FINAL REPORT: MODEL-BASED CONCEPTUAL DESIGN AND INNOVATION

Title: Developing a Card Game Experience to Simulate Value Conflicts in Project Management

Name: Fatema Yaser Abbas Ahmed DHAHI

ID: 82334543

Model Description:

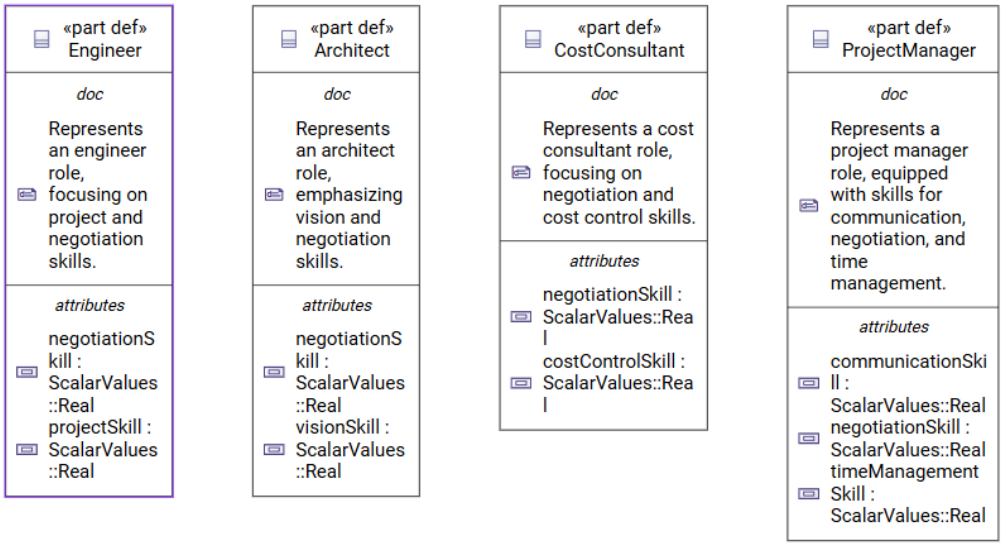
The Card Game model in SysML v2 offers an engaging way for professionals, specifically engineers, architects, project managers, and cost consultants, to explore dynamics they encounter in real-world projects. This game was designed with a primary use case focused on simulating real-world challenges, such as miscommunication, project management, negotiation, and conflicting values. Each role embodies specific professional skills relevant to these scenarios, including negotiation, time management, cost control, and vision. This setup allows players to experience role-specific tasks and communicate through an interaction port system to reflect the collaborative but occasionally conflicting nature of project work. The central objective is to build a "toy house" within defined budget and time constraints, emulating project requirements and constraints common in real-world engineering and construction projects. By representing each role as distinct SysML v2 parts with attributes like negotiationSkill, projectSkill, visionSkill, and costControlSkill, the game emphasizes the specialized contributions and perspectives that different professionals bring to a project, simulating scenarios where teamwork and negotiation are necessary to reach a successful outcome.

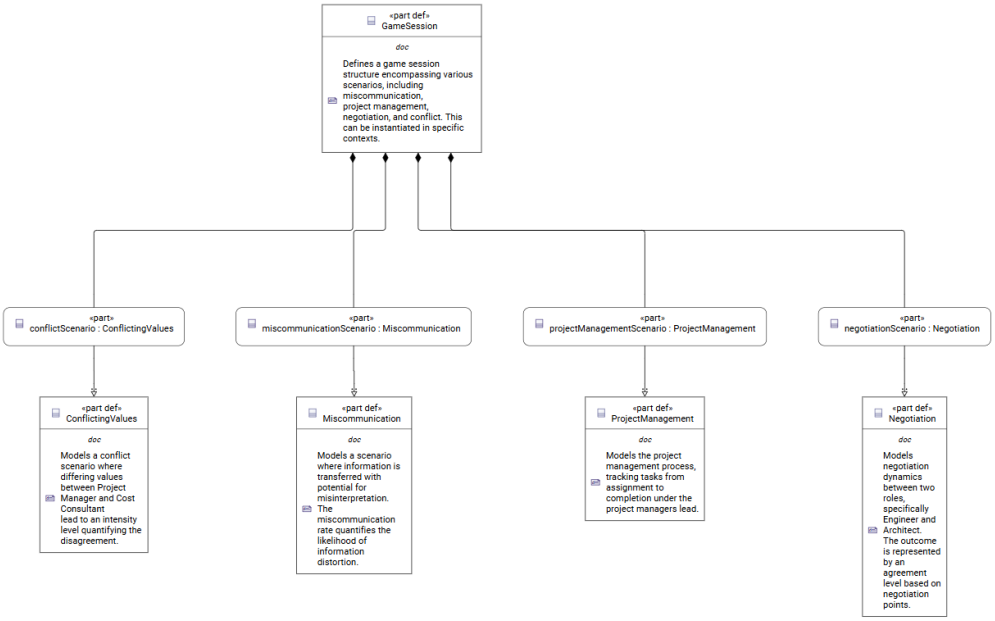
To create a proof of concept (PoC), the model includes a specific scenario where these roles collaborate to build a toy house using predefined materials (wood, plastic, glass) and tools (pin, glue, rope), each with its own cost and constraints. Key requirements are defined around cost management, timeline management, communication, and material/tool selection. For example, the total material and tool costs cannot exceed $10, while the timeline for completion is restricted to 5 units. The PoC showcases a structured approach, validating that each role can interact via an InteractionPort to negotiate material and tool selection and make budget-compliant decisions. The model verifies compliance with these requirements by implementing bind connections between interaction ports for communication, ensuring the toy house is completed within budget and on time. This PoC approach underscores the value of structured communication and role specialization in project-based activities, providing a realistic, gamified simulation of cross-functional project collaboration. Through careful role definition, scenario design, and adherence to strict project requirements, this model demonstrates how SysML v2 can be used to simulate and test complex, collaborative processes.

1) The project chosen was a product which aims to develop a card game experience to simulate value conflicts in project management. The initial assumption was that players representing different roles (engineers, architects, project managers, and cost consultants) will find the game mechanics relatable to their professions in terms of **miscommunication, project management, negotiation and conflicting values**. In the end, there will be a new training tool using gaming in project management. **(Further use cases explained in the last part)**
   *****important parts were highlighted in yellow (Prompts, requirements/ use case, table etc)**

2) Concept Design: Here we developed the abstract view of the product (docs were included in the code).

Notes while doing the model: while ChatGPT was useful, a user cannot rely on it fully as we noticed some issues while using the codes generated such as using the word *perform* which is not used in SysML V2 thus had to be revised to *action*. Also, types for attributes such as Float, String, Binary, etc.. does not exist. All attribute types must be imported from ScalarValues::Real, calarValues::Boolean, etc (in this model numerical values were used). In addition, parts are instantiated inside a part and not a part definition




«part def»
GameSession
doc
Defines a game session structure encompassing various scenarios, including miscommunication, project management, negotiation, and conflict. This can be instantiated in specific contexts.
«part»
conflictScenario : ConflictingValues
«part»
miscommunicationScenario : Miscommunication
«part»
projectManagementScenario : ProjectManagement
«part»
negotiationScenario : Negotiation
«part def»
ConflictingValues
doc
Models a conflict scenario where differing values between Project Manager and Cost Consultant lead to an intensity level quantifying the disagreement.
«part def»
Miscommunication
doc
Models a scenario where information is transferred with potential for misinterpretation. The miscommunication rate quantifies the likelihood of information distortion.
«part def»
ProjectManagement
doc
Models the project management process, tracking tasks from assignment to completion under the project managers lead.
«part def»
Negotiation
doc
Models negotiation dynamics between two roles, specifically Engineer and Architect. The outcome is represented by an agreement level based on negotiation points.

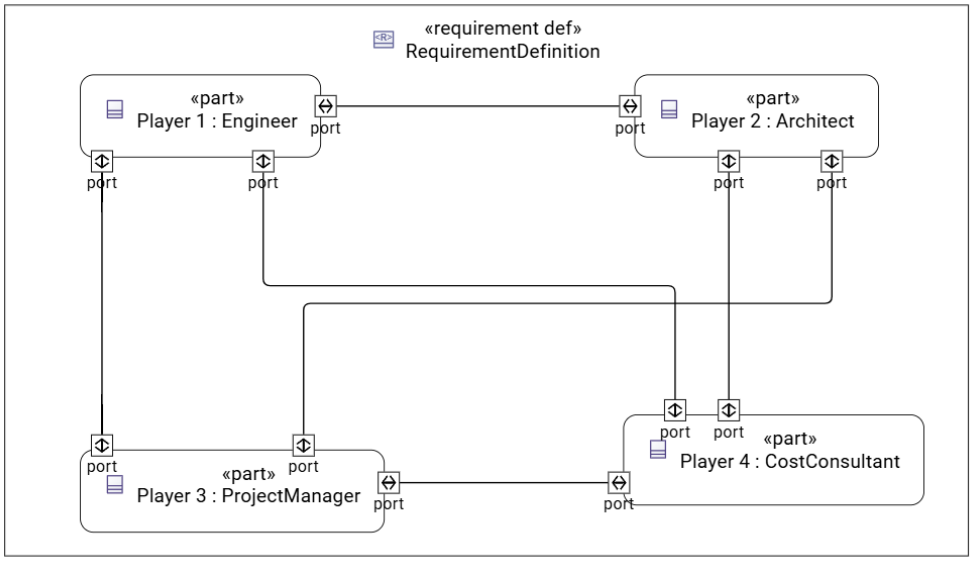
```
package CardGameModel {
  doc /* Models a simulated card game involving various professional roles, each with specific skills.
       This model also defines scenarios to illustrate game mechanics such as miscommunication and negotiation. */

  // Role Definitions
  part def Engineer {
    doc /* Represents an engineer role, focusing on project and negotiation skills. */
    attribute negotiationSkill: ScalarValues::Real;
    attribute projectSkill: ScalarValues::Real;
  }

  part def Architect {
    doc /* Represents an architect role, emphasizing vision and negotiation skills. */
    attribute negotiationSkill: ScalarValues::Real;
    attribute visionSkill: ScalarValues::Real;
  }

  part def ProjectManager {
    doc /* Represents a project manager role, equipped with skills for communication, negotiation, and time management. */
    attribute communicationSkill: ScalarValues::Real;
    attribute negotiationSkill: ScalarValues::Real;
    attribute timeManagementSkill: ScalarValues::Real;
  }
```

```
  part def CostConsultant {
    doc /* Represents a cost consultant role, focusing on negotiation and cost control skills. */
    attribute negotiationSkill: ScalarValues::Real;
    attribute costControlSkill: ScalarValues::Real;
  }

  // Game Mechanics
  part def Miscommunication {
    doc /* Models a scenario where information is transferred with potential for misinterpretation.
          The miscommunication rate quantifies the likelihood of information distortion. */
    in ref fromRole: ProjectManager;
    out ref toRole: Engineer;
    in msgContent: String;
    out misinterpretationRate: ScalarValues::Real;
  }

  part def ProjectManagement {
    doc /* Models the project management process, tracking tasks from assignment to completion under the project manager's lead. */
    in ref leader: ProjectManager;
    in projectTasks: ScalarValues::Natural;
    out completedTasks: ScalarValues::Natural;
    out projectProgress: ScalarValues::Real;
  }
```

```
  part def Negotiation {
    doc /* Models negotiation dynamics between two roles, specifically Engineer and Architect.
        The outcome is represented by an agreement level based on negotiation points. */
    in ref roleA: Engineer;
    in ref roleB: Architect;
    in negotiationPoints: ScalarValues::Natural;
    out agreementLevel: ScalarValues::Real;
  }

  part def ConflictingValues {
    doc /* Models a conflict scenario where differing values between Project Manager and Cost Consultant
        lead to an intensity level quantifying the disagreement. */
    in ref role1: ProjectManager;
    in ref role2: CostConsultant;
    out conflictIntensity: ScalarValues::Real;
  }

  // Game Session Definition
  part def GameSession {
    doc /* Defines a game session structure encompassing various scenarios, including miscommunication,
        project management, negotiation, and conflict. This can be instantiated in specific contexts. */
    part miscommunicationScenario: Miscommunication;
    part projectManagementScenario: ProjectManagement;
```

```
    part negotiationScenario: Negotiation;
    part conflictScenario: ConflictingValues;
  }
}
```

3) Later the type of connection was defined, in this model binding connection was chosen because two elements share or "bind" to the same data or resources and this element behaves as one. Connections are usually established to understand relationships better in this working progress as things develop. It is indeed easier to navigate and document when we have multiple views as well, and this can be reflected when defining relationships in an interconnection view.



4) PoC: a new package was defined where the Proof of Concept (PoC) simulates a **single toy house construction scenario** where the Engineer, Architect, CostConsultant, and ProjectManager collaborate. It demonstrates material and tool selection, calculates the construction cost, and checks for adherence to budget and time constraints (One was for a negotiation scenario and the other was for Miscommunication).

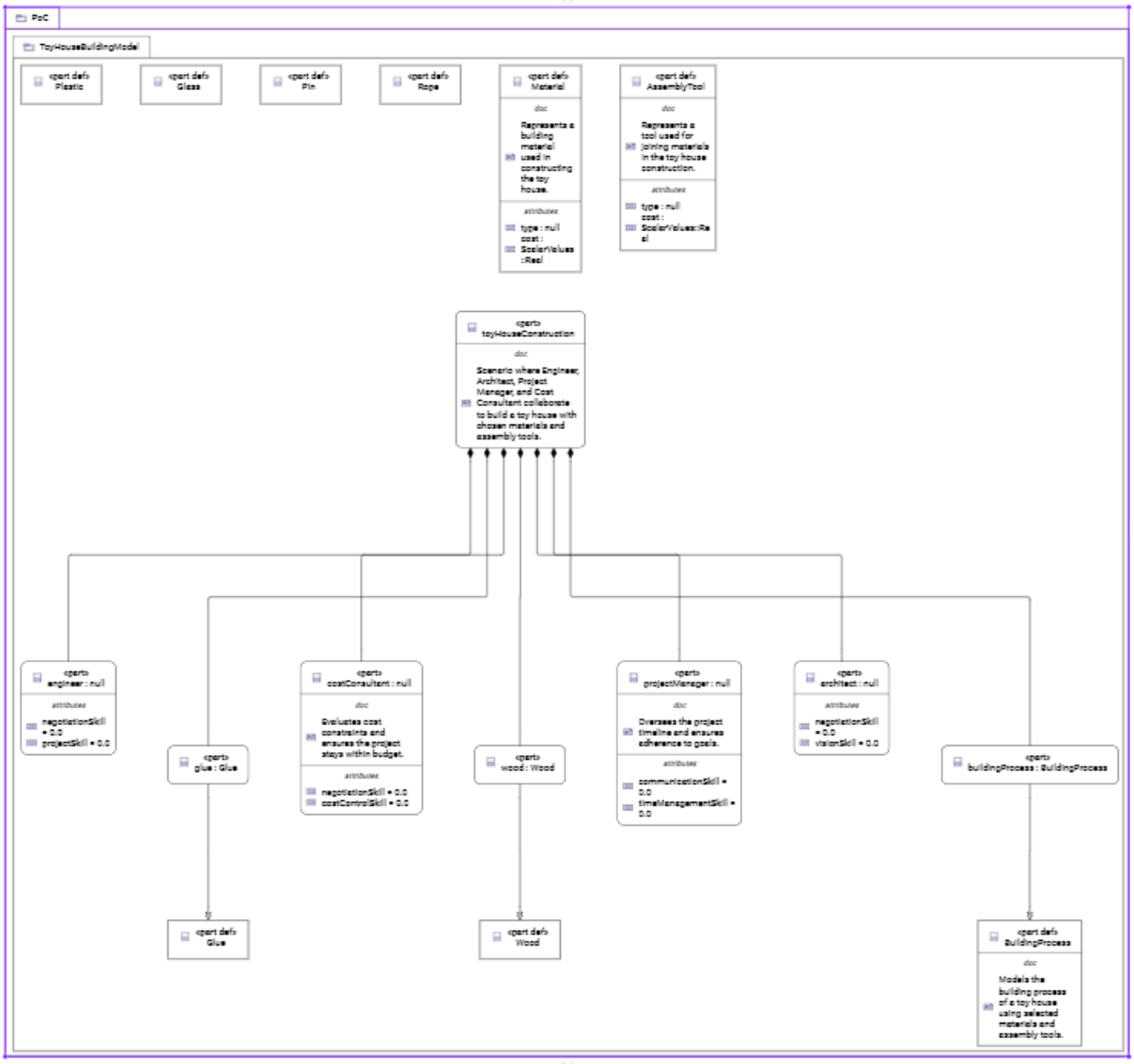
```
package ToyHouseBuildingModel {
```

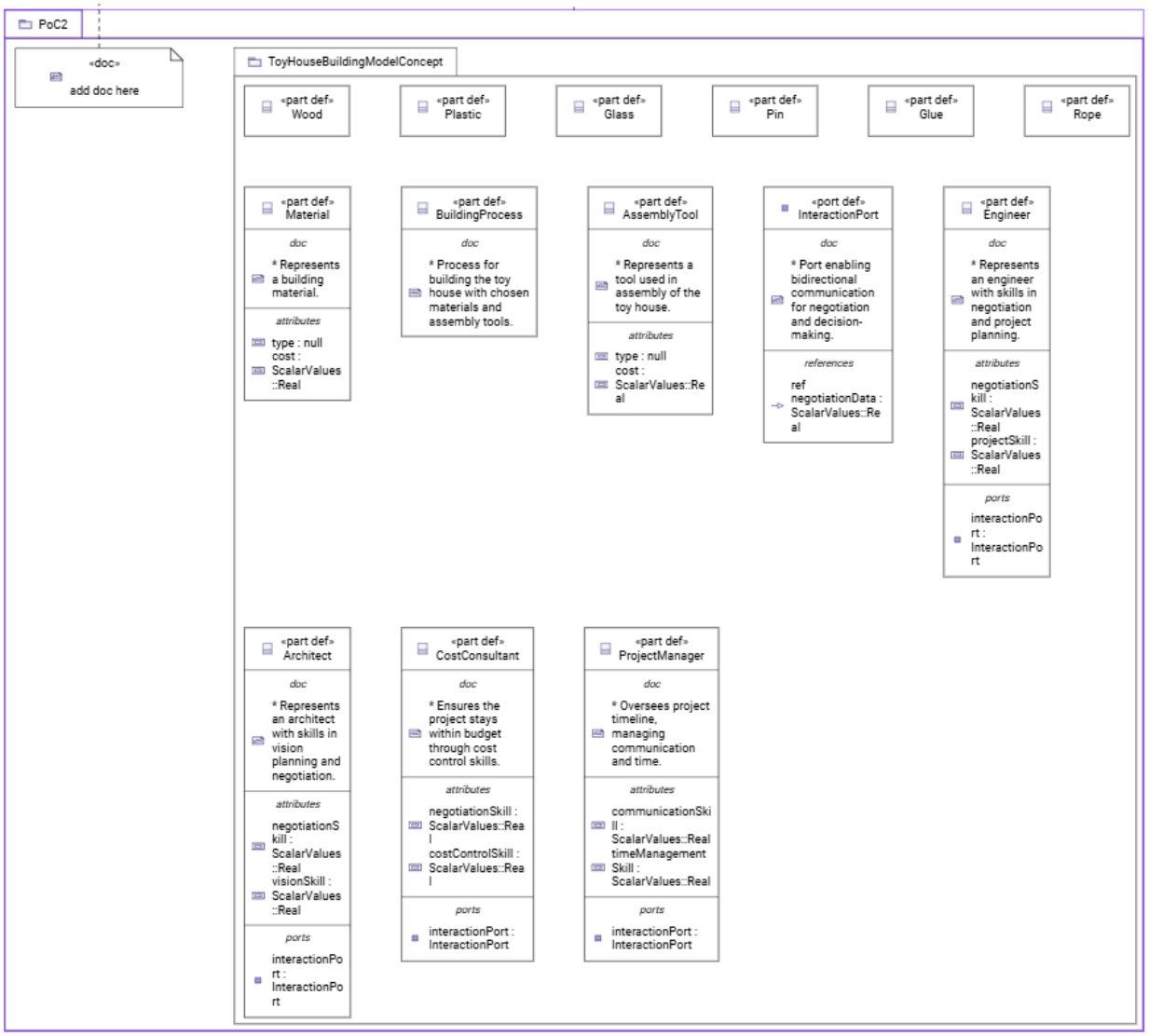
```
import CardGameModel::*;

// Material Definitions
part def Material {
    doc /* Represents a building material used in constructing the toy house. */
    attribute type: String;
    attribute cost: ScalarValues::Real;
}

part def Wood : Material {
    attribute type = "Wood";
    attribute cost = 5.0;
}

part def Plastic : Material {
    attribute type = "Plastic";
    attribute cost = 3.0;
}

part def Glass : Material {
    attribute type = "Glass";
    attribute cost = 8.0;
}

// Assembly Tool Definitions
part def AssemblyTool {
    doc /* Represents a tool used for joining materials in the toy house construction.
*/
    attribute type: String;
    attribute cost: ScalarValues::Real;
}

part def Pin : AssemblyTool {
    attribute type = "Pin";
    attribute cost = 1.0;
}

part def Glue : AssemblyTool {
```

```
    attribute type = "Glue";
    attribute cost = 2.0;
  }

  part def Rope : AssemblyTool {
    attribute type = "Rope";
    attribute cost = 1.5;
  }

  // Building Process Definition
  part def BuildingProcess {
    doc /* Models the building process of a toy house using selected materials and
assembly tools. */

    in ref primaryMaterial: Material;
    in ref secondaryMaterial: AssemblyTool;
    out completedToyHouse: String;
    out totalCost: ScalarValues::Real;

    behavior buildToyHouse {
      doc /* Combines materials and tools to create a toy house with cost
calculation. */

      perform construct {
        doc /* Constructs the toy house using specified materials and tools,
calculating total cost. */

        ref matType = primaryMaterial.type;
        ref toolType = secondaryMaterial.type;
        ref matCost = primaryMaterial.cost;
        ref toolCost = secondaryMaterial.cost;

        completedToyHouse = "Toy house built with " + matType + " and assembled
using " + toolType;
        totalCost = matCost + toolCost;
      }
    }
  }
```

```
  // Toy House Construction Scenario
  part toyHouseConstruction {
    doc /* Scenario where Engineer, Architect, Project Manager, and Cost
Consultant collaborate
         to build a toy house with chosen materials and assembly tools. */

    part engineer: Engineer {
      attribute negotiationSkill = 0.8;
      attribute projectSkill = 0.7;
    }

    part architect: Architect {
      attribute negotiationSkill = 0.7;
      attribute visionSkill = 0.9;
    }

    part projectManager: ProjectManager {
      doc /* Oversees the project timeline and ensures adherence to goals. */
      attribute communicationSkill = 0.85;
      attribute timeManagementSkill = 0.9;
    }

    part costConsultant: CostConsultant {
      doc /* Evaluates cost constraints and ensures the project stays within budget.
*/
      attribute negotiationSkill = 0.75;
      attribute costControlSkill = 0.9;
    }

    part buildingProcess: BuildingProcess;

    // Materials and Tools Selection
    part wood: Wood;
    part glue: Glue;

    // Assign selected materials to the building process
    bind wood to buildingProcess.primaryMaterial;
```

```
    bind glue to buildingProcess.secondaryMaterial;

    // Role Communication - Interaction Binding
    bind engineer.interactionPort to architect.interactionPort;
    bind projectManager.interactionPort to costConsultant.interactionPort;
    bind costConsultant.interactionPort to engineer.interactionPort;

    // Building Process Execution
    behavior toyHouseScenario {
        perform buildingProcess.buildToyHouse.construct;
    }

    // Cost and Timeline Monitoring by Project Manager and Cost Consultant
    attribute maxBudget = 10.0;
    attribute timeLimit = 5.0; // Represents a time limit for the project in arbitrary
units

    assert constraint checkCost {
        doc /* Ensures the total cost of materials and tools is within the project
budget. */
        buildingProcess.totalCost <= maxBudget
    }

    assert constraint checkTimeline {
        doc /* Checks that the project can be completed within the specified time
limit. */
        projectManager.timeManagementSkill * timeLimit >= 5.0
    }
  }
}
```


PoC
ToyHouseBuildingModel
«part def» Plastic
«part def» Glass
«part def» Pin
«part def» Rope
«part def» Material
doc
Represents a building material used in constructing the toy house.
attributes
type : null
cost : ScalarValues::Real
«part def» AssemblyTool
doc
Represents a tool used for joining materials in the toy house construction.
attributes
type : null
cost : ScalarValues::Real
«part» toyHouseConstruction
doc
Scenario where Engineer, Architect, Project Manager, and Cost Consultant collaborate to build a toy house with chosen materials and assembly tools.
«part» engineer : null
attributes
negotiationSkill = 0.0
projectSkill = 0.0
«part» glue : Glue
«part» costConsultant : null
doc
Evaluates cost constraints and ensures the project stays within budget.
attributes
negotiationSkill = 0.0
costControlSkill = 0.0
«part» wood : Wood
«part» projectManager : null
doc
Oversees the project timeline and ensures adherence to goals.
attributes
communicationSkill = 0.0
timeManagementSkill = 0.0
«part» architect : null
attributes
negotiationSkill = 0.0
visionSkill = 0.0
«part» buildingProcess : BuildingProcess
«part def» Glue
«part def» Wood
«part def» BuildingProcess
doc
Models the building process of a toy house using selected materials and assembly tools.


PoC2
«doc»
add doc here
ToyHouseBuildingModelConcept
«part def» Wood
«part def» Plastic
«part def» Glass
«part def» Pin
«part def» Glue
«part def» Rope
«part def» Material
doc
* Represents a building material.
attributes
type : null
cost : ScalarValues::Real
«part def» BuildingProcess
doc
* Process for building the toy house with chosen materials and assembly tools.
«part def» AssemblyTool
doc
* Represents a tool used in assembly of the toy house.
attributes
type : null
cost : ScalarValues::Real
«port def» InteractionPort
doc
* Port enabling bidirectional communication for negotiation and decision-making.
references
ref negotiationData : ScalarValues::Real
«part def» Engineer
doc
* Represents an engineer with skills in negotiation and project planning.
attributes
negotiationSkill : ScalarValues::Real
projectSkill : ScalarValues::Real
ports
interactionPort : InteractionPort
«part def» Architect
doc
* Represents an architect with skills in vision planning and negotiation.
attributes
negotiationSkill : ScalarValues::Real
visionSkill : ScalarValues::Real
ports
interactionPort : InteractionPort
«part def» CostConsultant
doc
* Ensures the project stays within budget through cost control skills.
attributes
negotiationSkill : ScalarValues::Real
costControlSkill : ScalarValues::Real
ports
interactionPort : InteractionPort
«part def» ProjectManager
doc
* Oversees project timeline, managing communication and time.
attributes
communicationSkill : ScalarValues::Real
timeManagementSkill : ScalarValues::Real
ports
interactionPort : InteractionPort

5) **Then and using CHATGPT SYSML V2 a table was created to shows which elements in the initial concept model were replaced in the PoC**

Element in Initial Concept Model	Replacement or Expansion in PoC Model	Description
`Engineer`, `Architect`	Same roles, with added specific attributes for negotiation	Both roles are reused in the PoC with predefined skills to simulate negotiation and decision-making in material/tool selection.
`ProjectManager`	Added role with time management and communication monitoring	Role expanded in PoC to oversee project timeline (`timeLimit`) and ensure goals are met within the set limits.
`CostConsultant`	Added role to evaluate material and tool costs	Expanded in PoC to include `costControlSkill`, managing budget constraints for the toy house project.
`Miscommunication`, `Negotiation`, `ConflictingValues`	Replaced by `BuildingProcess` behavior and constraints	Simplified PoC focused on building rather than communication dynamics, using cost and time constraints to guide project outcomes.
`InteractionPort`	Kept with bindings among all roles	Used in PoC to enable communication and negotiation between `Engineer`, `Architect`, `ProjectManager`, and `CostConsultant`.
`gameSession`	Replaced with `toyHouseConstruction` scenario	Specific construction scenario created to focus on toy house building with real-time material and tool selection.
`Material` and `AssemblyTool`	New parts (`Wood`, `Plastic`, `Glass`, `Pin`, `Glue`, `Rope`)	Material and tool parts created for realistic selection and cost tracking in the PoC.
`BuildingProcess`	Newly defined with `buildToyHouse` behavior	Introduced in PoC to simulate toy house construction, calculating `completedToyHouse` and `totalCost`.
Cost and Timeline Constraints	Added `checkCost` and `checkTimeline` constraints	New constraints ensure that construction stays within budget and meets the timeline, monitored by `CostConsultant` and `ProjectManager`.

6) A version of the table as a comment was created to be a package containing the entire PoC model (The code was longer, but was updated for the doc element to match the following comments " /* * This is documentation of the owning * package. * /" and only the specified part was used

/* * This is documentation of the owning * package. * Below is a summary of elements from the initial concept model that were replaced, expanded, or added. * * ---------- ---------- * | Element in Initial Concept Model | Replacement or Expansion in PoC Model | Description | * ---------- * | Engineer, Architect | Same roles, with added specific attributes for negotiation | Both roles are reused in the PoC with predefined skills to simulate negotiation and decision-making. | * | ProjectManager | Added role with time management and communication monitoring | Expanded in PoC to oversee project timeline (`timeLimit`) and ensure goals are met within limits. | * | CostConsultant | Added role to evaluate material and tool costs | Expanded in PoC to include `costControlSkill`, managing budget constraints for the toy house project.| * | Miscommunication, Negotiation, | Replaced by BuildingProcess behavior and constraints | Simplified PoC focused on building, using cost and time constraints to guide outcomes. | * | ConflictingValues | | | * | InteractionPort | Kept with bindings among all roles | Used in PoC to enable communication and negotiation among Engineer, Architect, ProjectManager, | * | | | and CostConsultant. | * | gameSession | Replaced with toyHouseConstruction scenario | Specific scenario to focus on toy house building with real-time material and tool selection. | * | Material and AssemblyTool | New parts (Wood, Plastic, Glass, Pin, Glue, Rope) | Created for realistic selection and cost tracking in the PoC. | * | BuildingProcess | Newly defined with buildToyHouse behavior | Simulates toy house construction, calculating `completedToyHouse` and `totalCost`. | * | Cost and Timeline Constraints | Added checkCost and checkTimeline constraints | Ensure project stays within budget and meets the timeline, monitored by CostConsultant and | * | | | ProjectManager. | * ---------- */

7) **Summary**

Since the free version of ChatGPT was used the package was limited and the model was updated and may had some changes, but the final version mainly covers the following (Note that PoC was developed according to a scenario for the card game):

Prompts

- **Prompt**: Create a SysML v2 model with roles representing engineers, architects, project managers, and cost consultants. The game mechanics should include scenarios for miscommunication, project management, negotiation, and conflicting values.

- **Prompt**: Add more `doc` elements and detailed descriptions to the SysML v2 code to enhance clarity for each model element.

- **Prompt**: Should the `gameSession` also be a part definition?

- **Prompt**: Suggest a simple interconnection view for `Engineer`, `Architect`, `ProjectManager`, and `CostConsultant` using inout ports.

- **Prompt**: Provide a Proof of Concept (PoC) model where the scenario is building a toy house with materials (wood, plastic, glass) and tools (pin, glue, rope).

- **Prompt**: Generate a table showing which elements in the initial concept model were replaced, expanded, or added in the PoC.

- **Prompt**: Convert the table to a `doc` code element for SysML v2.

- **Prompt**: Update the `doc` elements to match the following format: `"/* This is documentation of the owning package. */"`

- **Prompt**: Can you also apply the specified `doc` format to the previous PoC model code?

- **Prompt**: Provide all the **prompts, assumptions, PoC, requirements**, and the **concept model** for this model.

- **Prompt**: Create a version of the table as a comment in a package containing the entire PoC model.

- **Prompt**: Update the `doc` elements in the last provided code to match the specified format: `"/* This is documentation of the owning package. */"`

- **Prompt**: Extract a list of all prompts given since the assumption statement provided.

Assumptions

1. **Assumption**: Players representing different roles (Engineer, Architect, Project Manager, and Cost Consultant) will find the game mechanics relatable to their professions, reflecting aspects like miscommunication, project management, negotiation, and conflicting values.

2. **Assumption**: **Material Types and Costs**:
    - Available materials are limited to wood, plastic, and glass, each with predefined costs: wood ($5), plastic ($3), and glass ($8).
    - Assembly tools include pin, glue, and rope, each with a specific cost.
3. **Assumption**: **Role Responsibilities**:
    - Engineer and Architect collaborate on material selection based on skills such as negotiation and project skills.
    - CostConsultant ensures that the project remains within budget, using a costControlSkill attribute to evaluate total costs.
    - ProjectManager oversees the timeline and ensures project completion within a defined time limit.
4. **Assumption**: **Project Constraints for Toy House Construction**:
    - The maximum budget for the project is $10.
    - The project must be completed within a time limit of 5 units.
5. **Assumption**: **Interaction Method**:
    - Each role uses an InteractionPort to communicate with others, facilitating negotiation and decision-making on material and tool selection.

Requirements

1. **Requirement**: **Cost Management Requirement**:
    - *Requirement*: The total cost of materials and tools must not exceed $10.
    - *Stakeholders*: CostConsultant, Engineer, Architect
    - *Verification*: Calculated during the buildToyHouse process and validated by CostConsultant.
2. **Requirement**: **Timeline Management Requirement**:
    - *Requirement*: The toy house must be built within a specified time frame, with a time limit of 5 units.
    - *Stakeholders*: ProjectManager

- *Verification*: Managed by ProjectManager using timeManagementSkill to ensure timeline compliance.

3. **Requirement**: **Interaction Requirement**:
   - *Requirement*: All roles must communicate and negotiate material/tool selection through an InteractionPort.
   - *Stakeholders*: Engineer, Architect, ProjectManager, CostConsultant
   - *Verification*: Interaction is modeled through bind connections between roles' InteractionPorts.
4. **Requirement**: **Material and Tool Selection Requirement**:
   - *Requirement*: The project should use one primary material and one assembly tool selected collaboratively by Engineer and Architect.
   - *Stakeholders*: Engineer, Architect
   - *Verification*: Material and tool assignment in the toyHouseConstruction scenario demonstrates compliance.

3. Proof of Concept (PoC)

Objective:
The PoC simulates a **single toy house construction scenario** where the `Engineer`, `Architect`, `CostConsultant`, and `ProjectManager` collaborate. It demonstrates material and tool selection, calculates the construction cost, and checks for adherence to budget and time constraints.

Structure:
- **Roles**: Each role (Engineer, Architect, ProjectManager, CostConsultant) has predefined skills relevant to their functions in the project.
- **Materials and Tools**: The scenario selects wood and glue as primary material and assembly tool, respectively.
- **Building Process**: The process calculates the total cost and constructs a description of the completed toy house.
- **Constraints**:
  - `checkCost`: Ensures total project cost is within the $10 budget.
  - `checkTimeline`: Ensures the project completion falls within the time limit, based on the project manager's skills.

---

4. Requirements

1. **Cost Management Requirement**:
   - *Requirement*: The total cost of materials and tools must not exceed $10.
   - *Stakeholders*: `CostConsultant`, `Engineer`, `Architect`
   - *Verification*: Calculated during the `buildToyHouse` process and validated by `CostConsultant`.

2. **Timeline Management Requirement**:
   - *Requirement*: The toy house must be built within a specified time frame, with a time limit of 5 units.
   - *Stakeholders*: `ProjectManager`
   - *Verification*: Managed by `ProjectManager` using `timeManagementSkill` to ensure timeline compliance.

3. **Interaction Requirement**:
   - *Requirement*: All roles must communicate and negotiate material/tool selection through an `InteractionPort`.
   - *Stakeholders*: `Engineer`, `Architect`, `ProjectManager`, `CostConsultant`
   - *Verification*: Interaction is modeled through `bind` connections between roles' `InteractionPort`s.

4. **Material and Tool Selection Requirement**:
   - *Requirement*: The project should use one primary material and one assembly tool selected collaboratively by `Engineer` and `Architect`.
   - *Stakeholders*: `Engineer`, `Architect`
   - *Verification*: Material and tool assignment in the `toyHouseConstruction` scenario demonstrates compliance.

---

5. Concept Model

Here's the SysML v2 code representing the **concept model** of the `ToyHouseBuildingModel`, illustrating core roles, materials, tools, and building process:

```sysml-v2
package ToyHouseBuildingModelConcept {

    // Role Definitions
    part def Engineer {
        doc /*
        * Represents an engineer with skills in negotiation and project planning.
        */
        attribute negotiationSkill: ScalarValues::Real;
        attribute projectSkill: ScalarValues::Real;
        port interactionPort: InteractionPort;
    }

    part def Architect {
```

```
    doc /*
    * Represents an architect with skills in vision planning and negotiation.
    */
    attribute negotiationSkill: ScalarValues::Real;
    attribute visionSkill: ScalarValues::Real;
    port interactionPort: InteractionPort;
}

part def ProjectManager {
    doc /*
    * Oversees project timeline, managing communication and time.
    */
    attribute communicationSkill: ScalarValues::Real;
    attribute timeManagementSkill: ScalarValues::Real;
    port interactionPort: InteractionPort;
}

part def CostConsultant {
    doc /*
    * Ensures the project stays within budget through cost control skills.
    */
    attribute negotiationSkill: ScalarValues::Real;
    attribute costControlSkill: ScalarValues::Real;
    port interactionPort: InteractionPort;
}

// Material and Tool Definitions
part def Material {
    doc /*
    * Represents a building material.
    */
    attribute type: String;
    attribute cost: ScalarValues::Real;
}

part def Wood : Material {
    attribute type = "Wood";
    attribute cost = 5.0;
}

part def Plastic : Material {
    attribute type = "Plastic";
    attribute cost = 3.0;
}

part def Glass : Material {
    attribute type = "Glass";
    attribute cost = 8.0;
```

```
    }

    part def AssemblyTool {
        doc /*
        * Represents a tool used in assembly of the toy house.
        */
        attribute type: String;
        attribute cost: ScalarValues::Real;
    }

    part def Pin : AssemblyTool {
        attribute type = "Pin";
        attribute cost = 1.0;
    }

    part def Glue : AssemblyTool {
        attribute type = "Glue";
        attribute cost = 2.0;
    }

    part def Rope : AssemblyTool {
        attribute type = "Rope";
        attribute cost = 1.5;
    }

    // Building Process
    part def BuildingProcess {
        doc /*
        * Process for building the toy house with chosen materials and assembly tools.
        */
        in ref primaryMaterial: Material;
        in ref secondaryMaterial: AssemblyTool;
        out completedToyHouse: String;
        out totalCost: ScalarValues::Real;

        behavior buildToyHouse {
            doc /*
            * Builds the toy house with selected materials and calculates total cost.
            */
            perform construct {
                ref matType = primaryMaterial.type;
                ref toolType = secondaryMaterial.type;
                ref matCost = primaryMaterial.cost;
                ref toolCost = secondaryMaterial.cost;

                completedToyHouse = "Toy house built with " + matType + " and assembled using " +
toolType;
                totalCost = matCost + toolCost;
```

```
        }
    }
  }

  // Interaction Port Definition
  port def InteractionPort {
    doc /*
    * Port enabling bidirectional communication for negotiation and decision-making.
    */
    inout message: String;
    inout negotiationData: ScalarValues::Real;
  }
}
```

Summary
- The **concept model** defines all roles, materials, tools, the building process, and an `InteractionPort` for role communication.
- **Requirements** focus on budget, timeline, communication, and material/tool selection.
- The **PoC** scenario validates collaboration, ensuring constraints on cost and time are respected, while allowing roles to make decisions and perform the toy house construction.

This structure provides a comprehensive foundation for testing and iterating on the `ToyHouseBuildingModel`.