

Software Design Document For JARVIS



J. A. R. V. I. S.

FOR MBSE



Table of Contents

- 1. Introduction
 - 1.1 Purpose
 - 1.2 Statement of scope
 - 1.3 Software context
 - 1.4 Major constraints
- 2. System Overview
 - 2.1 System Components
 - 2.1.1 Hardware Components
 - 2.1.1.1 MPF
 - 2.1.1.2 Backend
 - 2.1.1.3 AR/VR
 - 2.1.1.4 AI
 - 2.1.2 Software Components
 - 2.1.2.1 MPF
 - 2.1.2.2 Backend
 - 2.1.2.3 AR/VR
 - 2.1.2.4 AI
 - 2.2 Goals and objectives
- 3. Design Considerations
 - 3.1 Design Assumption and Dependencies
 - 3.2 Design Constraints and Limitations
 - 3.3 Design Goals and Guidelines
- 4. Data Design
 - 4.1 Data Organization
 - 4.2 Database Schema
- 5. Architectural and component-level design
 - 5.1 Architecture diagram
 - 5.2 Component Description
 - 5.2.1 Component 1: Multi-Platform Editor
 - 5.2.1.1 Processing narrative
 - 5.2.1.2 Interface description
 - 5.2.1.3 Processing detail



- 5.2.1.4 Dynamic Behavior
 - 5.2.1.5 Interaction Diagrams
 - 5.2.2 Component 2: Backend
 - 5.2.2.1 Processing narrative
 - 5.2.2.2 Interface description
 - 5.2.2.3 Processing detail
 - 5.2.2.4 Dynamic Behavior
 - 5.2.2.5 Interaction Diagrams
 - 5.2.3 Component 3: MBSE Assistant
 - 5.2.3.1 Processing narrative
 - 5.2.3.2 Interface description
 - 5.2.3.3 Processing detail
 - 5.2.3.4 Dynamic Behavior
 - 5.2.3.5 Interaction Diagrams
 - 5.2.4 Component 4: AR viewer
 - 5.2.4.1 Processing narrative
 - 5.2.4.2 Interface description
 - 5.2.4.3 Processing detail
 - 5.2.4.4 Dynamic Behavior
 - 5.2.4.5 Interaction Diagrams
- 6. User interface design
 - 6.1 Overview of the user interface: MPF
 - 6.1.1 Screen images
- 7. Libraries and tools
- 8. Appendix
- 9. References



J. A. R. V. I. S.

FOR MBSE

Revision History

| Name | Date | Reason For Changes | Version |
|-------------|------------|--------------------|---------|
| JARVIS Team | 30-10-2018 | Initial Draft | 1.0 |
| | | | |
| | | | |



1. Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the design for the “JARVIS” software. It will present the purpose and complete declaration for the development of the system.

1.2 Statement of Scope

With the advancement in the field of information and communication technologies emerges the development of intelligent technical systems (ITS). ITS need to integrate mechanical, electrical, and software components in order to realize adaptive, robust, anticipatory, and user-friendly behavior.

With these objectives, the scope of “JARVIS” is defined to provide an ease of understanding and a common framework for using model based system development software for all the domain people. Also an additional enhancement to the developed models can be achieved through the recommendations provided by the MBSE assistance.

2. System Overview (Daniel)

JARVIS for MBSE is a product developed for all the users involved in Model Based System Engineering. Users, irrespective of their engineering discipline would be able to create CONSENS models in the MPF (Multi-Platform Frontend) editor which is the User Interface. The MPF is easy to understand for engineers coming from all kinds of engineering disciplines and comes with enhanced user functionality in the form of recommendations. JARVIS provides an intuitive and interactive model based approach for systems engineering with system model creation tools, dynamic, UWP based editorial GUI's, distributable to all integrated Windows 10 platforms. It uses advanced user interaction and visualization techniques and has the ability to assist engineers in their engineering process.

All the user designed models will be stored in the backend and will be technically refined and developed into SysML models with CONSENS profiling.

MBSE assistance would provide recommendations to enhancing the system design based on the existing models.

Most of the time user designed Consens models are very complex and consist of many elements that are related to each other. To get a better overview the advantages of Augmented Reality are used. The large 3D free space allows the arbitrary arrangement of many elements. Due to the offered freedom of movement in 3D space, a user can view certain parts of the models by simply walking towards it and thus understand models and their relations faster and easier.



The above mentioned components should collaborate with each other to complete output. Furthermore, the models should be stored somewhere to access or manipulate them. This functionality is performed by Backend.

This product is very useful especially during workshops and its review processes. Due to the (often) limited time within workshops a quick and easy to understand MBSE process is required. In order to achieve this a user is given the MPF editor, allowing him/her to model Consens models by hand drawing shapes and writing texts which are directly converted to Consens models instead of having to click through several menus as offered by many tools so far.

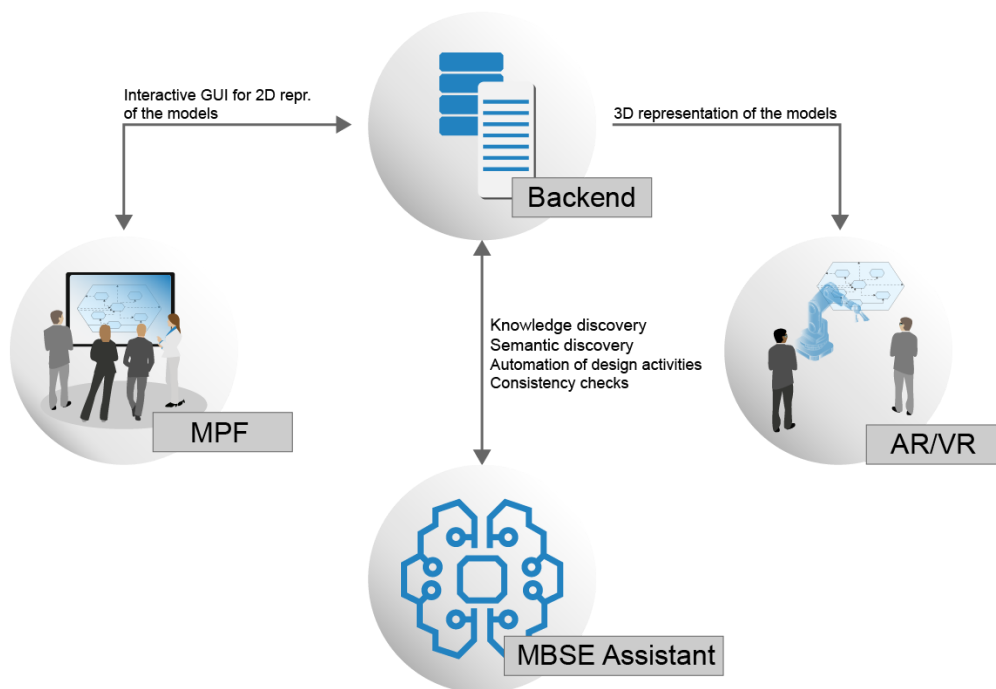


Figure 1: Image of JARVIS Overview (Picture: Daniel)

2.1 System Components

This section describes the four components the JARVIS system consists of. The MPF is the part of the system in which engineers can create CONSENS models. The backend is used to store the models, the AI makes suggestions for users of the MPF and the AR viewer allows a user to look at the model in AR.

2.1.1 MPF

The Multi-Platform Frontend, short MPF, is used by engineers from different engineering backgrounds to easily create CONSENS models. It provides various kinds of techniques to add new elements to the model, for example with drag-and-drop from a side panel or by drawing an element by hand. The MPF runs on all Windows 10 devices.



2.1.2 Backend

The backend stores all models users create on a designated server. The other components can use HTTP requests to store a model on the backend, update an existing model or download an existing model. To store the models, Papyrus is used, which is an additional plugin to eclipse. RPC protocols will be used to link the server and the MBSE assistant.

2.1.3 AR viewer

To have a better visualization of the created models and relations between them, an AR viewer is provided. After starting the AR viewer app on a Microsoft HoloLens the user sees a list of projects they can work on. The user can load a model from the backend which the AR viewer then shows as a 3d representation in the environment of the user.

2.1.4 MBSE assistant

The models stored in the backend are also stored as a Knowledge Graph using Neo4j. This allows the AI component to provide recommendations to a user of the MPF on how to improve the model that they create. The functionalities of AI can be visualized in the MPF editor and MPF is connected to AI via Backend.

2.2 Goals and objectives

JARVIS for MBSE is a product developed for all the engineers involved in Model Based System Engineering. Its objectives are:

- Non-IT professionals from different engineering backgrounds should be able to draw and analyze new models and update old ones for the purpose of system designing without the need for IT professionals.
- The users can be helped by providing the recommendations from AI to improve system design based on prior experiences.
- Engineers do not need to set the environment for modeling.
- System designs are developed from the beginning till end in digital models without the need of pen and paper.

3. Design Considerations

This section describes the issues surrounding the design of the system and what need to be taken in consideration. It shows the assumptions that are taken, constraints that need to be kept in mind and the goals that were followed when designing.

3.1 Design Assumption and Dependencies

Assumptions are statements that the design team will not spend time or effort in verifying. These statements will simply be accepted as facts. Following are the assumptions of system to be developed:

- Touch screen monitors (Surface Hub) and the working microphone feature should be available for MPF



3.2 Design Constraints and Limitations

Design constraints are conditions that need to happen for a project to be successful. Design constraints help narrow choices when creating a project. Following are some of design constraint:

- There should be backup server for Backend if main server fails
- For MBSE Assistant heavy processing will be done in limited time. This computer must have more than two CPU cores to handle concurrent processing better.
- (Sven) Since the AR viewer is used on a Microsoft HoloLens, the performance and screen size the AR viewer can use are limited to a relatively weak processor and only about 35 degrees field of view. Also, the interaction with the HoloLens is restricted because the user should have their hands free, so mouse and keyboard can not be used for interaction with the user.

3.3 Design Goals and Guidelines

- **Simplicity:** Regarding MPF and AR/VR, an important design goal is keeping it simple. User interface should not be so complicated.
- **Performance:** System should have a good performance such that user can receive the recommendation from MBSE Assistant within 10 seconds of requesting through MPF.

4. Data Organization

When a new project or UML diagram is created and saved in Backend a new record is created in MySQL database containing important information and to uniquely identify each UML diagram or project. Whenever that particular object is updated or deleted the record in database is amended as well.

5. Architectural and component-level design

5.1 Architecture diagram

The system architecture diagram is shown below:

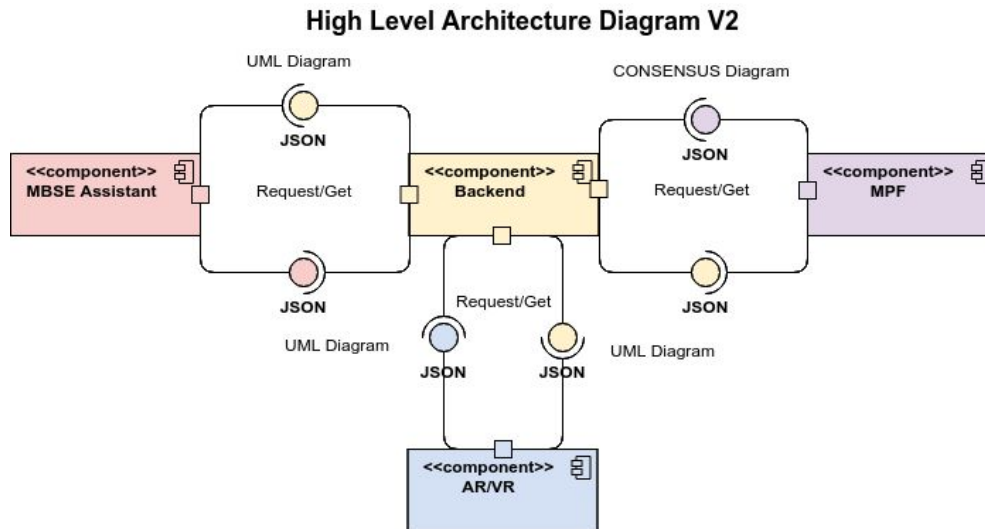


Figure 2: Image of System Architecture

All the components will communicate with other components through Backend. The interface between all the components is JSON format. For processing in the Backend, it will first be transformed from JSON to UML and then processed. The UML model will be converted to JSON format if it is to be sent to other components.

5.2 Component Description

5.2.1 Component 1: Multi-Platform FrontEnd

5.2.1.1 Processing narrative

- MPF is responsible for providing User Interface to the users. Provide an easy to use UI editor that anybody can work with.
- MPF provides an editor to create and save new models, load the existing models from backend and save the updated models to the backend.
- MPF also receives the suggestions from Backend which come from MBSE Assistant and provides them to the user while creating a model in the case of a familiar model [FR-9,TR-4].
- MPF makes the interaction with the UI Editor easy by providing the Shape recognition[FR-2,TR-6], Speech recognition[FR-4,TR-5], Text recognition [FR-3,TR-7] and Drag & Drop[FR-1,TR-8] features to the users.

5.2.1.2 Interface description

- MPF communicates using HTTP request using methods such as GET, POST, PUT and so on depending upon the request with Backend[TR-12].



- It stores model in the form of UML/SysML but components send the model in JSON format.

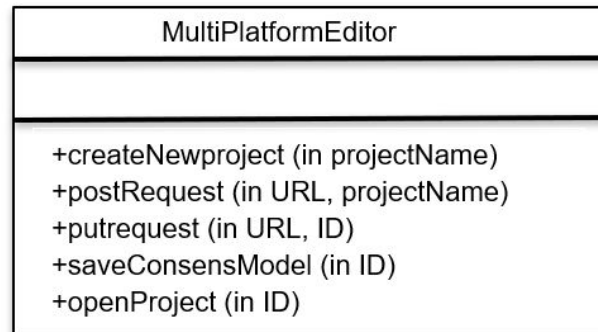


Figure 3: Interface description for Multi-Platform Editor

5.2.1.3 Processing detail

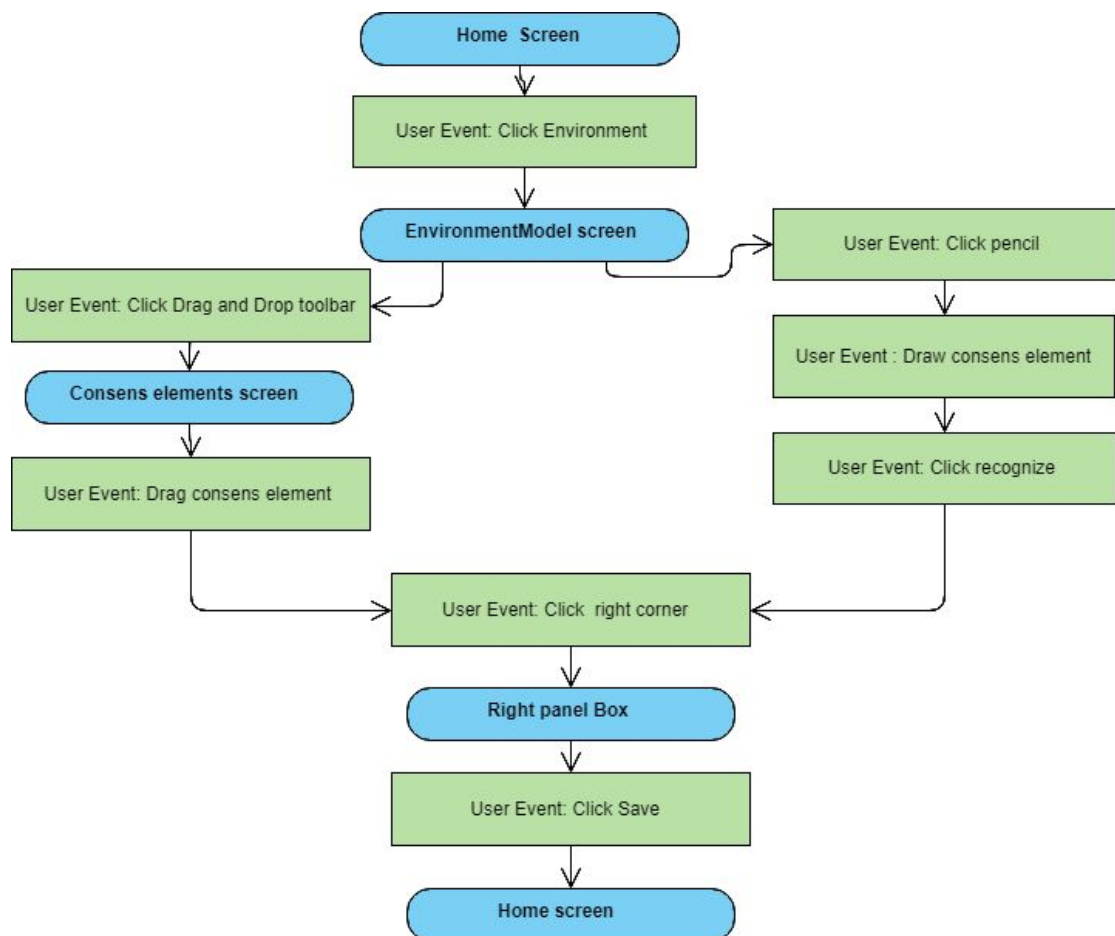


Figure 4: Functional pipeline for MPF



5.2.1.4 Dynamic Behavior

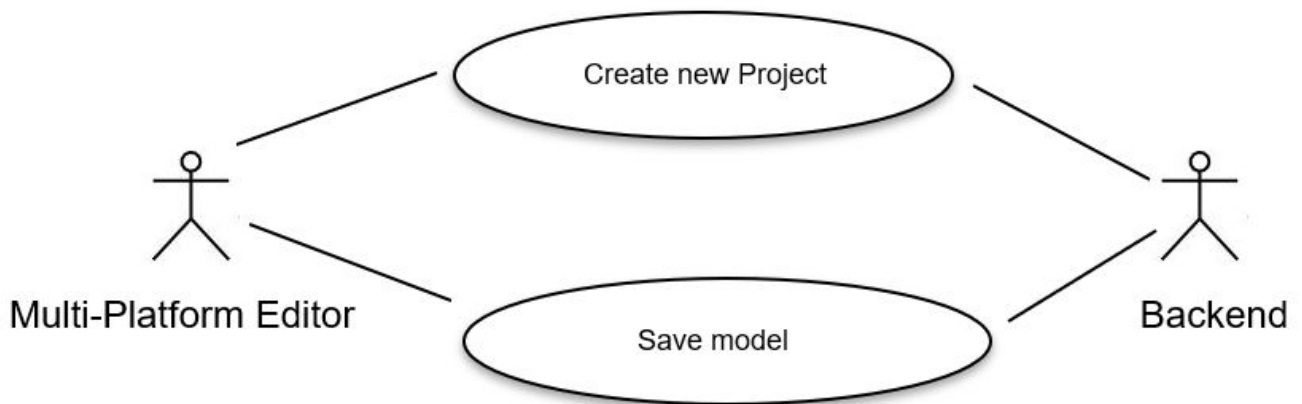


Figure 5: Use case 1 for MPF [FR-5,TR-1,TR-2]

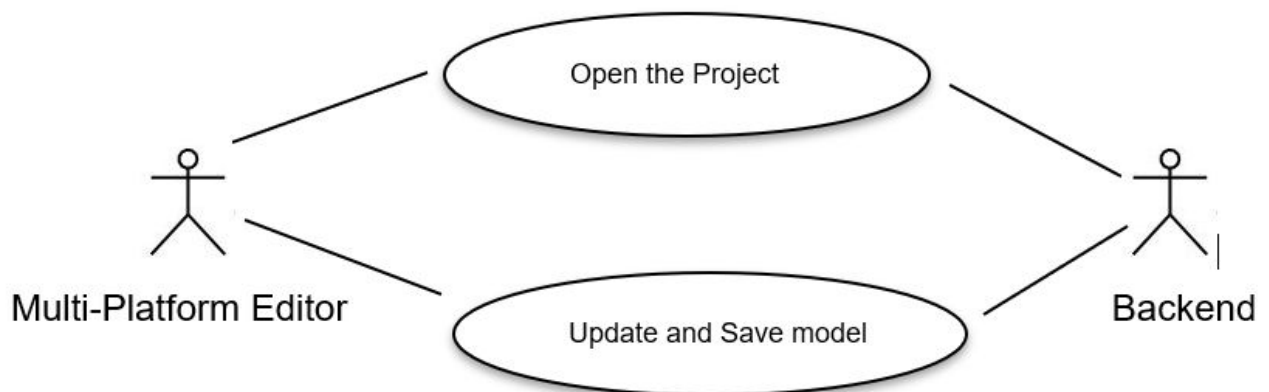


Figure 6: Use case 2 for MPF [FR-6,FR-7,TR-2]



5.2.1.5 Interaction Diagrams

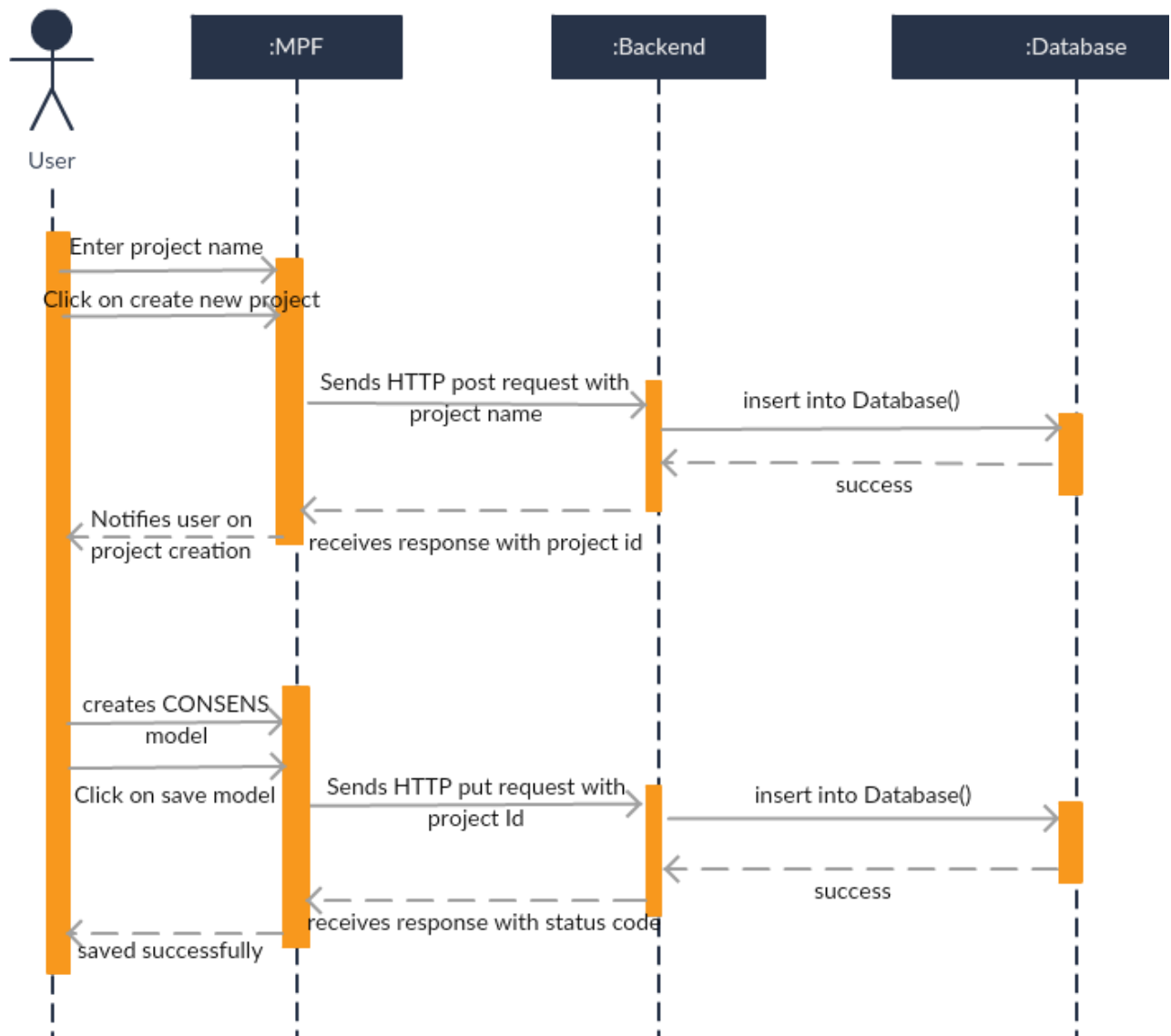


Figure 7 : Interaction diagram for MPF Use Case 1

Use case 1: User enters the Project name and clicks the Create new Project [TR1] on the Home Page which in turn invokes a http post request to backend to create the Project. MPF receives the project Id as the response when it is successful. User creates a model and click save and that invokes a http put request with the project ID[FR-5,TR-2].

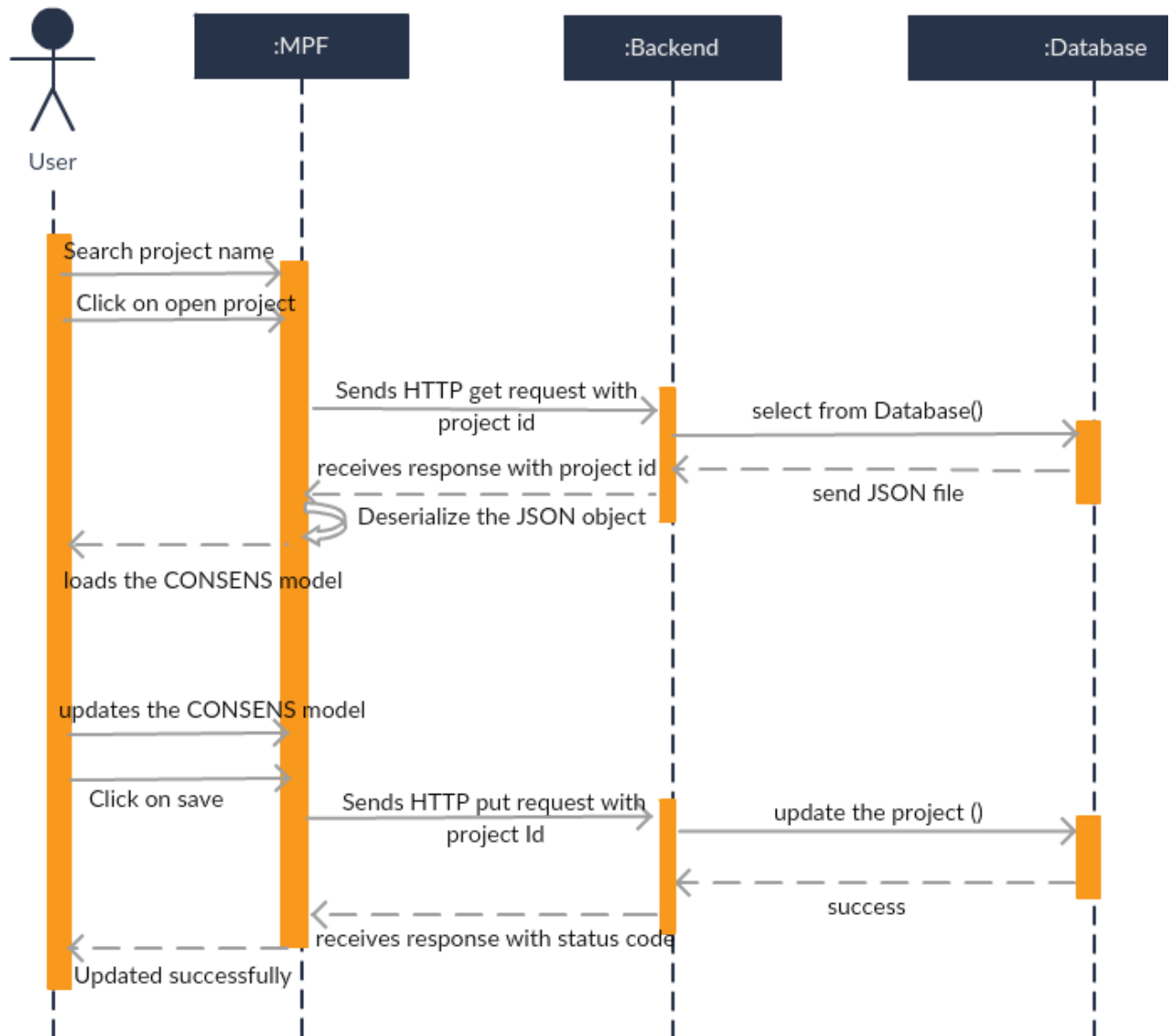


Figure 8: Interaction diagram for MPF Use Case 2

Use case 1: User opens the existing Project by sending a http get request to the backend [FR-6] and modify/ update the model and clicks Save which in turn invokes a http put request to backend to update the existing Project [FR-7,TR-2].

5.2.2 Component 2: Backend

5.2.2.1 Processing narrative

- Backend is a persistent data storage
- All other component communicate with Backend.



- When a CONSENS model is drawn in MPF it will be sent to Backend and stored there. They can be sent back to MPF from Backend for further processing.
- After models are stored in Backend from MPF, then they can be sent to MBSE Assistant for recommendation. Then MBSE sends back recommendation to Backend. Finally backend sent them to MPF.
- When the recommendations are complete and models are finalized then AR/VR request final model to show to users in 3D form.
- It also updates, deletes and replaces present models.
- It can also store views.
- It can also communicate between MBSE Assistant and MPF for showing requirement consistency to user.

5.2.2.2 Interface Description.

- It will communicate with all other components using HTTP request using methods such as GET, POST, PUT and DELETE
- It stores the Consens models as Papyrus SysML Models.

| Backend |
|------------------------------|
| +ConvertUMLtoJSON(file UML) |
| +ConvertJSONtoUML(file JSON) |
| +RequestRecommendation() |
| +SendUMLtoMBSE() |
| +SendModeltoARVR() |
| +SendModeltoMPF() |
| +SendRecommendationtoMPF() |
| +CreateProject() |
| +SaveProject() |
| +CreateUML() |
| +UpdateUML() |
| +DeleteUML() |
| +CreateRecord() |
| +UpdateRecord() |
| +DeleteRecord() |

Figure 9: Interface description for Backend

5.2.2.3 Processing detail

5.2.2.4 Dynamic Behaviour

5.2.2.5 Interaction Diagrams

5.2.2.6 Consens model to SysMI

5.2.3 Component 3: MBSE Assistant

5.2.3.1 Processing narrative



- AI is responsible for providing assistance to MPF editor. Functionalities that AI is going to support in this system are text recognition[FR-8,TR-4], Speech recognition[FR-1,TR-1], shape recognition[FR-7,TR-3], providing recommendations/ suggestions while adding an element[FR-6,TR-2].
- For recommendation system, the models are stored as Knowledge graphs[FR-5,FR-9,TR-2].
- AI makes the interaction with the MPF Editor via Backend whereas rest of the functionalities are incorporated in the MPF editor itself[FR-1,FR-2,FR-3].

5.2.3.2 Interface description.

- AI communicates using HTTP request on the ip address that the backend is running upon[TR-5].
- The backend then sends us all the project id's it has stored and AI request the project based on those id's[TR-5].
- Backend sends us the response in the form of a JSON and in AI component, JSON is parsed and is stored in the form of a Knowledge Graph[TR-5].

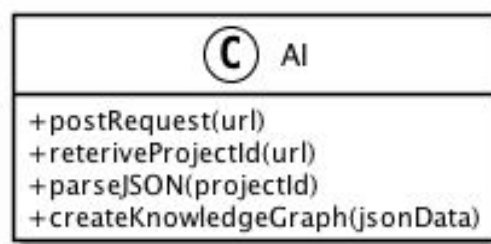


Figure 10: Interface description for AI

5.2.3.3 Processing detail

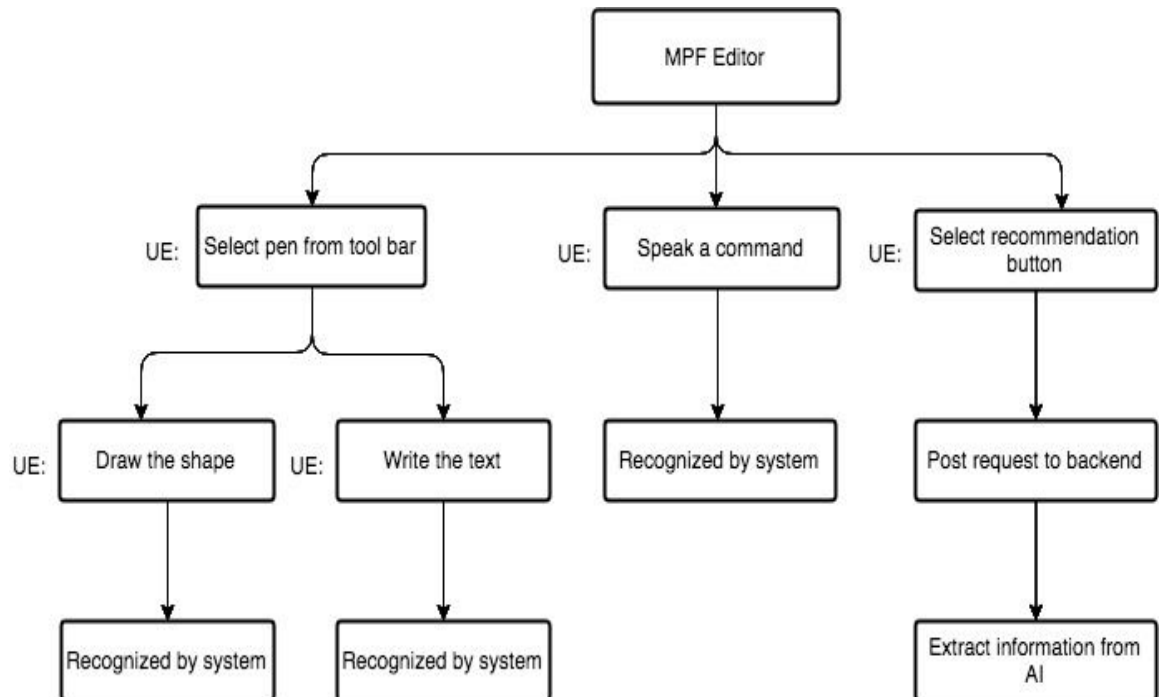


Figure 11: Functional pipeline for AI

5.2.3.4 Dynamic Behavior

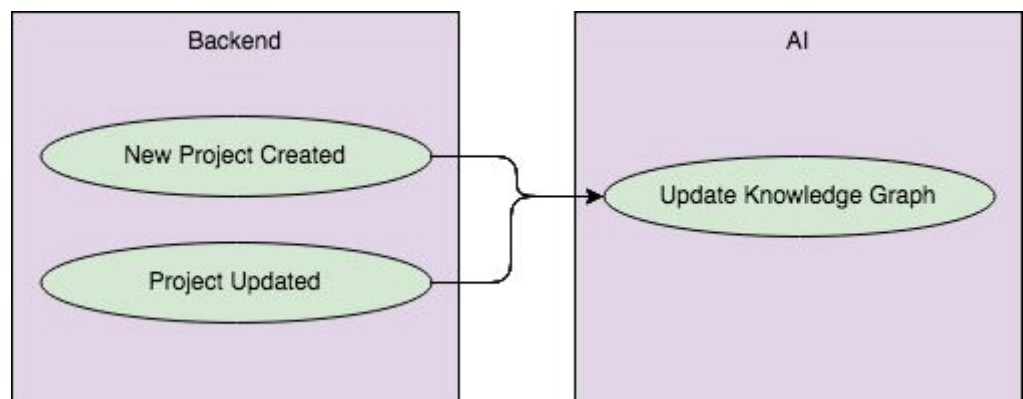


Figure 12: Use-Case diagram for AI

5.2.3.5 Interaction Diagrams

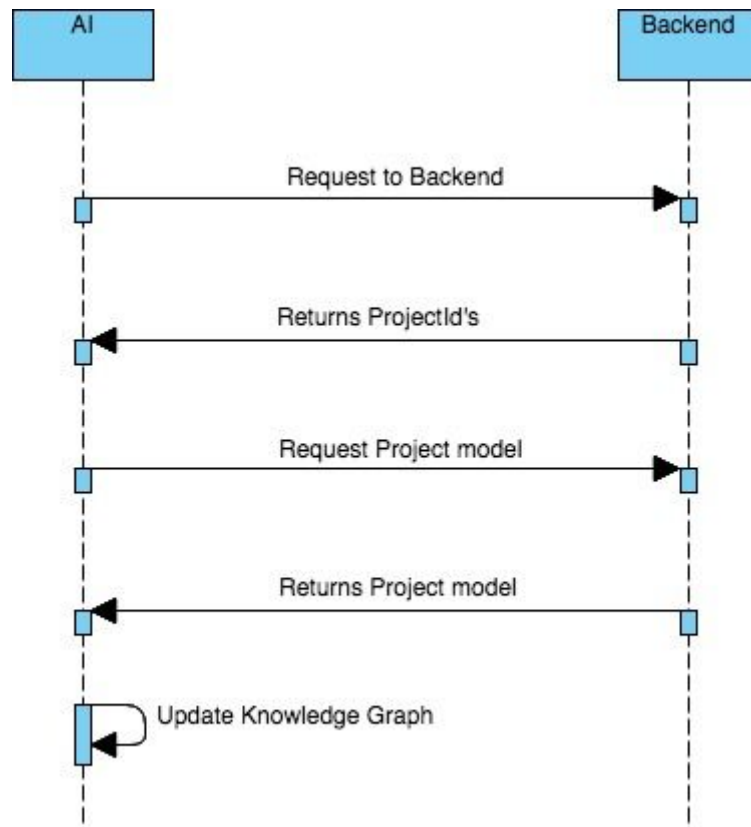


Figure 13: Interaction diagram for AI

AI component will poll the backend for checking any new project model or update in the existing project model, in response to which the backend responds with the list of project id's. AI checks the project id's and requests to the backend for retrieving the project model data and updates the knowledge graph on receiving data from backend model.

5.2.4 Component 4: AR viewer (Sven)

5.2.4.1 Processing narrative

The AR viewer allows a user to look at the model in a different way. When using a Microsoft HoloLens, the user's hands are free and they can move around the room freely. This allows for the model to be arranged in the whole room, giving a good detailed view on the model. A menu will be provided in which the user can choose which part of the model they want to see [FR7, FR8] and to allow the user to switch to a different project [FR6].

5.2.4.2 Interface description.

The AR viewer communicates with the backend via HTTP requests over the internet, sending and receiving data as JSON strings. Upon starting, the AR viewer sends an HTTP GET request to the backend to receive the projects the user can work on. This fulfills FR1 and TR2. The projects are returned as a JSON string containing a list of possible projects. After the user chose the project

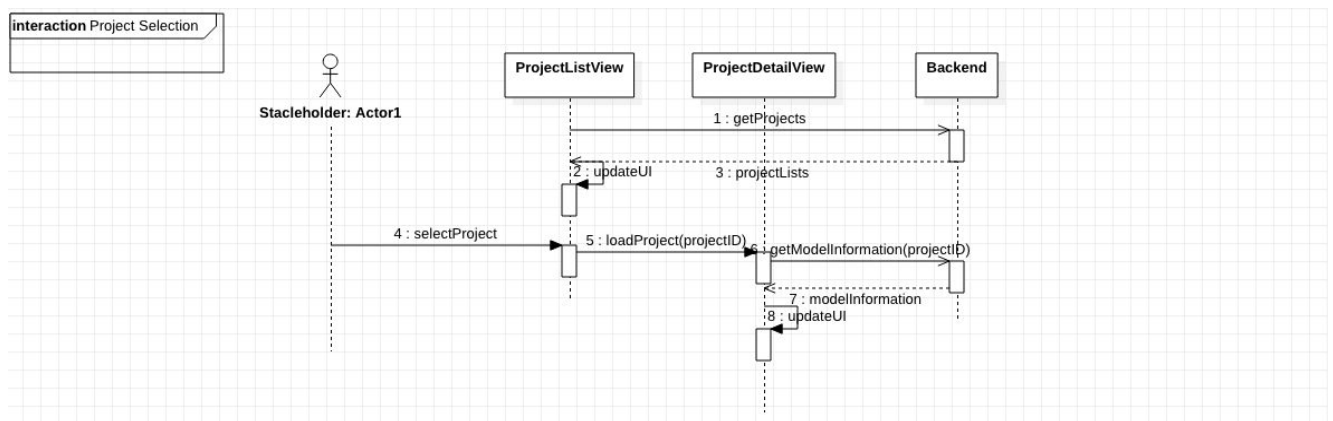


[FR2] they want to work on, the AR viewer sends an HTTP GET request to the backend to receive the respective model. The model is returned as a JSON string which contains all the information about the model stored in the backend. [FR3, FR4, FR5, TR3, TR4]. This includes all the submodels and all model elements of the requested model. TR1 is fulfilled by developing the app for the Microsoft HoloLens.



Figure 14: Interface description for AR viewer (Sven)

5.2.4.3 Interaction Diagrams



6 User interface design

6.1 Overview of the user interface: MPF

User can interact with MPF (Multi-Platform FrontEnd) by touch (Surface Hub), Mouse click, Drawing and Speech Recognition.



J. A. R. V. I. S.

FOR MBSE

6.1.1 Screen images

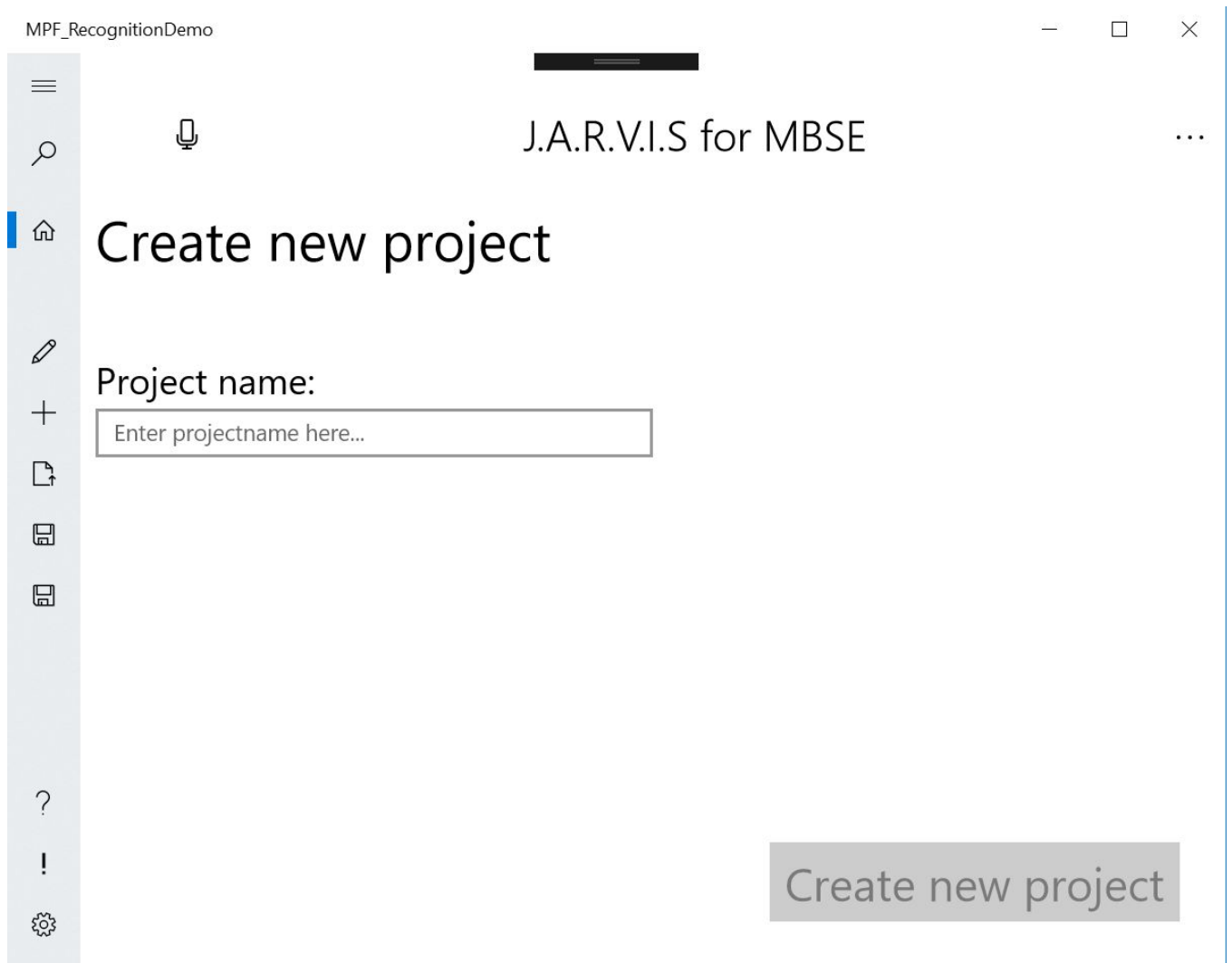


Figure 15: Image of MPF Home Screen

The above figure describes the Home screen of JARVIS. User is provided with many options in the home page like settings, help, info and etc.

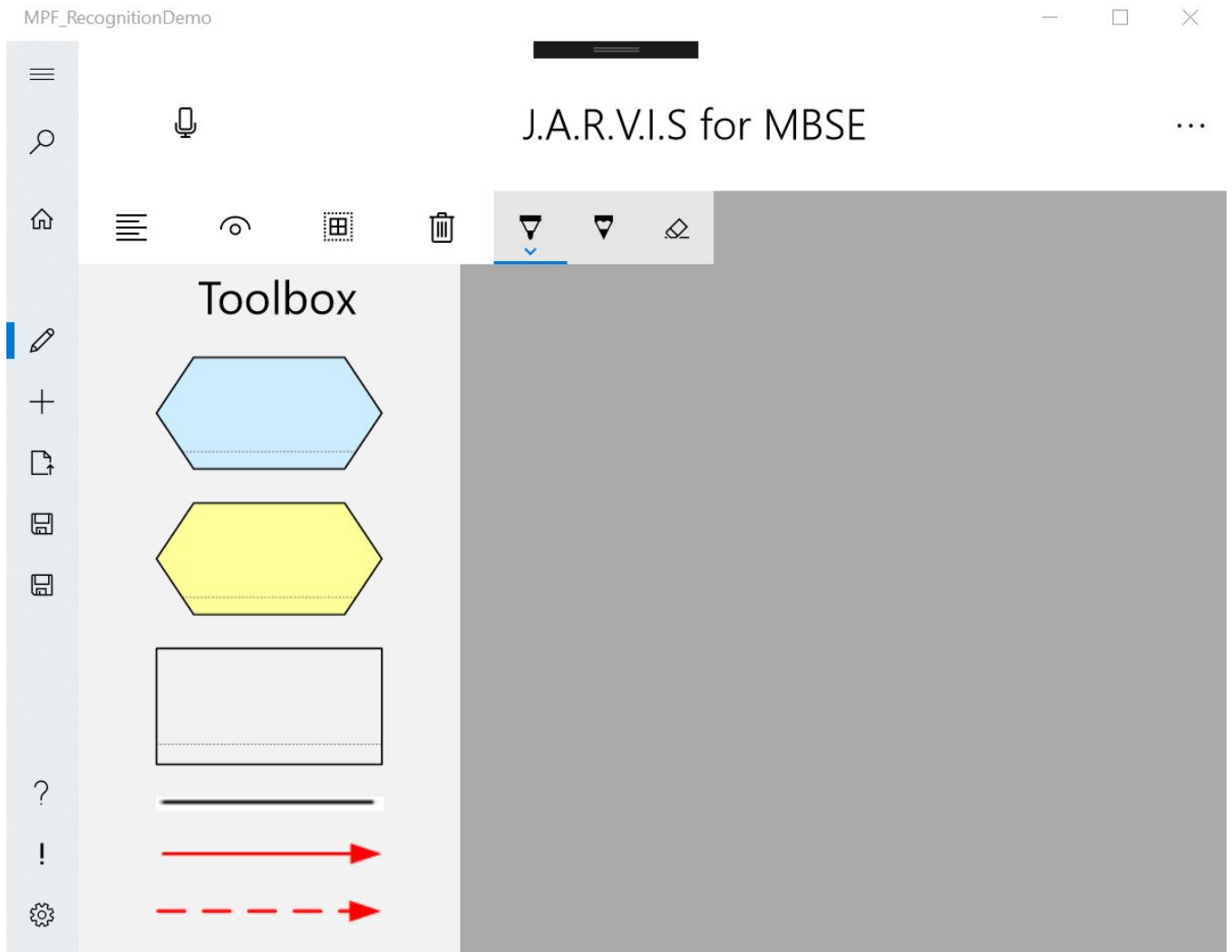


Figure 16: Image of MPF Environment Model Screen

The above figure describes the Environment Model Page where an user is allowed to create Environment partial model of CONSENS.

6.2 User Interface: AR viewer (Sven)

The user can choose one of the projects shown to them at startup of the app with the air tap gesture integrated in the HoloLens. After choosing a project, the model gets shown as a 3d representation in the environment of the user. The user can choose which part of the model should be shown with air tap on a menu that also gets placed in the environment of the user.

7. Libraries and tools