现在这两个机器人已经物理存在，我的目标是在软件层面上实现它们的数字化和交互。为此，我需要为每个机器人构建一个 AAS（Asset Administration Shell），即它们的数字孪生。

URDF 文件（基于 XML）是用于描述机器人的几何结构和运动学特性，我将基于这个文件开始我的项目，最终实现Unity可视化建模和仿真。

同时，我将使用 IDTA 定义的 Submodels 来构建 AAS 的各个功能模块，打包并标准化描述机器人的状态、位置、技能、通信能力等信息。这些信息基于urdf文件，为此我需要理解xml，并实现跨文件的通讯。

最终，通过 Unity，我们将 URDF 的几何信息和 Submodel 中的实时数据结合，实现一个可视化、可交互的 3D 数字孪生系统。，，我的任务是实现机器人从 URDF 到 Unity 中的数字孪生建模，并通过 AAS + Submodels 让两个机器人在软件层面实现数据交互。

我的任务分配是

实现机器人从 URDF 到 Unity 中的数字孪生建模，并通过 AAS + Submodels 让两个机器人在软件层面实现数据交互。

 **解析 URDF 文件**  
→ 提取机器人静态结构和参数，构建 Submodels 的“基础信息”部分。

 **定义 Submodels 的结构**  
→ 包括静态信息（模型参数）+ 动态信息（实时状态、位置、动作等）。

 **搭建数据接口**  
→ 通过软件接口获取机器人运行时状态（关节角度、传感器数据等），动态填充 Submodels。

 **在 Unity 中实现**  
→ 同时展示 URDF 解析的几何结构 + 读取 Submodels 的动态状态，实现数字孪生的可视化和交互。

**✅ 第一阶段：理解 URDF，明确机器人基础信息**

**目标**：梳理机器人模型（2个）、运动结构与接口能力，为后续建模与联动打基础

| **子任务** | **目标说明** | **工具/产出** |
| --- | --- | --- |
| 1.1 | 熟悉 URDF 文件结构与语义 | URDF 学习笔记、模型解析文档 |
| 1.2 | 确定两个机器人型号、运动结构（移动/关节等） | 机器人型号信息、运动拓扑图 |
| 1.3 | 梳理各自可提供的接口与能力（如状态、位置、动作等） | 功能对照表（机器人1 vs 机器人2） |

**✅ 第二阶段：构建 AAS 系统（数字身份）**

**目标**：为每个机器人构建对应的 AAS（Asset Administration Shell）和基础 Submodel

| **子任务** | **目标说明** | **工具/产出** |
| --- | --- | --- |
| 2.1 | 为每个机器人建立 AAS 实例 | JSON 模板 / Eclipse BaSyx 配置文件 |
| 2.2 | 实现 1~2 个 IDTA Submodel（如 Location, Status） | Submodel JSON + 属性值动态模拟 |
| 2.3 | Unity 中连接 AAS，读取 Submodel 数据 | Unity HTTP/REST 脚本 |

**✅ 第三阶段：数据联动与交互逻辑**

**目标**：实现机器人间 AAS 数据交互 + Unity 中的可视反馈逻辑

| **子任务** | **目标说明** | **工具/产出** |
| --- | --- | --- |
| 3.1 | 实现 AAS 间“读取/响应”逻辑（如位置感知、指令共享） | AAS 交互脚本、逻辑模拟流程图 |
| 3.2 | Unity 中视觉反馈（如颜色变化、路径动画） | Unity 控制脚本、动画状态机 |
| 3.3 | 模拟一个协作任务流程（如A完成后通知B开始） | 简易状态机或任务调度机制 |

📌 **阶段目标**：你能在 Unity 中展示两个机器人间的“协作或通信”流程。

**✅ 第四阶段：整合 + 演示环境搭建**

**目标**：构建完整演示环境，整合所有组件实现一键运行展示

| **子任务** | **目标说明** | **工具/产出** |
| --- | --- | --- |
| 4.1 | 整合 AAS、URDF 模型、Unity 场景 | Unity 项目结构 + 场景搭建脚本 |
| 4.2 | 添加控制面板（显示状态、触发交互） | Unity UI + 按钮功能绑定 |
| 4.3 | 准备演示版本（可录屏展示完整流程） | 一键运行包 + 演示脚本或录像 |

📌 **阶段目标**：完整展示“机器人 → AAS → Unity → 交互”的过程。

**✅ 第五阶段：文档撰写与分析总结**

**目标**：产出文档与演示材料，便于项目总结与汇报

| **子任务** | **目标说明** | **工具/产出** |
| --- | --- | --- |
| 5.1 | 记录实现过程与各模块作用 | 项目结构图、模块说明文档 |
| 5.2 | 总结实现内容、当前限制、未来拓展点 | 分析报告 / 论文讨论材料 |
| 5.3 | 完成论文初稿 + 最终展示视频 | 报告、PPT、录屏演示视频 |

**✅ Phase 1: Understanding URDF and Identifying Robot Information**

**Goal**: Understand the robot models (2), their motion structure, and interface capabilities to support later modeling and interaction tasks.

| **Subtask** | **Description** | **Tools/Deliverables** |
| --- | --- | --- |
| 1.1 | Learn the structure and semantics of URDF files | URDF study notes, model analysis documentation |
| 1.2 | Identify two robot models and their motion structures (e.g. mobility, joints) | Robot model info, motion topology diagrams |
| 1.3 | Clarify the provided interfaces and capabilities (e.g. status, position, actions) | Capability comparison table (Robot 1 vs Robot 2) |

**✅ Phase 2: Building the AAS System (Digital Identity)**

**Goal**: Create an AAS (Asset Administration Shell) for each robot and define basic Submodels.

| **Subtask** | **Description** | **Tools/Deliverables** |
| --- | --- | --- |
| 2.1 | Create an AAS instance for each robot | JSON templates / Eclipse BaSyx configuration |
| 2.2 | Define and implement 1–2 IDTA Submodels (e.g. Location, Status) | Submodel JSON + simulated property updates |
| 2.3 | Connect Unity to the AAS and access Submodel data | Unity HTTP/REST scripts |

**✅ Phase 3: Data Interaction and Logic**

**Goal**: Implement AAS-based interaction logic and visualize robot behavior in Unity.

| **Subtask** | **Description** | **Tools/Deliverables** |
| --- | --- | --- |
| 3.1 | Implement “read/respond” logic between the two robot AAS instances (e.g. task commands, position sharing) | AAS communication scripts, interaction flow diagrams |
| 3.2 | Visual feedback in Unity (e.g. color change, path animation) | Unity control scripts, animation state machines |
| 3.3 | Simulate an interaction task (e.g. Robot A finishes and notifies Robot B to start) | Simple task chain or state machine |

📌 **Phase Goal**: You can demonstrate a "collaboration or communication" process between two robots in Unity.

**✅ Phase 4: Integration and Demo Environment Setup**

**Goal**: Build an integrated demo environment that combines all components for a one-click demonstration.

| **Subtask** | **Description** | **Tools/Deliverables** |
| --- | --- | --- |
| 4.1 | Integrate AAS, URDF models, and Unity scene | Unity project setup, scene scripts |
| 4.2 | Add a control panel (show status, trigger interactions) | Unity UI, button bindings |
| 4.3 | Prepare demo version (recordable end-to-end process) | One-click executable, demo script or recording |

📌 **Phase Goal**: Demonstrate the complete process from “Robot → AAS → Unity → Interaction”.

**✅ Phase 5: Documentation and Analysis**

**Goal**: Produce documentation and presentation materials for project summary and reporting.

| **Subtask** | **Description** | **Tools/Deliverables** |
| --- | --- | --- |
| 5.1 | Record the implementation process and roles of each module | Project architecture diagram, module descriptions |
| 5.2 | Summarize implemented features, limitations, and possible future improvements | Analysis report, discussion materials for paper |
| 5.3 | Complete the initial thesis draft and final presentation video | Report, presentation slides, demo video |