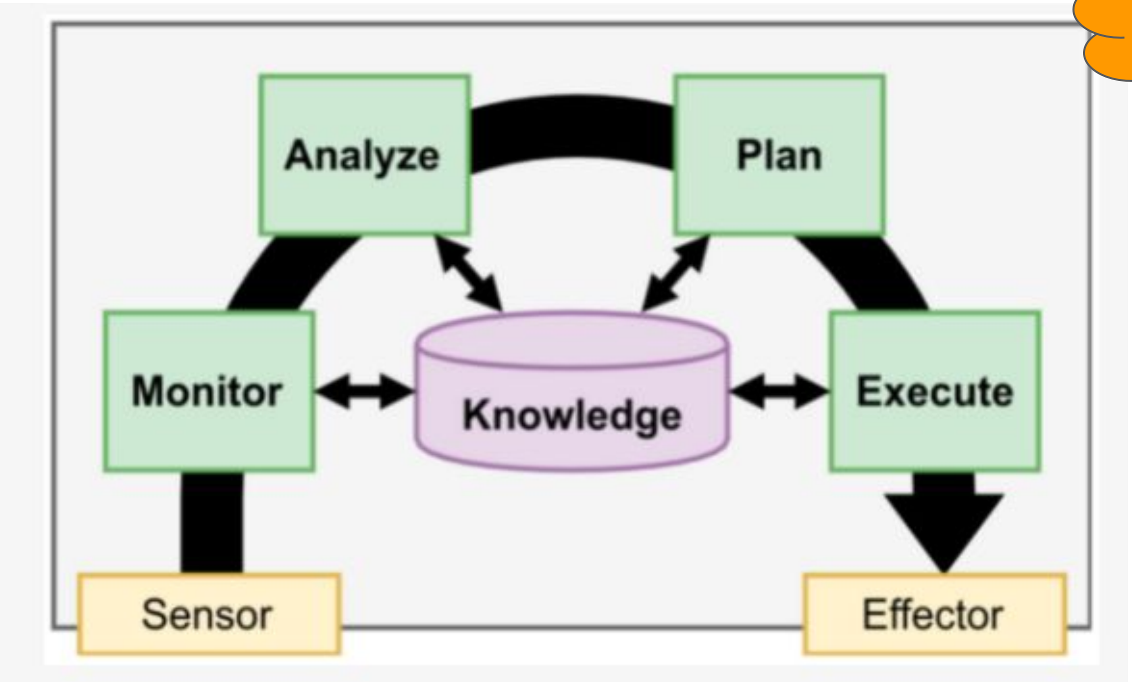
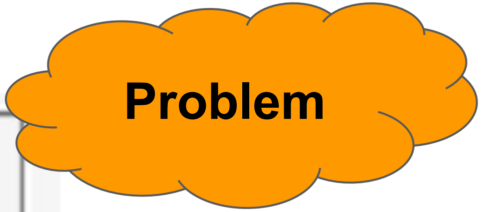
ATAM-4SAS

Mestranda: Caiza Fortunato
Professor: Ivan Machado

Self-adaptive system



MAPE-K model



What is the Purpose of the ATAM?

1. The ATAM can be done early in the software development life cycle.

2. The ATAM is meant to be a risk identification method, a means of detecting areas of potential risk
within the architecture of a complex software intensive system.

3. The ATAM will produce analyses commensurate with the level of detail of the architectural specification.

The purpose of the ATAM is to assess the consequences of architectural decisions in light of quality attribute requirements.

ATAM


Step 1 - Present the ATAM
Artifacts Generated
Step 2 - Present Business Drivers
Business context presentation
Step 3 - Present Architecture
Architecture presentation
Management presentation
Phase 1
Step 4 - Identify Architectural Approaches
Potential design decisions
Step 5 - Generate Quality Attribute Utility Tree
Utility Tree
Step 6 - Analyze Architectural Approaches
Documentation of the most important decision
Step 7 - Brainstorm and Prioritize Scenarios
Prioritized decisions
if fail
Step 8 - Analyze Architectural Approaches
Potential risks and issues
Revised decisions
Phase 2
Step 9 - Present Results
Review report

ATAM-4SAS

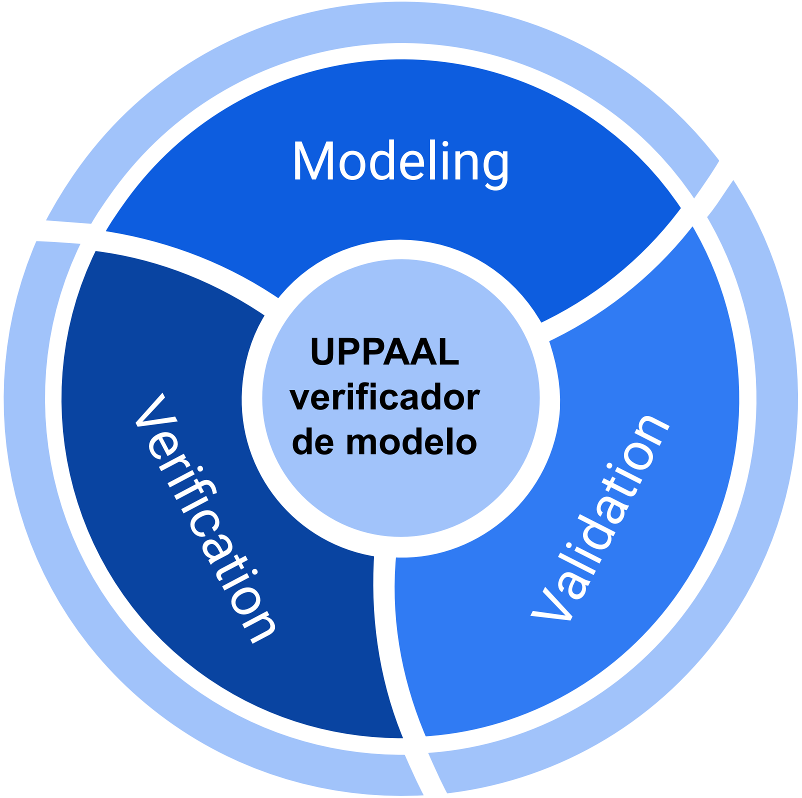
Step 7 Brainstorm and Prioritize Scenarios

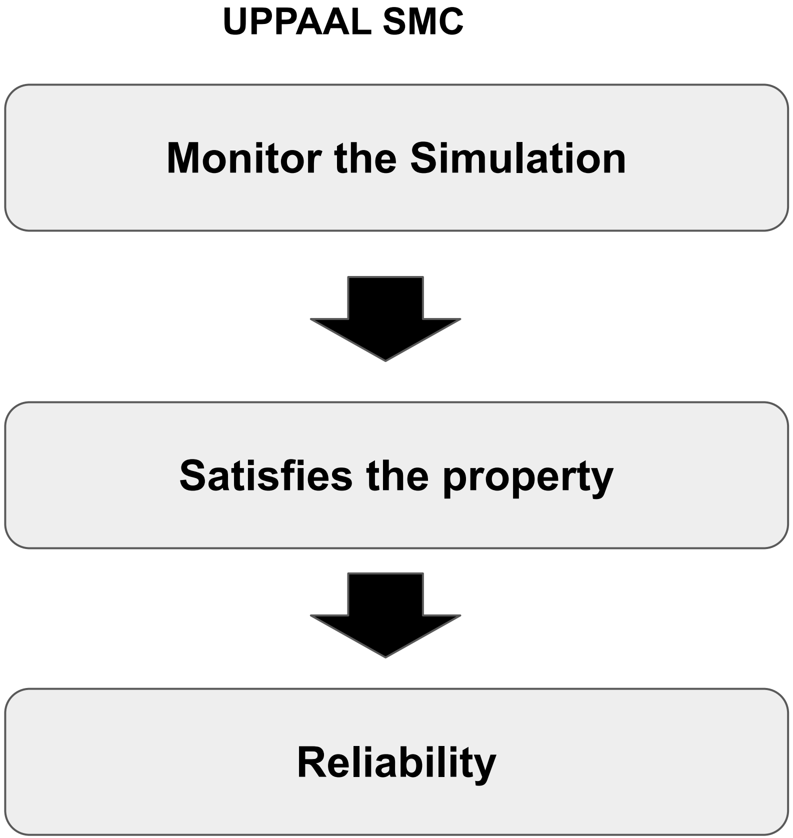
UPPAAL model-checker

Formal methods for MAPE-K

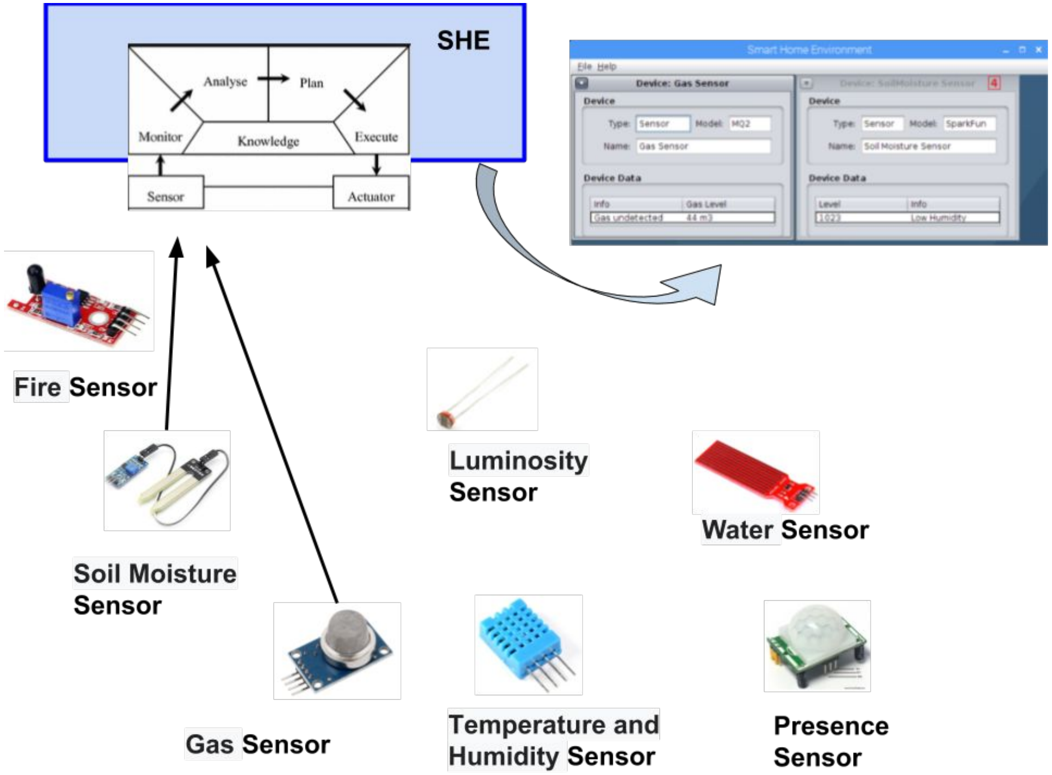
Simulation

UPPAAL tool

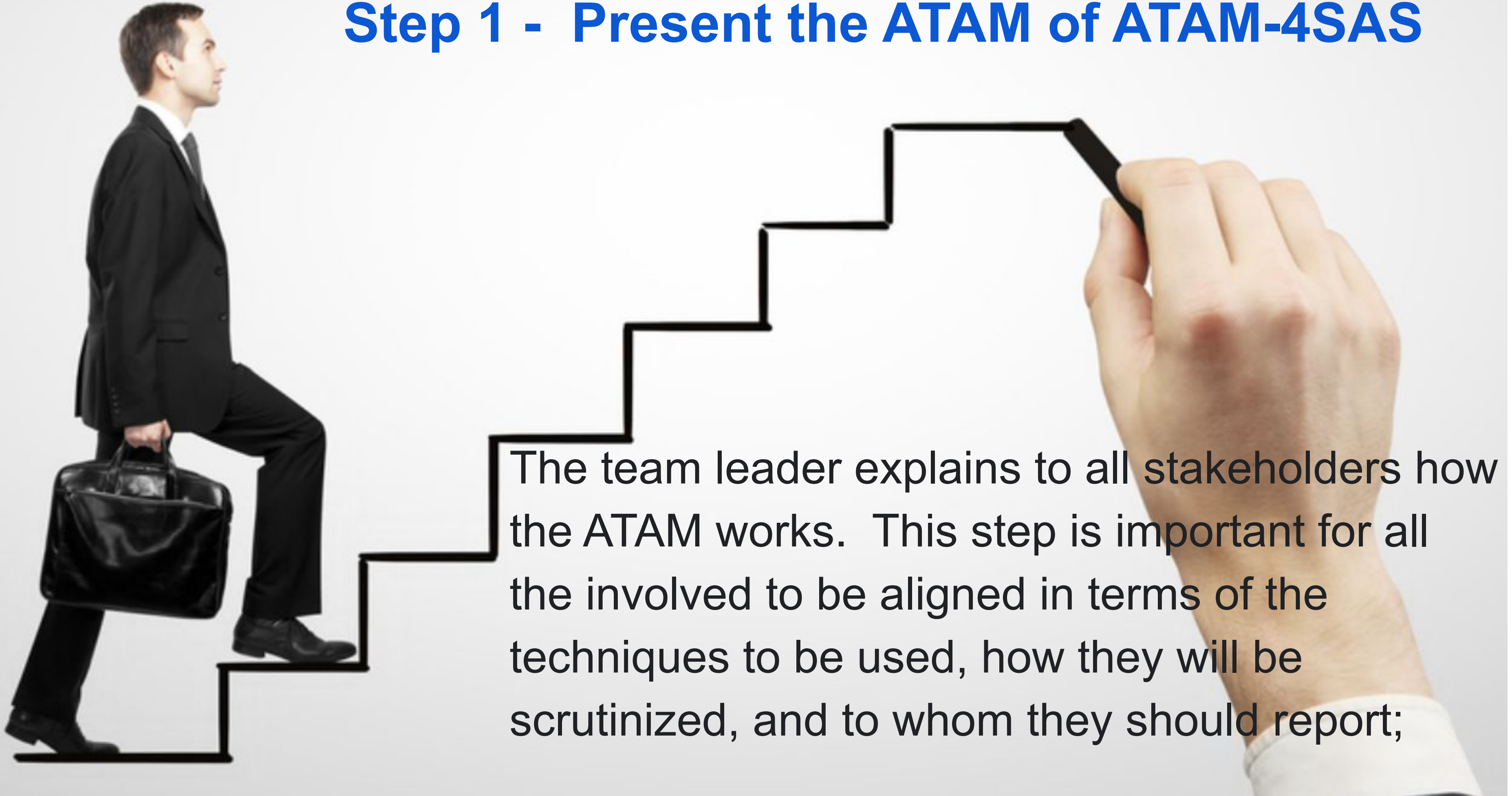

Modeling
UPPAAL
verificador
de modelo
Verification
Validation


UPPAAL SMC
Monitor the Simulation
Satisfies the property
Reliability

Applying ATAM-4SAS in the Smart Home Environment (SHE) project



Step 1 - Present the ATAM of ATAM-4SAS



The team leader explains to all stakeholders how the ATAM works. This step is important for all the involved to be aligned in terms of the techniques to be used, how they will be scrutinized, and to whom they should report;

Step 2 - Present Business Drivers.

The project manager shows an overview of the business goals and context. It is demonstrated in a high level of abstraction: its major stakeholders, major QA goals which form the architecture, the most important Functional Requirements (FR), and technical, managerial, economic or political constraints;

Step 2 - Present Business Drivers.

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 4 - Identify Architectural Approaches.

Apresentação das abordagens arquitetônicas usadas para atender aos requisitos dos atributos de qualidade e restrições técnicas.

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.

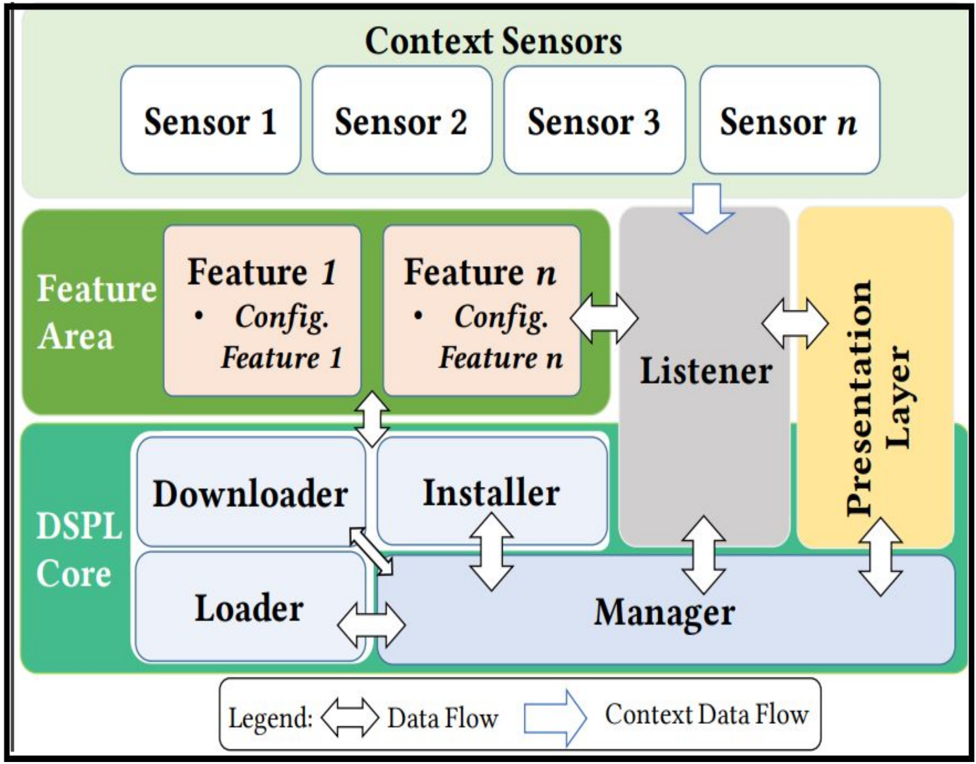
5. Architectural approaches or styles employed.

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.

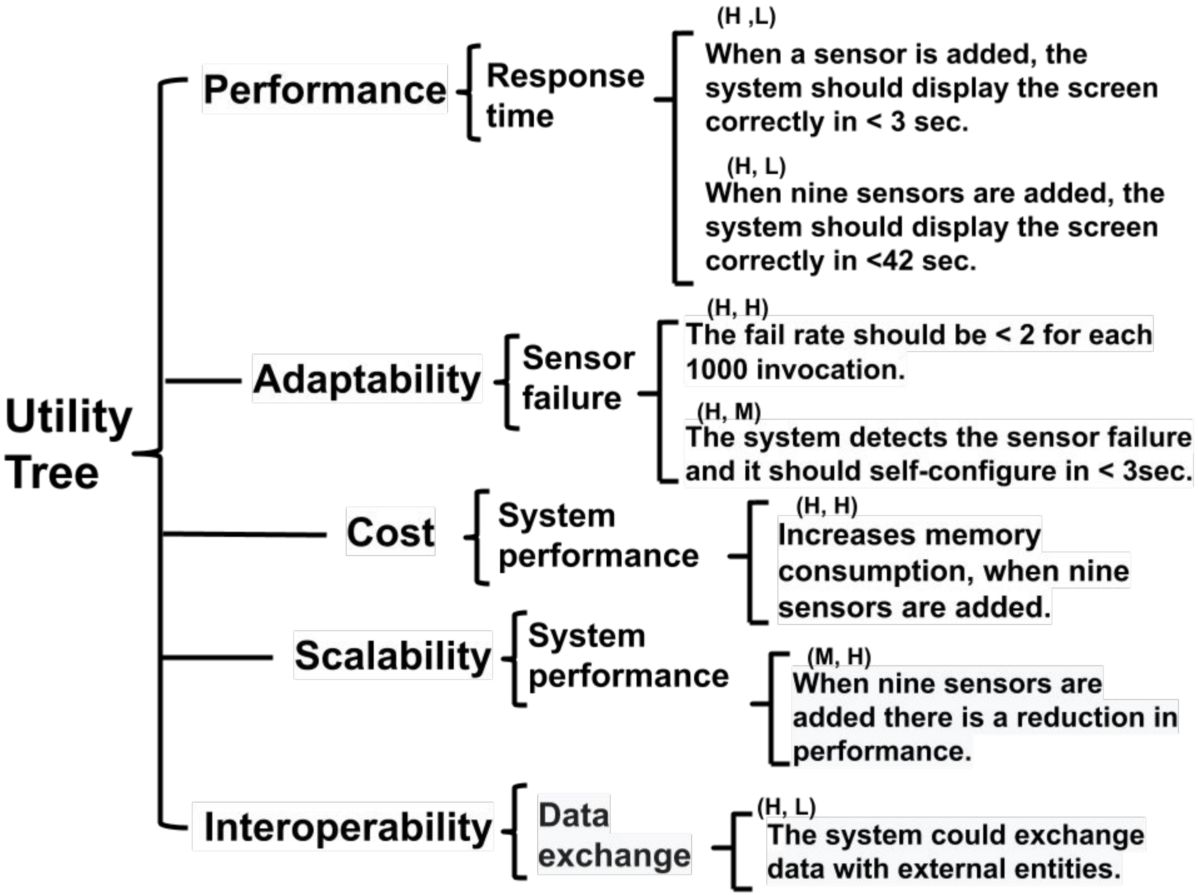
Architecture Presentation based on the ATAM template.

Etapa 4: Identificação da abordagem arquitetural

In this step, the focus is on the analysis of architecture by understanding the approaches that are used in it.



Step 5 - Generate QA Utility Tree.



Step 6 - Analyze Architectural Approaches.

After the architecture team elicits the utility tree, the evaluation team might analyze architectural decisions and identify their potential risks, sensitivity points, and tradeoff points through the documentation.

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.

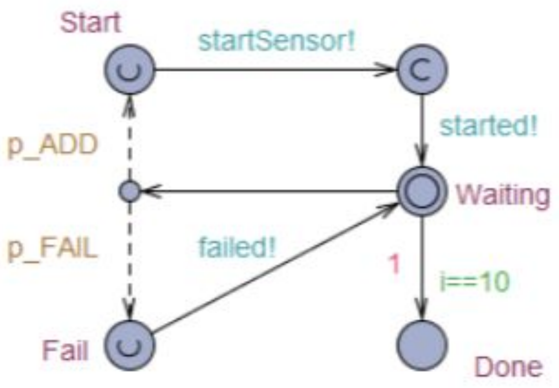
Sample SHE Usage Scenario 1.

Step 7 - Brainstorm and Prioritize Scenarios.

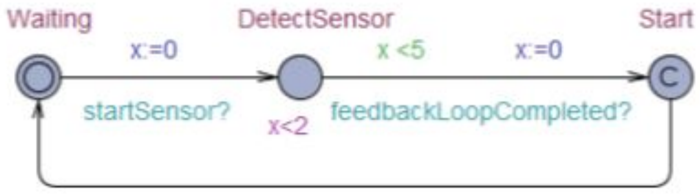
No	Quality Attributes	Scenario
1	Performance and Adaptability	When the sensor is added, the system performs self-configuration and the sensor should display the screen correctly when installing the feature.
2	Cost, Availability, Performance, Interoperability, Scalability.	The system is running without 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.

MAPE-K behaviors on the SHE platform

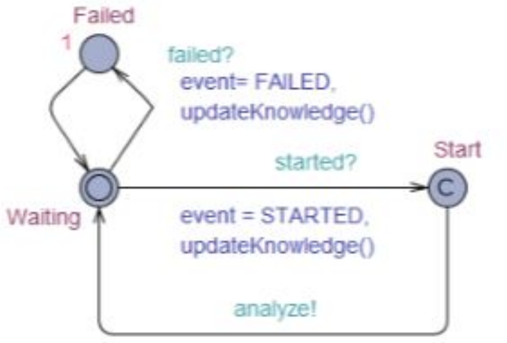


Sensor behavior.

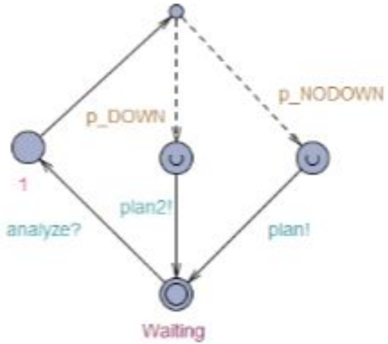


SHE platform behavior.

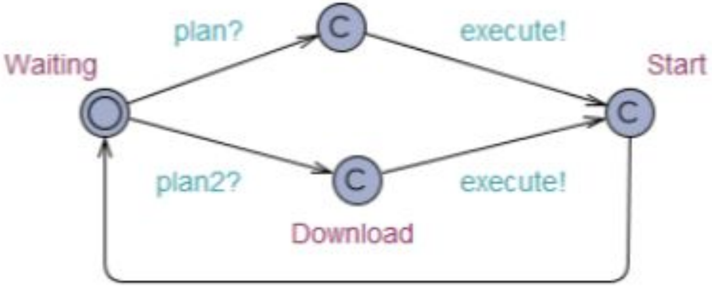
MAPE-K behaviors on the SHE platform



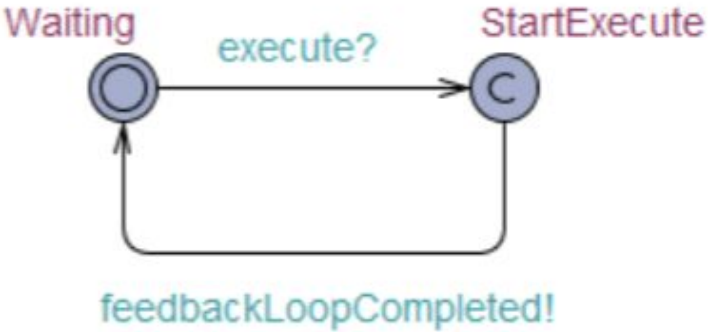
(a) Monitor



(b) Analyze

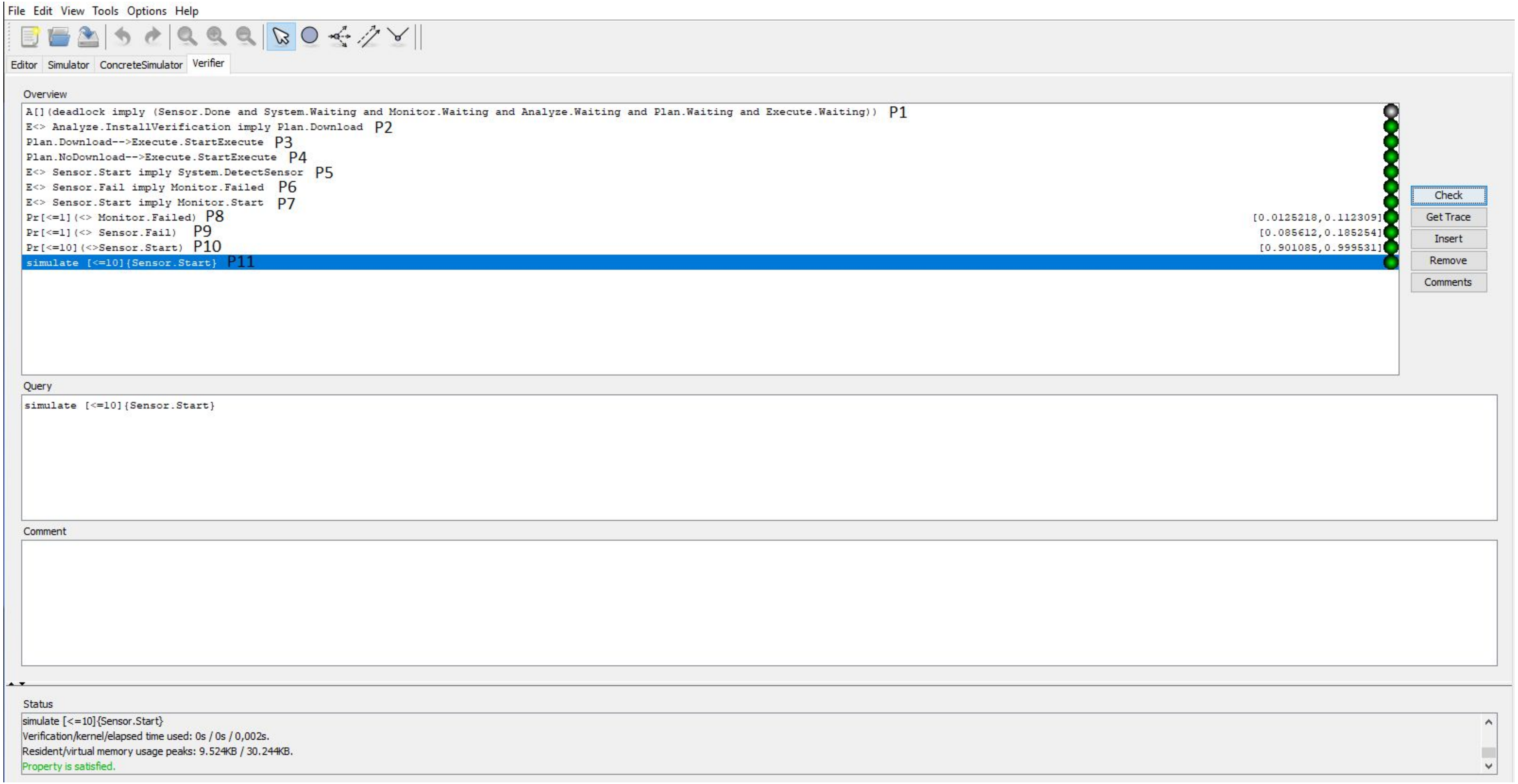


(c) Plan

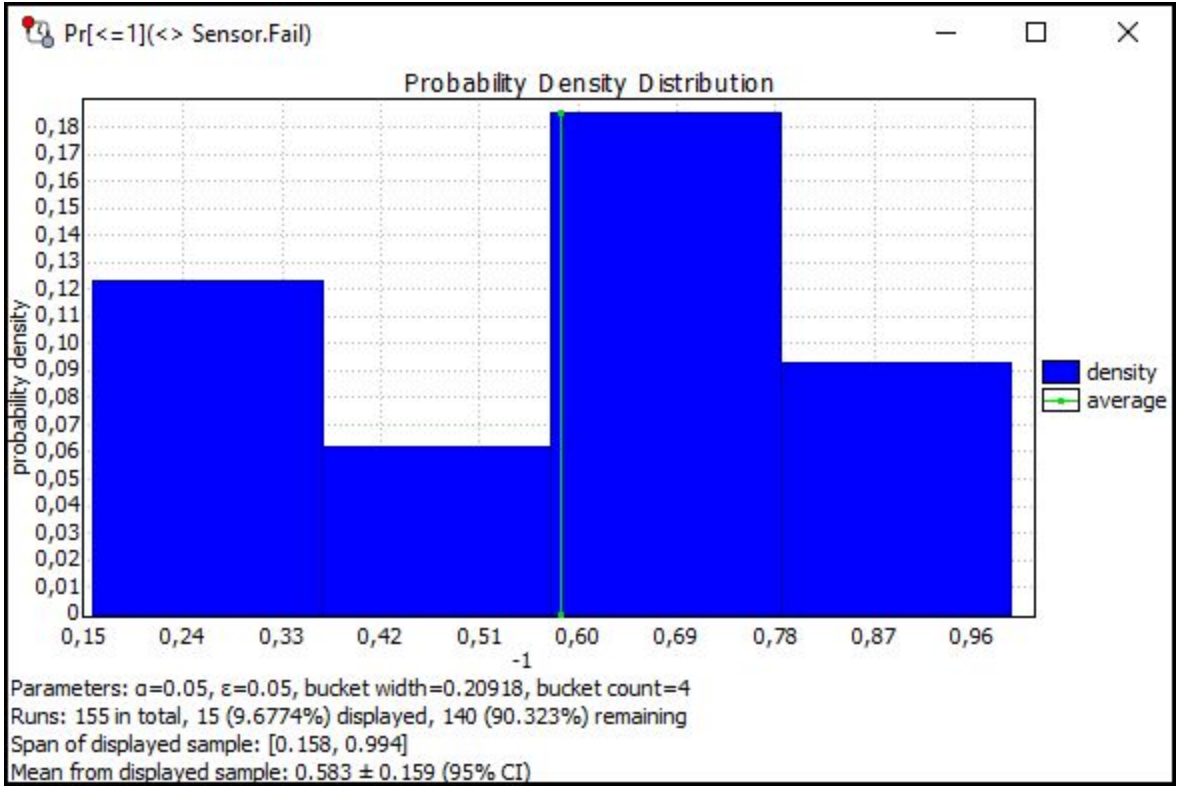


(d) Execute

MAPE-K behaviors on the SHE platform

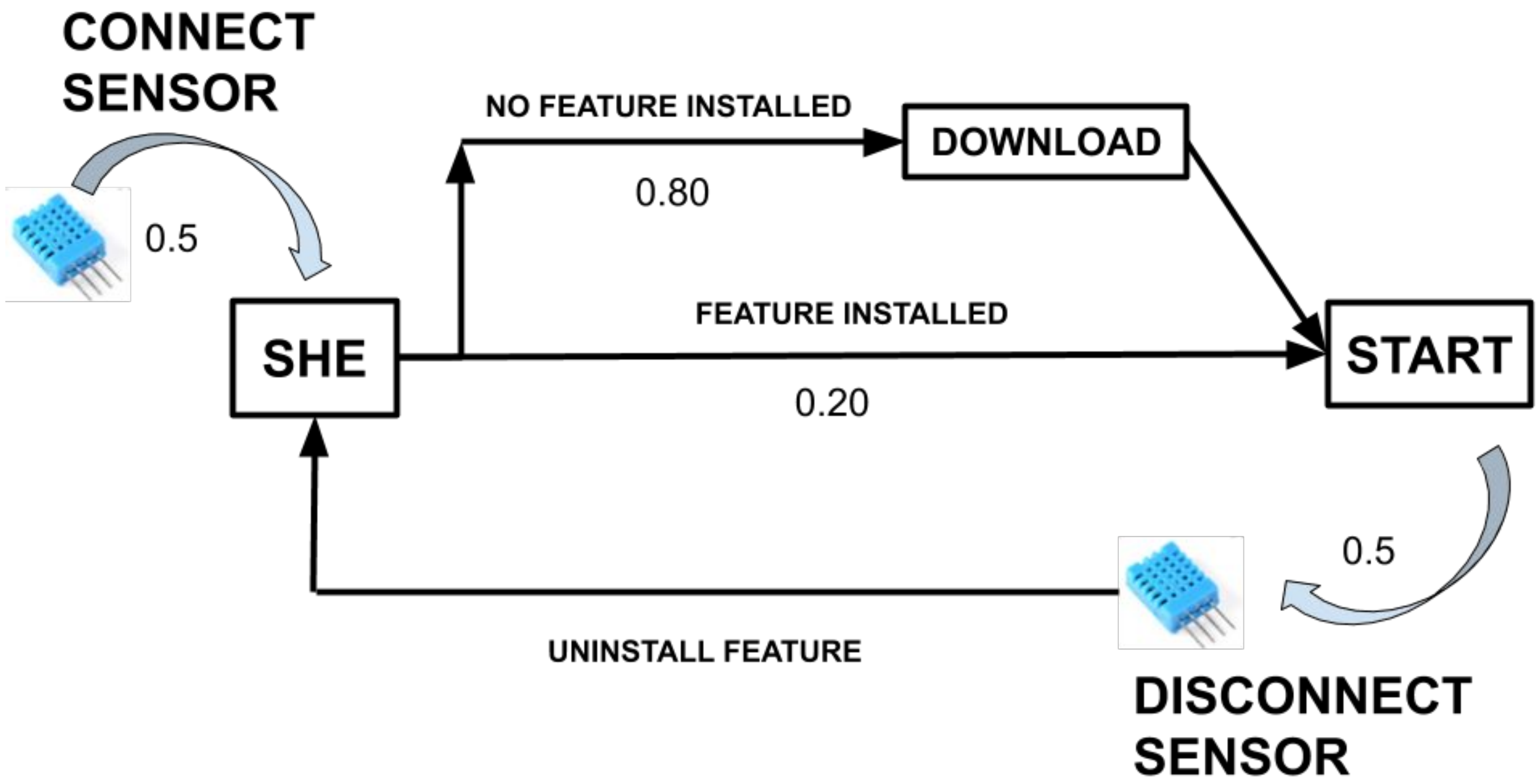


MAPE-K behaviors on the SHE platform

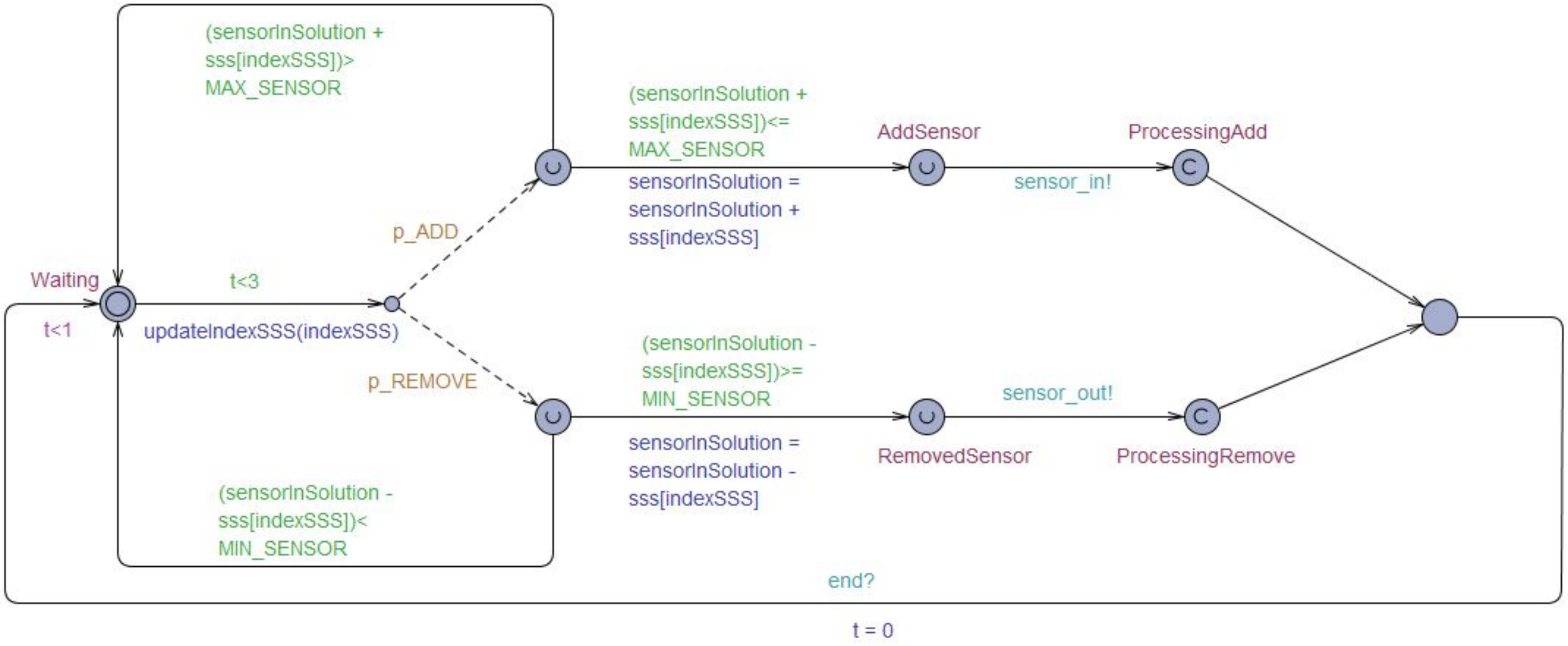


Propriedade de estimativa de probabilidade de falha.

Modelo de Simulação

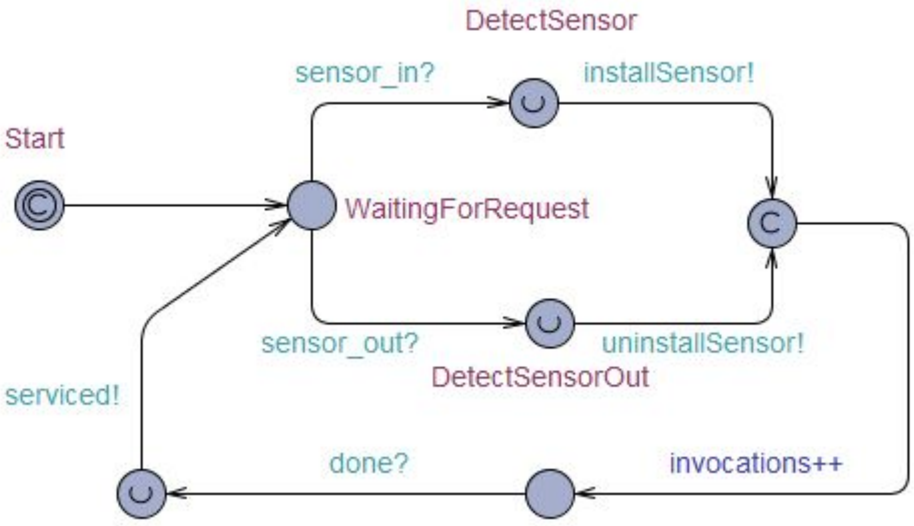


Modelo de Simulação



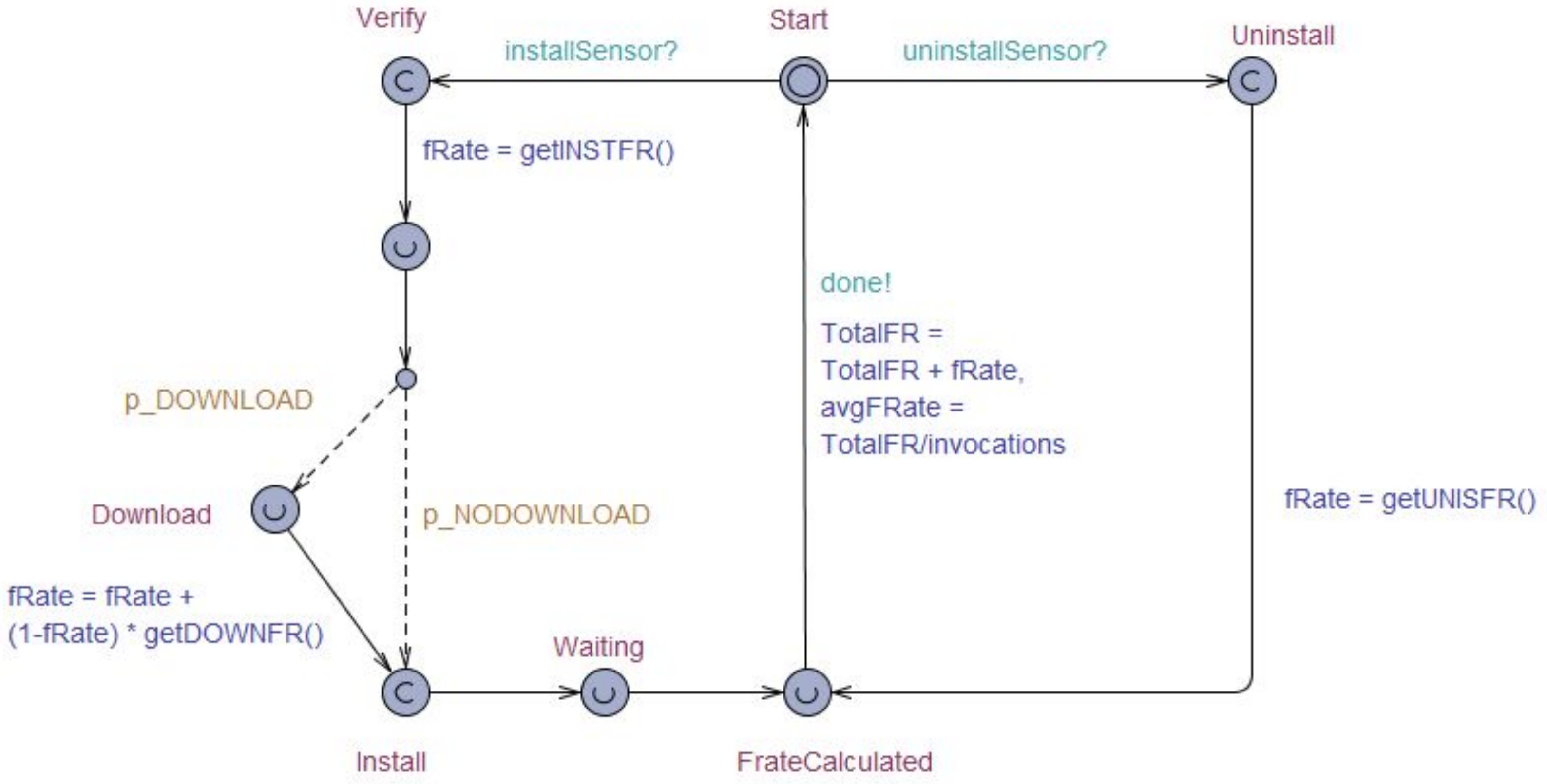
SHE - Environment model.

Modelo de Simulação



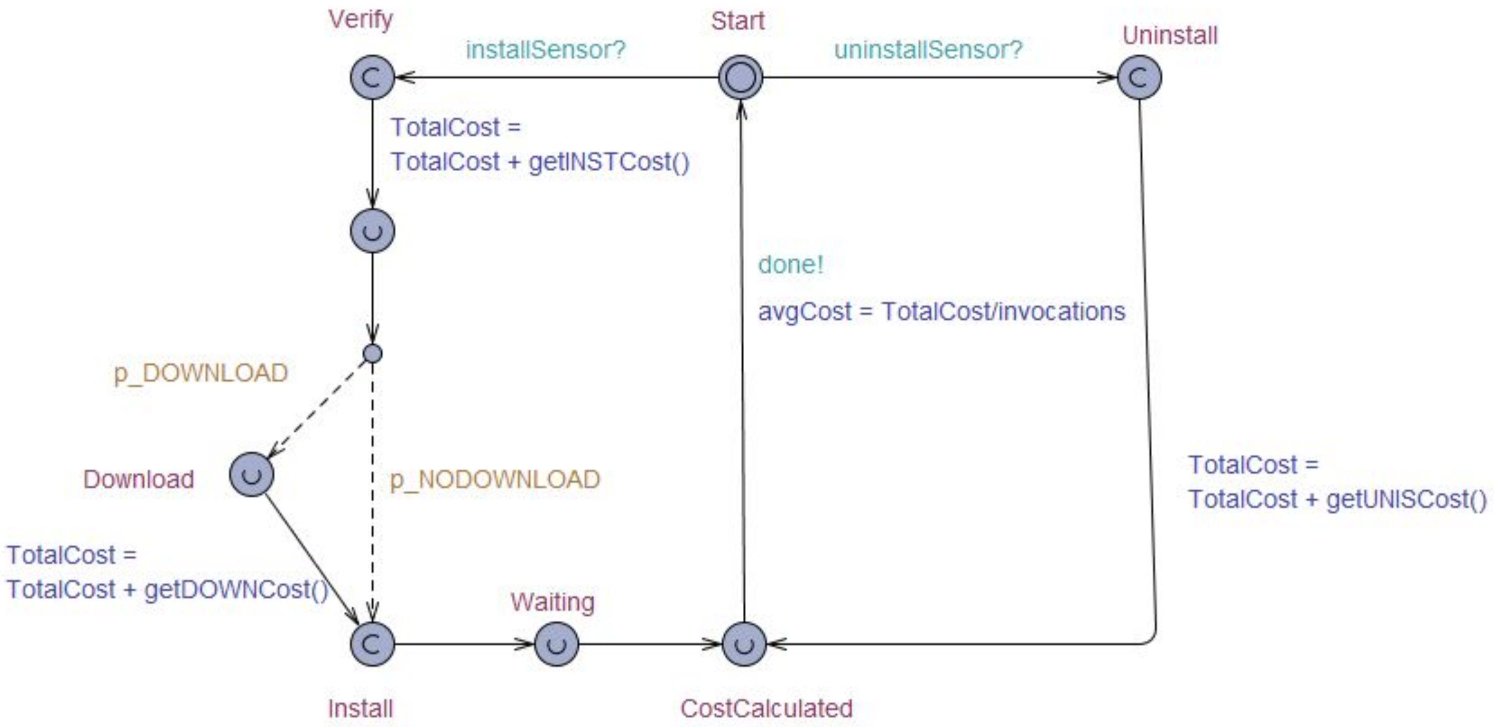
SHE - Managed System model.

Modelo de Simulação



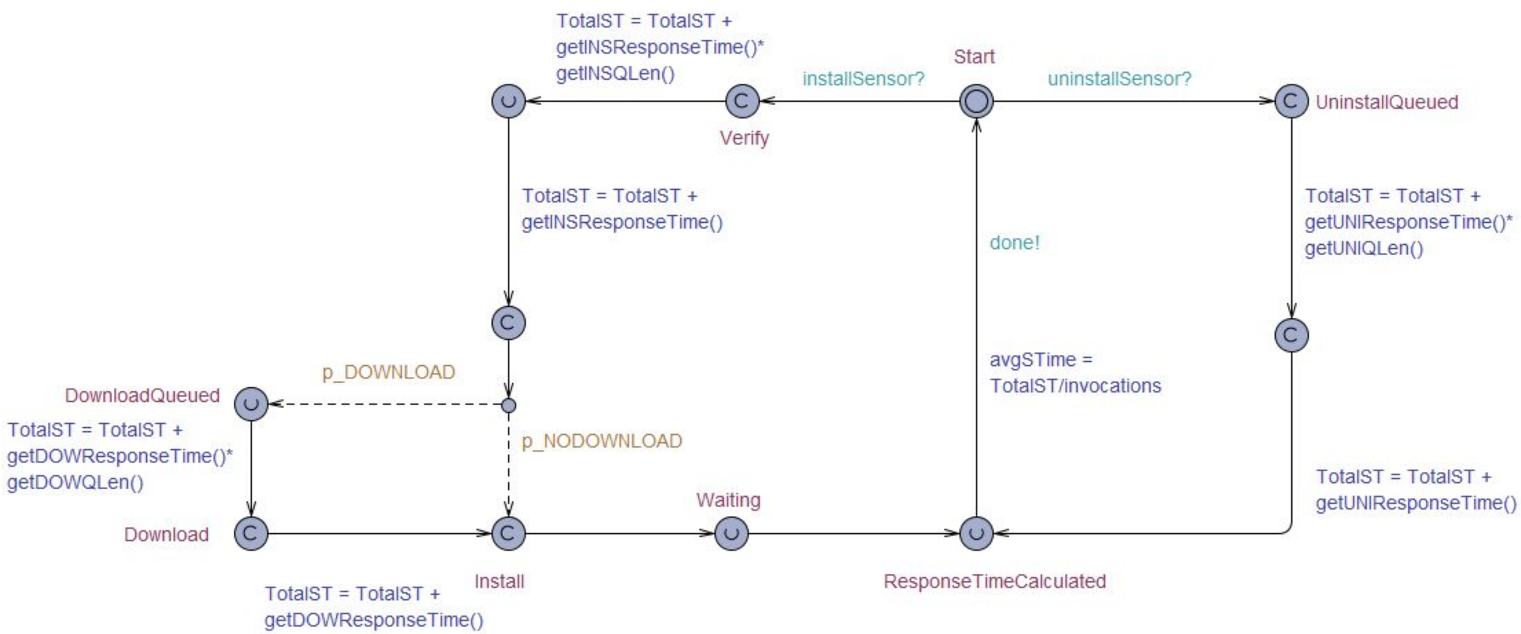
SHE - Assistance Feature model: Failure rate.

Modelo de Simulação



SHE - Assistance Feature model: Cost.

Modelo de Simulação



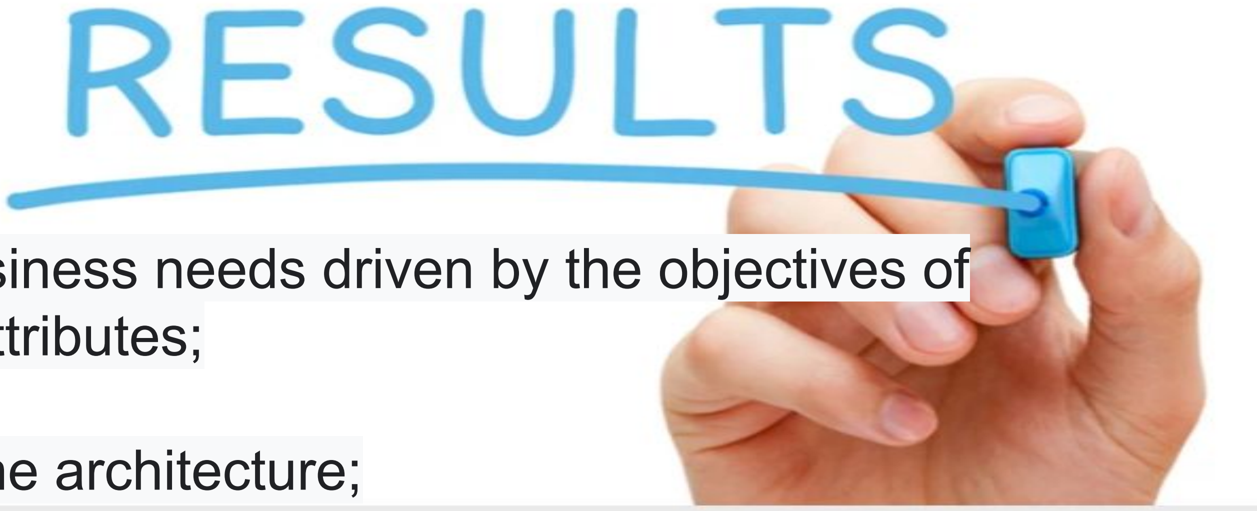
SHE - Assistance Feature model: Response Time.

Step 8 - Analyze Architectural Approaches.

In this phase, the architecture starts the process of mapping the most important scenarios raised in step 7. It is considered as a testing activity. If there was a lack of information that should be collected in the previous steps the team needs to go back to step 4 and work through it.

Step 9 - Present Results.

All the gathered information is organized and presented to all the stakeholders. The presentation includes all important decisions taken, i.e., the business context, driving requirements, constraints, and the architecture. In addition, the outputs of the ATAM, such as: the architectural approaches/styles documented, the set of scenarios and their priorities, the set of attribute-based questions, the utility tree, the risk discovered, the non-risks documented, the sensitivity points and tradeoff points found.



- Presentation of business needs driven by the objectives of high-level quality attributes;
- A presentation of the architecture;
- A Utility Tree that presented a priority list of quality attributes, such as performance, adaptability, cost, scalability and interoperability;
- Preparation of scenarios;
- Verification of quality attributes.

Results

- ❖ **Identified Risks**
- ❖ **Sensitivity points**
- ❖ **Tradeoff**

Do you think that ATAM-4SAS gives supports in the architectural assessment of a self-adaptive system?

Do you think that the ATAM-4SAS supports a strategic choice in the objectives of the attributes of qualities of a self-adaptive system?

Could ATAM-4SAS be used in industry?

Do you think the proposed method is easy to implement?

Would you change any steps? In other words, would you add or remove something from the proposed method?