2IMP30 Project – Digital Twin of a Smart Room (DTSR)

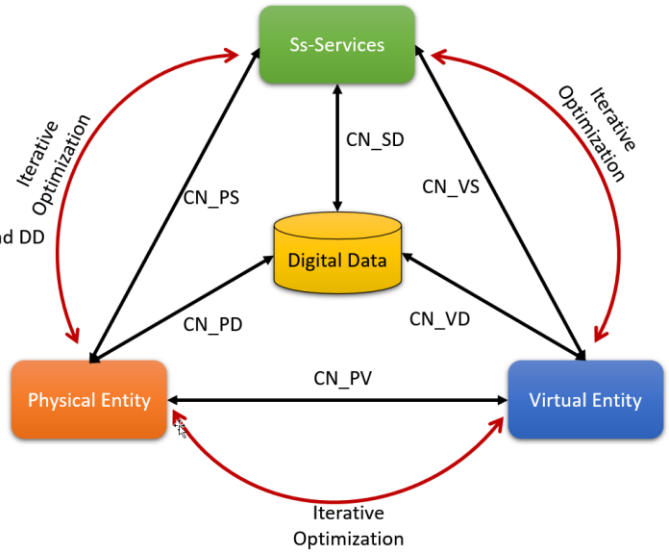
The advancement of intelligent building technology has become a crucial subject in the construction industry. By integrating smart technology, we can convert an ordinary building into an instinctive, self-rejuvenating, and eco-friendly one. Information technology plays a vital role in the development of intelligent buildings, affecting our work environment and the way we operate within it. These buildings can adapt to the needs of their occupants. Nevertheless, due to the intricate nature of designing an intelligent building, it's imperative to employ models that can depict the building's structure and functionality, including its services.

A model serves as a depiction of one or more concepts that can manifest in the real world. It defines a specific area of interest, where a particular field can correspond to an application domain. A model is essentially an abstraction, where it doesn't encompass all the intricacies of the entities being modeled within the area of interest, as well as the characteristics required to fulfill the intended purpose of the model.

Assignment

Employing the Systems Engineering technique that relies on the Model-Driven methodology (SYSMOD), create, execute, and validate a Digital Twin of Smart Room (DTSR). This DTSR combines the tangible elements of the Intelligent Meeting Room with their virtual counterparts, integrated into the Unity Game Engine. To achieve this objective, utilize the Agile SCRUM framework for project management. A Digital Twin comprises five fundamental elements, which are as follows:

CN_SD: Connection between Services (Ss) and DD
CN_PD: Connection between PE and DD
CN_VD: Connection between VE and DD
CN_PS: Connection PE – Services Data
CN_VS: Connection VE – Services Data
CN_PV: Connection PE and VE



For the DTSM we consider the following five components:

DT Components	Consists of
Physical Entity – PE (not given)	*Intelligent lamps, HVAC systems, fire sensors, movement detection sensors, CO2 sensors, and occupancy sensors.* *A system for communication and audio-video.* *A system for security and access.* *At least two touch panels enabling user interaction.*
Virtual Entity – VE	*3D CAD-CAM models of the PE; models of the software controllers*
Digital Data – DT	*The data collected from PE and VE;* *Different Data from specific data source: weather data*
Services - Ss	1. *Services offered to the user, services for the Physical Environment (PE), and services for the Virtual Environment (VE).* 2. *The air quality in the room is monitored and regulated by the DTSR.* 3. *The DTSR utilizes weather forecasts to intelligently regulate and supervise the room's temperature.* 4. *The temperature and lighting in the room are contingent on the presence of individuals in the room.* 5. *The user has remote access to all the services provided by the DTSR*
Connections	*Connections between the PE, VE, DT, and Ss:* *cybersecurity, network communication, network functionality, TCP/IP.*

Develop a SysML model of the DTSR that encompasses its primary structural features and services. Verify and validate the DTSR system, while limiting the focus to the development of the Virtual Entity and its services. The development of the Physical Entity is not within the scope of this project.

You will be working in groups of three students and collaborating to complete the project together. Upon completion, you will also be required to evaluate your peers.

Methodology

To effectively implement the DTSM system, it is necessary to adopt the subsequent approach:

1. Comprehend the problem and the context of the project. Identify the stakeholders of the DTSR and consider how users can interact with and utilize the DTSR. Employ the TRIZ 9 Boxes tool to identify and comprehend the current project and problem 's issues.
2. Create the Project's Context
3. Collect and model the DTSR requirements by utilizing SysML Requirement Diagrams. Categorize the requirements into various packages. Implement TRIZ tools to:
    1. Identify the contradictions that require resolution and utilize the TRIZ contradiction matrix to determine the TRIZ solution for the overall issue.
    2. Employ the suggested solution to solve the contradictions.
4. Create a System Context Diagram for the DTSR, and determine which actors (systems) interact with the DTSR (e.g., electricity, users). Employ a SysML Block Definition Diagram or an Internal Block Diagram to illustrate the system's context.
5. Create the DTSR's Operating Environment using a SysML Block Definition Diagram.
6. Model the services of the DTSR using Use Case Diagrams. Create an essential description for every Use Case.
7. Model the behavior to the DTSR using: Activity Diagrams, Sequence Diagrams, or State Machine Diagrams
8. Construct the architecture of the DTSR by employing a SysML Block Definition Diagram. Populate the block components with values, operations, and relationships.
9. Create the Mission Event Time of the DTSR.
10. Develop a graphical user interface (GUI) in DTSR Rhapsody that simulates a touch panel and demonstrates how a user can interact with the DTSR.
11. Implement and evaluate the model in IBM Rhapsody (utilizing the Visual Studio tool) and incorporate it into the Unity Game Engine Environment that has been provided to you.
12. Generate a concise project report, no longer than 20 pages.
13. Produce a brief video, up to 10 minutes in length, to showcase the functionality of the system and demonstrate your results.
14. Develop a brief PowerPoint presentation outlining your results, including your group number and members, along with their IDs.
15. Upload the IBM Rhapsody project (in a zipped file), along with the video(s), PowerPoint presentation, and Canvas report.