OOSE BASED APPLIICATION DEVELOPMENT CASE STUDY-2

Text Generation for Technical Orders from DEPTH Simulations

MAY 4, 2024

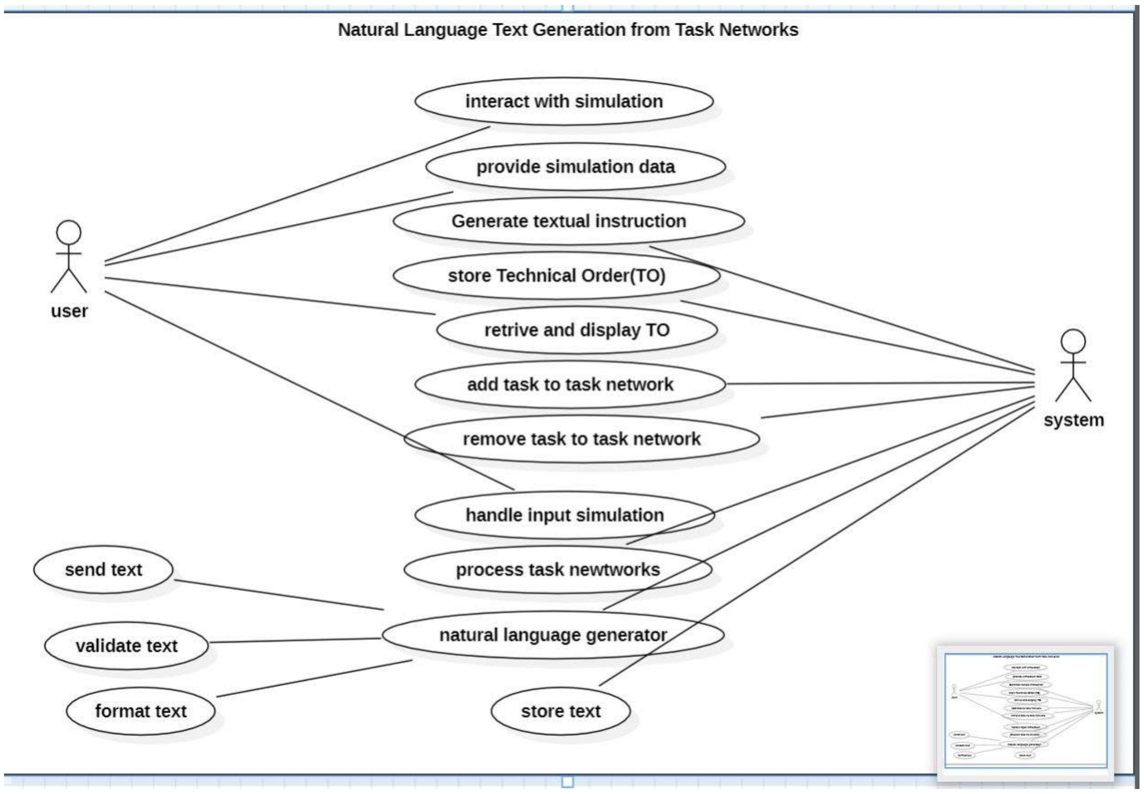
HU21CSEN600108

Sr.No	Title	Pg.No	Date of Performance	Date of submission	Remark	Sign
1						
2						
3						
4						
5						
6						
7						
8						
9						

Final Report: Feasibility of Natural Language Text Generation from Task Networks for Use in Automatic Generation of Technical Orders from DEPTH Simulations

TITLE: Text Generation for Technical Orders from DEPTH Simulations

USE CASE DIAGRAM:

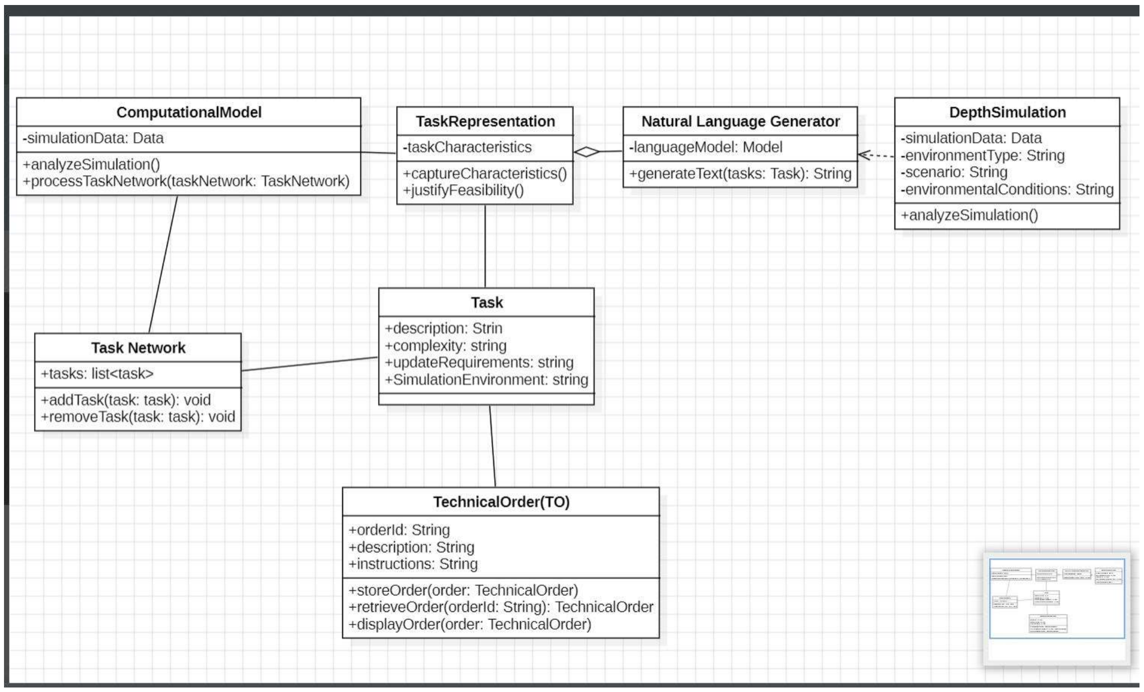


Description:

1. Interact with Simulation: The user interacts with the system to initiate a simulation process. This interaction allows the user to observe simulations and interact with them as needed.
2. Provide Simulation Data: The user provides input simulation data to the system, which serves as the basis for the simulation process. This data could include parameters, settings, or initial conditions required for the simulation.
3. Generate Textual Instructions: The system autonomously generates textual instructions based on the simulation data and other relevant factors. These instructions may include task descriptions, procedures, or guidelines for performing specific actions within the simulated environment.
4. Store Technical Order: The system stores the generated textual instructions as technical orders. This ensures that the instructions are preserved for future reference and can be retrieved when needed.
5. Retrieve and Display Technical Order: The user retrieves stored technical orders from the system as needed. These orders are then displayed to the user for reference or use in carrying out tasks within the simulated environment.
6. Add Task to Task Network: The system allows the user to add tasks to the task network, which represents the sequence or hierarchy of tasks to be performed within the simulation. This enables the user to organize and manage tasks effectively.

7. Remove Task from Task Network: Similarly, the user can remove tasks from the task network when they are no longer needed or have been completed. This helps in keeping the task network up to date and relevant to the current simulation scenario.
8. Handle Input Simulation Data: The user interacts with the system to handle input simulation data, which may involve modifying or updating parameters, settings, or initial conditions for the simulation process.
9. Process Task Networks: The system processes task networks to analyze the sequence or hierarchy of tasks and determine the optimal way to execute them within the simulated environment.
10. Generate Text from Task Networks: Based on the processed task networks, the system generates textual instructions that guide how to perform the tasks effectively and efficiently.
11. Send Generated Text: The system sends the generated textual instructions to the user or other relevant stakeholders for review, feedback, or further action.
12. Validate Text: The system validates the generated textual instructions to ensure accuracy, completeness, and compliance with predefined standards or requirements.
13. Format Text: The system formats the generated textual instructions to enhance readability, clarity, and usability, making them easier to understand and follow.
14. Store Text: Finally, the system stores the formatted textual instructions for future reference or use, ensuring that they are easily accessible and retrievable when needed.

<u>Class Diagram:</u>


ComputationalModel
-simulationData: Data
+analyzeSimulation()
+processTaskNetwork(taskNetwork: TaskNetwork)
TaskRepresentation
-taskCharacteristics
+captureCharacteristics()
+justifyFeasibility()
Natural Language Generator
-languageModel: Model
+generateText(tasks: Task): String
DepthSimulation
-simulationData: Data
-environmentType: String
-scenario: String
-environmentalConditions: String
+analyzeSimulation()
Task
+description: Strin
+complexity: string
+updateRequirements: string
+SimulationEnvironment: string
Task Network
+tasks: list<task>
+addTask(task: task): void
+removeTask(task: task): void
TechnicalOrder(TO)
+orderId: String
+description: String
+instructions: String
+storeOrder(order: TechnicalOrder)
+retrieveOrder(orderId: String): TechnicalOrder
+displayOrder(order: TechnicalOrder)

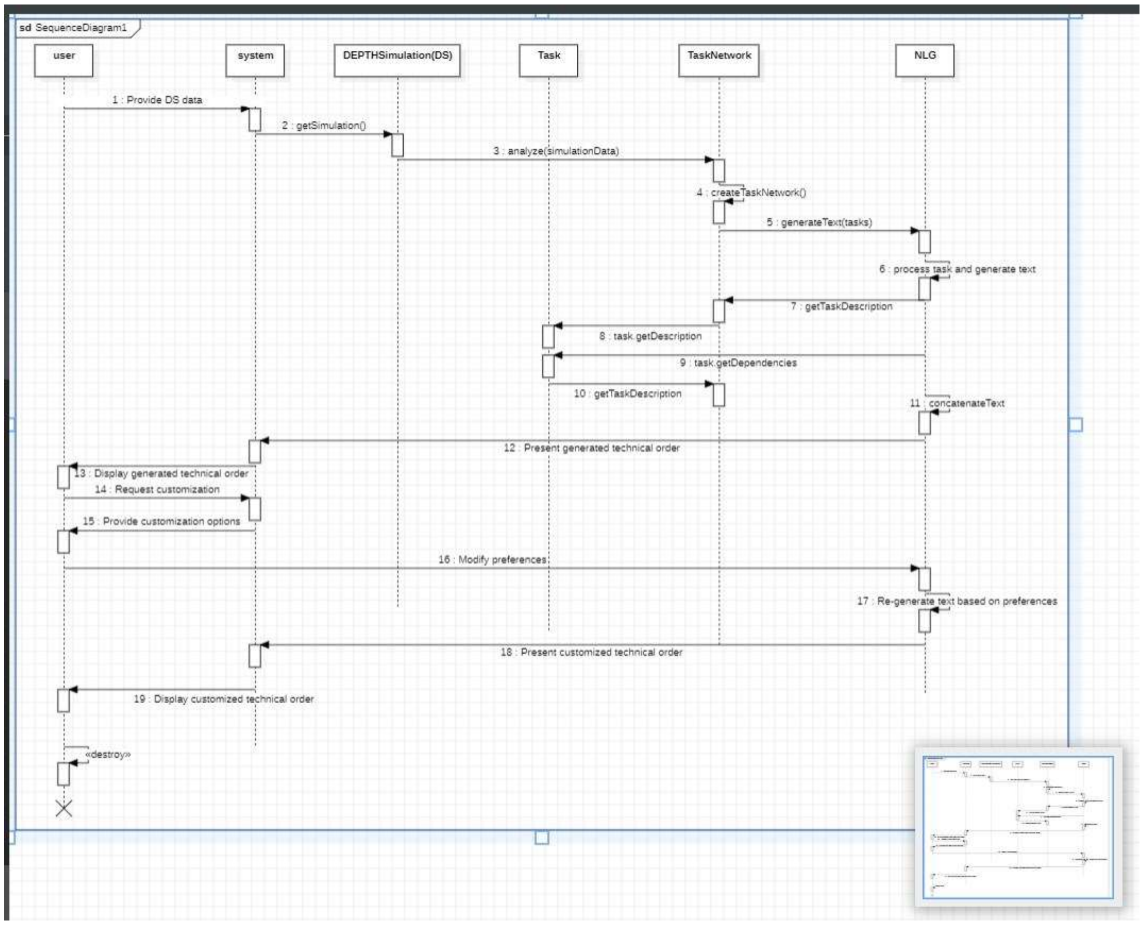
Description:

1. Computational Model Class: The Computational Model class represents the computational model component of the system. It encapsulates simulation data and provides operations such as analyzing simulation data (analyze simulation ()).
2. Task Representation Class: The Task Representation class encapsulates task characteristics within the system. It contains attributes representing task characteristics

and operations such as capturing task characteristics (captureCharacteristics()) and justifying feasibility (justifyFeasibility()).

3. Natural Language Generator Class: The Natural Language Generator class represents the component responsible for generating textual instructions based on task networks. It contains operations such as generating text from tasks (generate text (tasks)).
4. DEPTH Simulation Class: The DEPTH Simulation class represents the component responsible for managing DEPTH simulations within the system. It encapsulates simulation data and provides operations such as analyzing simulations (analyze simulation ()).
5. Task Network Class: The Task Network class represents the structure that captures the sequence or hierarchy of tasks within the system. It contains attributes representing tasks (tasks) and operations such as adding tasks (add Task(task)), removing tasks (removeTask(task)), and retrieving tasks (getTasks()).

Sequence Diagram:

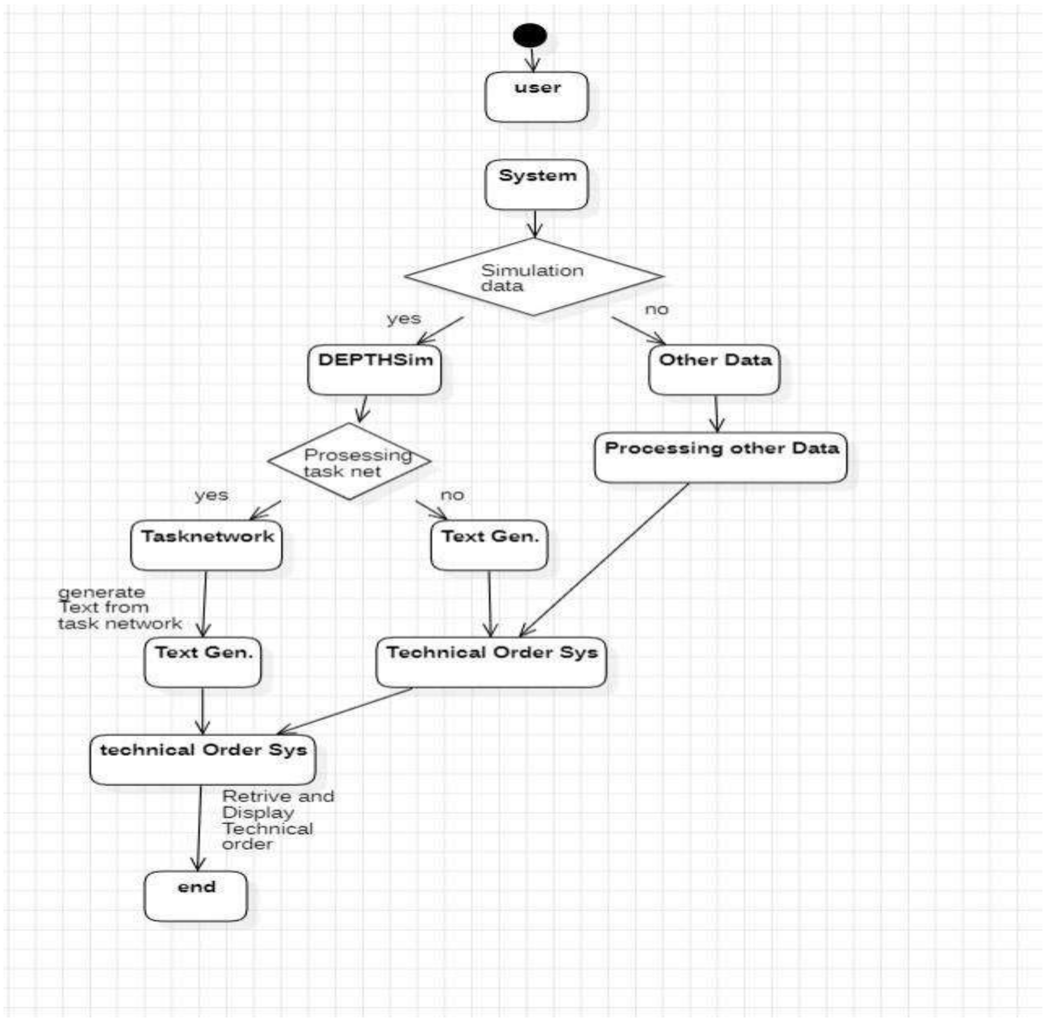

sd SequenceDiagram1
user
system
DEPTHSimulation(DS)
Task
TaskNetwork
NLG
1 : Provide DS data
2 : getSimulation()
3 : analyze(simulationData)
4 : createTaskNetwork()
5 : generateText(tasks)
6 : process task and generate text
7 : getTaskDescription
8 : task.getDescription
9 : task.getDependencies
10 : getTaskDescription
11 : concatenateText
12 : Present generated technical order
13 : Display generated technical order
14 : Request customization
15 : Provide customization options
16 : Modify preferences
17 : Re-generate text based on preferences
18 : Present customized technical order
19 : Display customized technical order
«destroy»

Description:

- The user provides DEPTH simulation data to the system.
- The system retrieves the simulation data from the DEPTHSimulation component and analyzes it.

- The analyzed data is used to create a task network, representing the sequence or hierarchy of tasks within the simulation.
- The TaskNetwork component generates tasks based on the analysis and passes them to the NaturalLanguageGenerator.
- The NaturalLanguageGenerator processes the tasks, generates textual instructions for each task, and compiles them into a technical order.
- The generated technical order is presented to the user by the system for review and reference.
- The user requests customization options from the system.
- The system provides various customization options to the user.
- The user modifies preferences related to the generated technical order.
- The NaturalLanguageGenerator regenerates the text based on the modified preferences.
- The customized technical order is presented to the user by the system for review and reference.

ACTIVITY DIAGRAM:



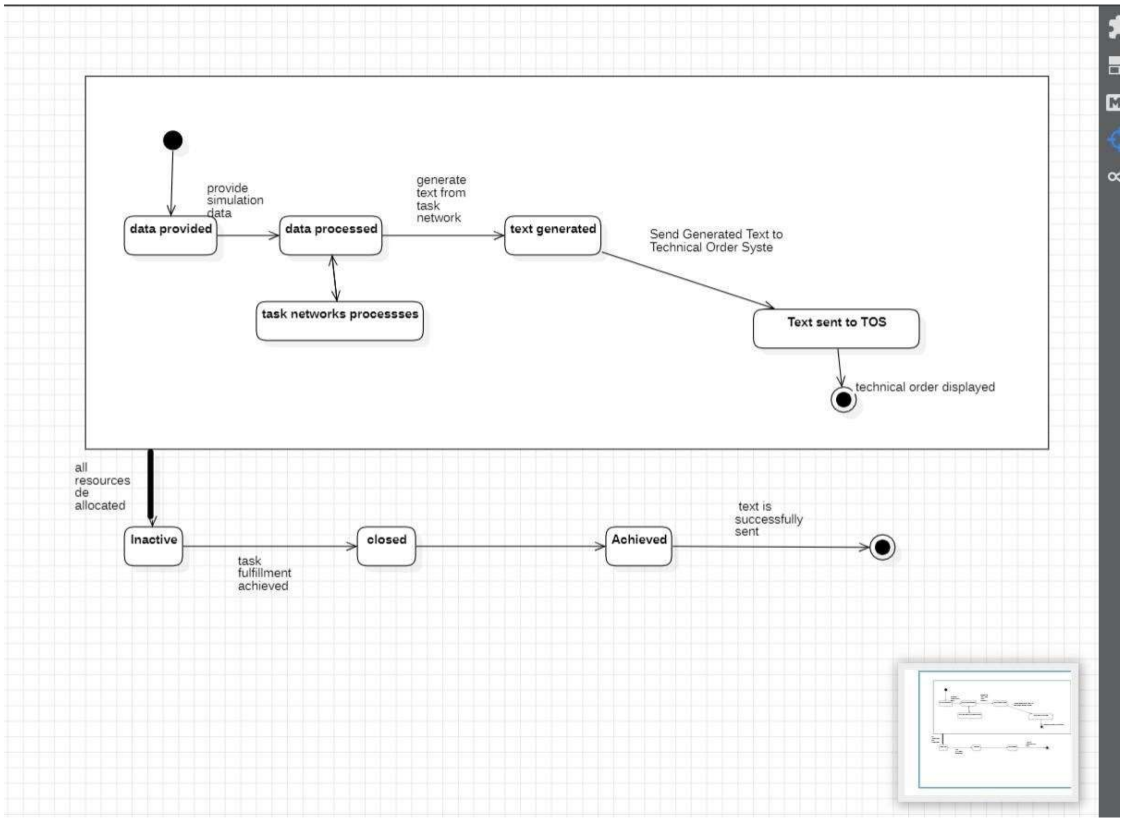
DESCRIPTION:

- User: Initiates the process by providing simulation data.
- System: Receives simulation data from the user.
- Decision: Simulation Data: Determines if the provided data is related to DEPTH Sim

or other data.

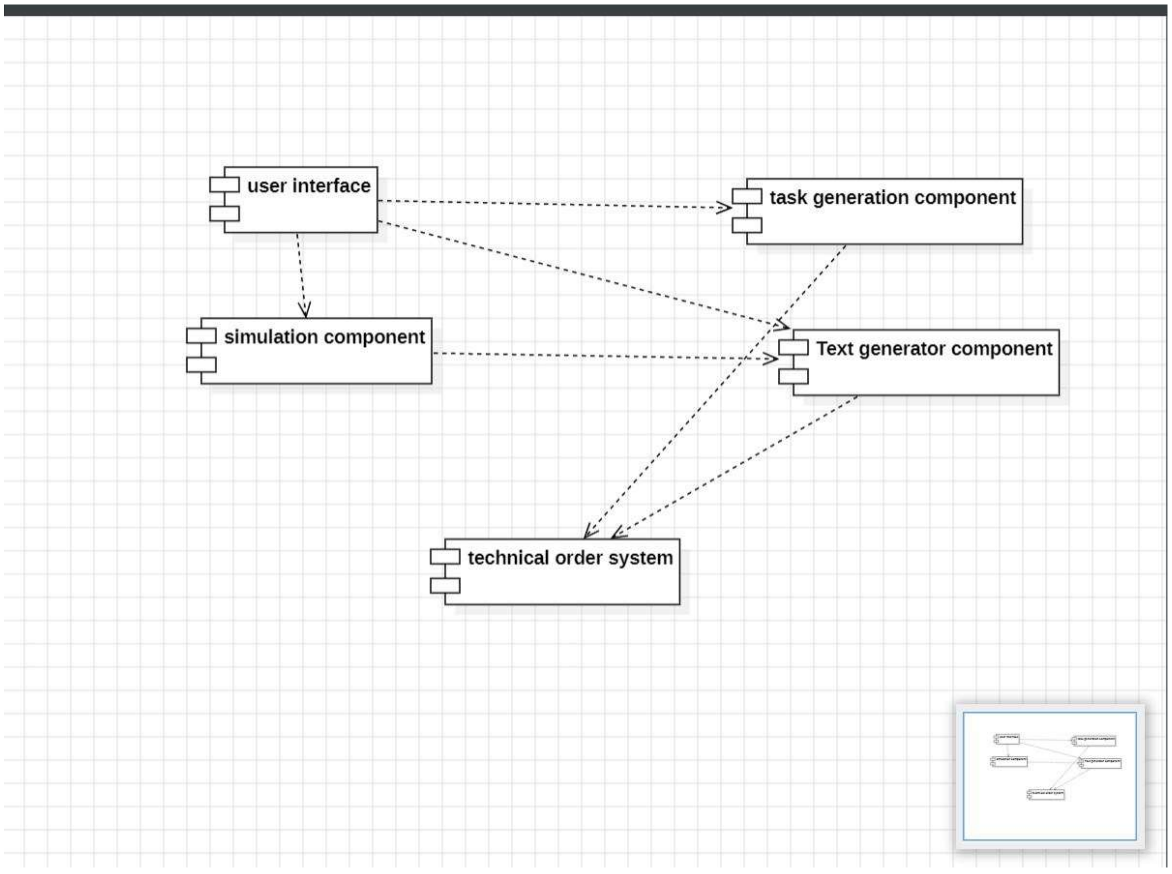
- DEPTH Sim: If the simulation data is related to DEPTH Sim, the system processes it accordingly.
- Other Data: If the simulation data is not related to DEPTH Sim, the system handles it as other data.
- Decision: Processing Task Net: Determines if the system should process task networks.
- Task Network: If task network processing is required, the system processes the task networks.
- Process Task Networks: The system analyzes and processes the task networks.
- Text Gen.: The system generates text based on the processed task networks.
- Technical Order Sys: Manages technical orders and receives the generated text.
- Send Generated Text: The system sends the generated text to the Technical Order System.
- Technical Order Sys: Receives and manages the generated text for technical orders.
- Retrieve and Display Technical Order: The system retrieves and displays the technical order for the user.
- User: Interacts with the displayed technical order.

<u>State Chart Diagram</u>



provide simulation data
data provided
data processed
generate text from task network
text generated
task networks processses
Send Generated Text to Technical Order Syste
Text sent to TOS
technical order displayed
all resources de allocated
Inactive
task fulfillment achieved
closed
Achieved
text is successfully sent

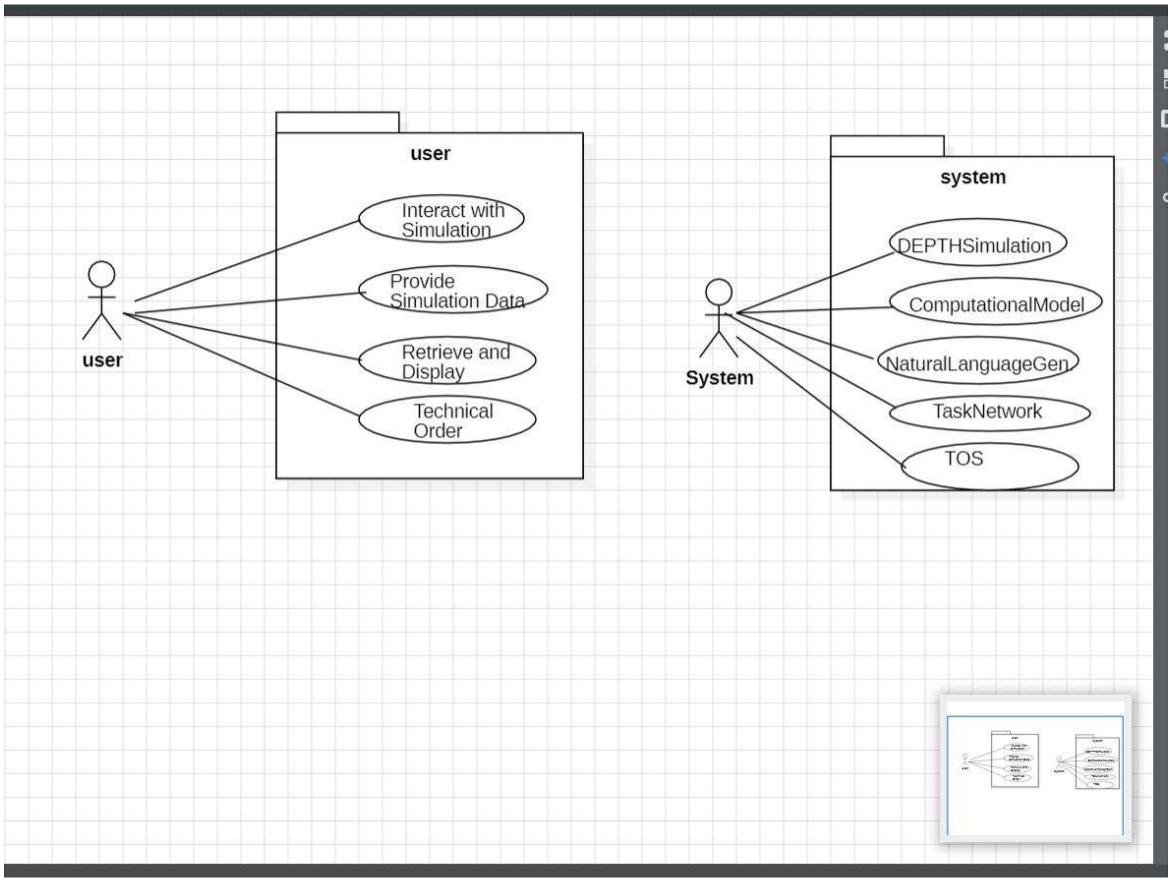
Component Diagram



Description:

- The User Interface component serves as the interface for user interaction, facilitating input of simulation data and output of generated textual instructions.
- The Simulation Component processes simulation data provided by the user, which serves as the input for task generation.
- The Task Generation Component analyzes the simulation data and generates task networks based on predefined algorithms or rules.
- The Text Generator Component takes the generated task networks and converts them into textual instructions or technical orders.
- The Technical Order System component manages the storage and retrieval of technical orders, providing access to previously generated instructions for users.

PACKAGE

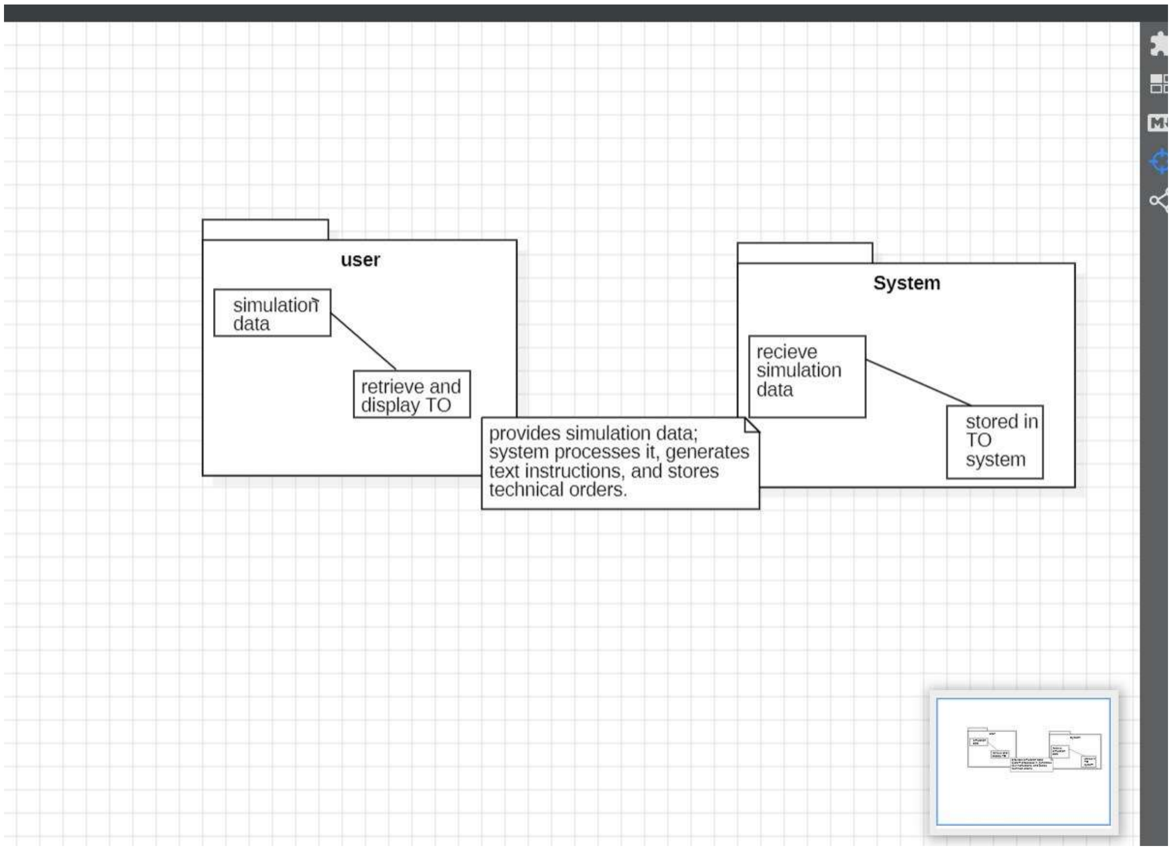


Description:

- User Package:
    - Contains actors representing users interacting with the system.
    - Actors: User
    - Use Cases:
        - Interact with Simulation
        - Provide Simulation Data
        - Retrieve and Display Technical Order
- System Package:
    - Encompasses system components responsible for simulation, data processing, and generating textual instructions.
    - Actors: System
    - Use Cases:
        - DEPTHSimulation

- ComputationalModel
- NaturalLanguageGenerator
- TaskNetwork
- TechnicalOrderSystem

<u>Note</u>

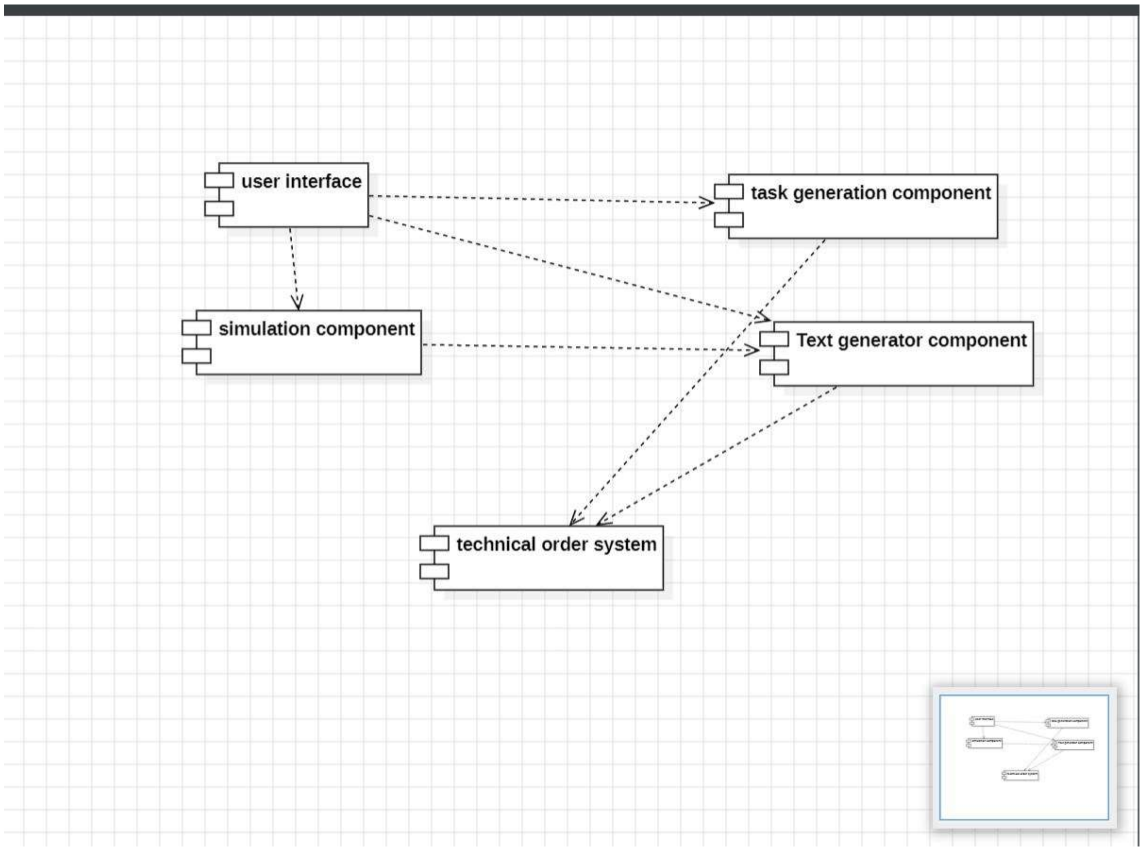

user
simulation data
retrieve and display TO
System
recieve simulation data
stored in TO system
provides simulation data; system processes it, generates text instructions, and stores technical orders.

Description:

- The user interacts with the simulation, providing simulation data.
- The system receives the simulation data and processes it using components like DEPTHSimulation and ComputationalModel.
- Based on the processed data, the system generates textual instructions using the NaturalLanguageGenerator.
- The generated text, representing technical orders, is stored in the TechnicalOrderSystem.
- The user can then retrieve and display the stored technical orders as needed.

- Note from user to system : User provides simulation data; system processes it, generates text instructions, and stores technical orders.

DEPLOYEMENT DIAGRAM



Description:

- The User Interface component is deployed on a client machine where users interact with the system.
- The Simulation Component and Task Generation Component are deployed on a server machine responsible for processing simulation data and generating task networks.
- The Text Generator Component is also deployed on the server machine to generate textual instructions from the task networks.
- The Technical Order System component, responsible for managing technical orders, is deployed on a separate server machine.