our elicitation task, we could generate the Utility Tree. Through the Utility Tree construction process, we observed that five QA were the major architectural drivers for the overall system quality: performance, adaptability, cost, scalability and interoperability. As part of our elicitation process, we ensured that each of the Utility Tree scenarios had a specific stimulus and response associated with it.

Figure utility shows a sample Utility Tree. In the Figure, performance, adaptability, cost, scalability, and interoperability are the high-level nodes. Under each of these QA there are specific sub-factors. For example, performance is refined in response time. In its turn, response time is broken down into: ``(i) When a sensor is added, the system should display the screen correctly in $<$ 3 sec"; and (ii) ``When nine sensors are added, the system should display the screen correctly in $<$ 42 sec". These are the key stimuli. Notice that these sub-factors are related to the QA characterizations.

The prioritization of the Utility Tree is carried out along two dimensions: by the importance of each node to the success of the system and to the degree of the perceived risk posed by the achievement of this node (i.e., how easy the architecture

team feels this level of performance, adaptability, or other attributes will be achieved). Both importance and risk could be High, Medium or Low. For example, ``When a sensor is added, the system should display the screen correctly in $<$ 3 sec" has priorities of (H,L), which means it is of high (H) importance to the success of the system and pursues a low (L) risk to achieve. Meanwhile, ``When nine sensors are added there is a reduction in performance" has as priorities the values (M,H), which means that the success of the system is of \textit{medium (M) importance} and the achievement of this scenario is perceived to be as a high (H) risk.



Output Artifact - Step 5: Utility Tree

Step 6 - Analyze architectural approaches: This step aims to analyze the location of the risks, non-risks, sensitivity points, and tradeoff points of each scenario.

Scenario	When the sensor is added, the system performs self-configuration and the sensor should display the screen correctly when installing the feature.
Attributes	Performance and Adaptability.
Stimulus	Adding a sensor.
Response	When adding the sensor, the system must detect an unknown sensor, it is carried out by the identification of which sensor the system verifies if this sensor is installed. If yes, the sensor starts, if not, the sensor is downloaded, installed, and started. This response should have a duration of less than 3 seconds.
Architectural decisions	1)Install/activate plugins using the MAPE-K model as support, because its loop support identification of the context change and promote proper system configuration and reconfiguration; 2) It uses the Publish/Subscribe architecture style for interoperability; 3) It installs the feature and it builds the data view; The system must comply with the installation of the feature correctly.
Risk	1) When adding the sensor, the system does not detect. Consequently, the feature is not installed; 2) Installing a feature that is not available on maven, consequently causing an exception; 3) The non-identification of context change.
Sensitivity Point	1) Hardware Processing Time; 2) If the transmitted data does not have the correct formatting, it impacts the adaptation sequence.
Tradeoff	The greater the need for adaptation, the lower the system performance.
Reasoning	The MAPE-K model was chosen as the reference model for self-adaptive systems, as one of the possible mechanisms to identify the change of context and to promote the adaptations of the system.

Sample SHE Usage Scenario 1.

Scenario	The system is working and the sensor fails.
Attributes	Availability, Robustness and Maintainability.
Stimulus	The system detects the failure and uninstalls the faulty feature.
Response	When adding the sensor, the system must detect an unknown sensor, it is carried out by the identification of which sensor the system verifies if this sensor is installed. If yes, the sensor starts, if not, the sensor is downloaded, installed, and started. This response should have a duration of < 3 seconds. And, the fail rate should be less than 2 for each 1000 invocation.
Architectural decisions	1)Install/activate and uninstall/deactivate plugins using the MAPE-K model as support, because its loop support identification the context change and promote proper system configuration and reconfiguration; 2) It uses the Publish/Subscribe architecture style for interoperability; 3) We used OSGI framework;
Risk	The system does not recognize the sensor failure.
Sensitivity Point	1) If the transmitted data does not have the correct formatting it impacts on the sequence of the adaptation.
Tradeoff	*n/a*
Reasoning	The MAPE-K model was chosen as the reference model for self-adaptive systems, as one of the possible mechanisms to identify the change of context and to promote the adaptations of the system. Also, we chose an OSGI framework due to it gives support to failure exception at runtime.

Table 4: Sample SHE Usage Scenario 2.

Scenario	The system is running without sensors. Then, nine sensors are added in the system.
Attributes	Cost, Availability, Performance, interoperability, scalability.
Stimulus	Nine sensors are to add to system at the same time.
Response	Each feature has to respond in less 3 seconds. Then, the system has 45 sec for installation and configuration of the 9 sensors. Evaluate storage and processing capacity to be defined.
Architectural decisions	1)Install/activate and uninstall/deactivate plugins using the MAPE-K as support, because its loop support identification the context change and promote proper system configuration and reconfiguration; 2) It uses the Publish/Subscribe architecture style (via MQTT Broker); 3) It installs the feature and it builds the data view; The system must comply with the installation of the feature correctly.
Risk	The greater the number of sensors, the higher the impact of Raspberry PI's processability. Then, installing the feature reduces storage capacity.
Sensitivity Point	1) The higher the number of sensors, the greater the risk of degradation of the process, reducing performance.
Tradeoff	Increased performance, hence, increases cost.
Reasoning	The MAPE-K model was chosen as the reference model for self-adaptive systems, as one of the possible mechanisms to identify the change of context and to promote the adaptations of the system. Also, we used MQTT Broker to support to interoperability.

Table 5: Sample SHE Usage Scenario 3.

Scenario	The system is running with nine sensors. And the sensors are removed in the system.
Attributes	Performance and interoperability.
Stimulus	Uninstalling nine sensors.
Response	The system returns to the initial version and has to have the core of the SHE in less than 24 seconds.
Architectural decisions	1) Uninstall/deactivate plugins using the MAPE-K as support, because its loop support identification the context change and promote proper system configuration and reconfiguration; 2) It uses the Publish/Subscribe architecture style for interoperability; 3) It uninstalls the feature and it removes the data view; The system must comply with the uninstall of the feature correctly.
Risk	The system does not detect the removal of the sensor.
Sensitivity Point	1) Hardware Processing Time. 2) If the transmitted data does not have the correct formatting it impacts on the sequence of the adaptation.
Tradeoff	The greater the need for adaptation, the lower the system performance.
Reasoning	The MAPE-K model was chosen as the reference model for self-adaptive systems, as one of the possible mechanisms to identify the change of context and to promote the adaptations of the system.

Table 6: Sample SHE Usage Scenario 4.

Output Artifact - Step 6: Scenarios

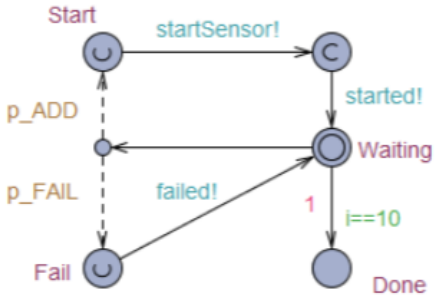
Phase 2

Step 7 - Brainstorm and Prioritize Scenarios:
After collecting the scenarios, we prioritize them and rank all of the QA. Next, we modeled the MAPE, by following the scenarios 2 and 3 in the UPPAAL tool. We next delve into details about the design of the MAPE behaviors within the SHE platform.

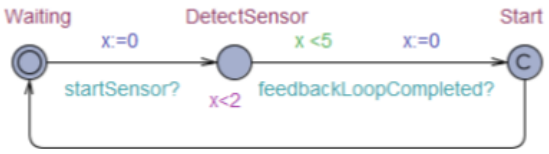
We used a set of formal models for self-adaptive components to use in the UPPAAL SMC tool. We differentiate between the Monitor, Analyze, Plan and Execute components as the main components of a MAPE control loop.

No	Quality Attributes	Scenario
1	Performance and Adaptability	When the sensor is added, the system performs self-configuration and the sensor should display the screen correctly when installing the feature.
2	Cost, Availability, Performance, Interoperability, Scalability.	The system is running without sensors. Then, nine sensors are added in the system.
3	Robustness and Adaptability	The system is working and the sensor fails.
4	Performance	The system is running with nine sensors. And the sensors are removed.

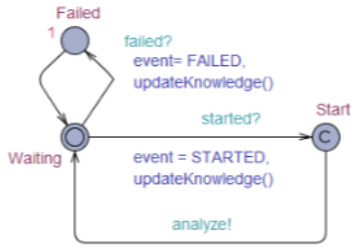
High priority scenarios.



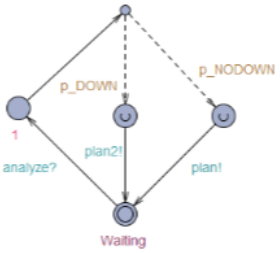
Sensor behavior.



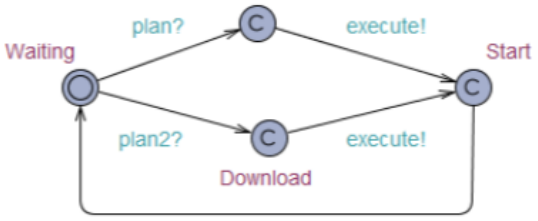
SHE platform behavior.



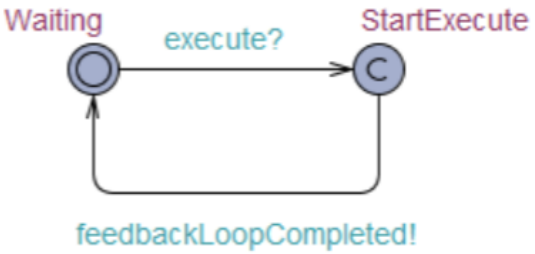
(a) Monitor



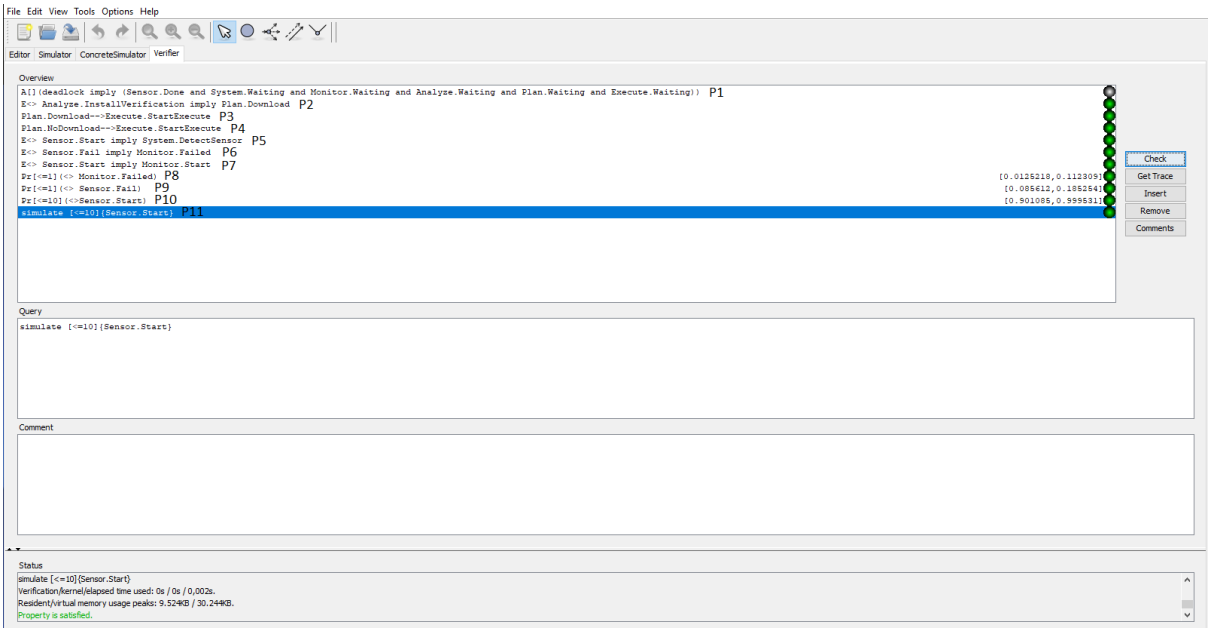
(b) Analyze

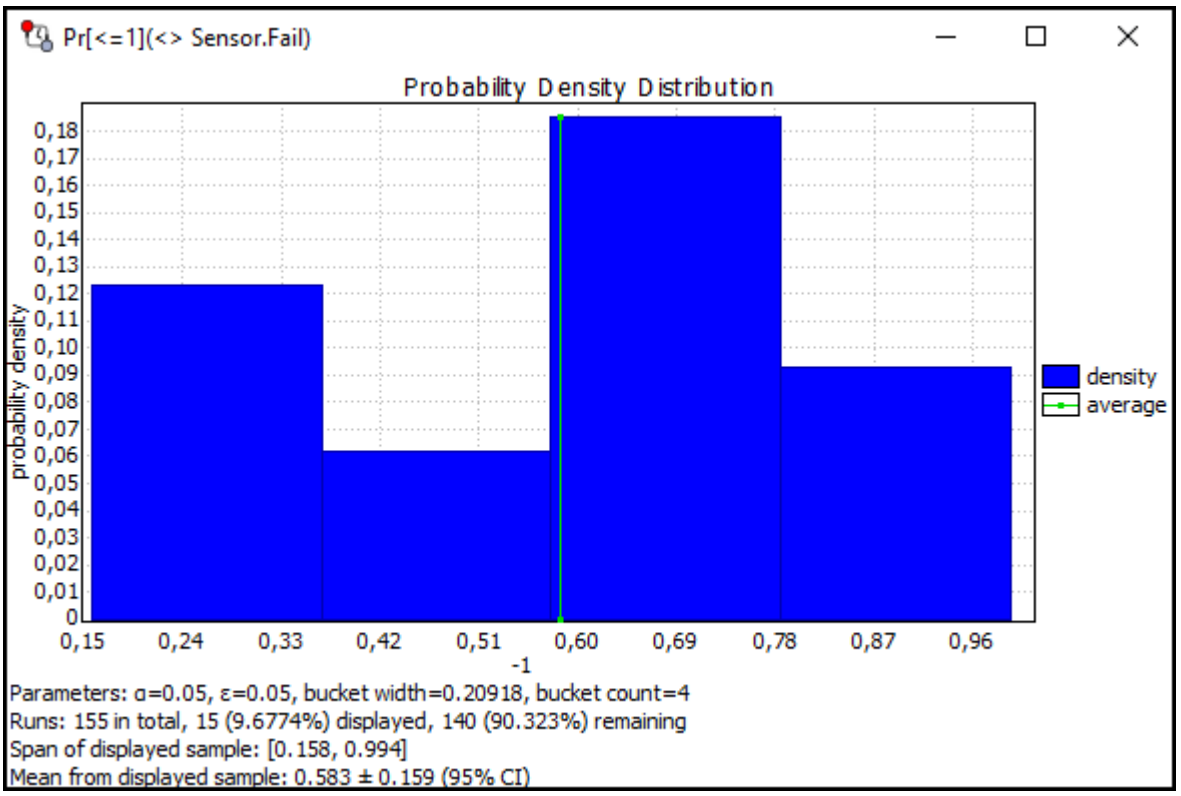


(c) Plan



(d) Execute





Through the scenarios and the QA leveraged by ATAM, we were able to perform the MAPE modeling. We could verify the robustness and performance when nine sensors are added, as it was described in Scenario 2 and Scenario 3, by the UPPAAL SMC. As result, the simulation presented a failure rate of less than 2 percent per 1000 simulations.

Furthermore, we used the Weyns (2016) study to analyze the average cost, failure rate and response time by UPPAAL SMC. The result of a series of 10000 invocations of the feature with simulation queries of RSEM 5% and 10%:(a) average failure rate in the RSEM 5% and (b) average failure rate in the RSEM 10% most simulations had values less than 0.1 units (this is less than avgFRate 2 (x 10(-3)); (c) average cost in the RSEM 5% shows most simulations between 4.0 to 4.5 and (d) average cost in the RSEM 10% shows most simulations between 5.5 to 6.0 units (this is less than avgCos 8 (x 10-3)); (e) average response time in the RSEM 5% most simulations between 5.5 to 6.5 units of time and (f) average response time in the RSEM 10% most simulations between 3.5 to 4.0 units of time (this is bigger than avgResponseTime 3s (x 10-3)).

Output Artifact - Step 7: Scenario prioritization table, formal MAPE-K model, verification of quality attributes (robustness, performance and cost)

Step 8 - Analyze Architectural Approaches: In this phase, we analyzed whether we could find new information that has not been documented yet (e.g. a given scenario or QA which has not been previously elicited and documented). However, in our

study, we did not find any new relevant information we forgot in the previous steps. Then, we moved on to the next step;

Step 9 - Present the results: In the last step, we performed a meeting where we presented the stakeholders all the gathered information (from the previous steps), the results of the analysis.

Output Artifact - Step 9: The documented architectural approaches/styles, the set of scenarios and their priorities, the set of attribute-based questions, the utility tree, the discovered risk, the documented non-risks, the sensitivity points and points of compensation found