

# IADL 與 FDL 說明文件

版本: 1.0 日期: 2025-10-16 作者: 林志錚 Michael Lin(Chih Cheng Lin)

## 1. 簡介

本文件旨在提供 IDTF (Industrial Digital Twin Framework) 中 IADL (Industrial Asset Definition Language) 和 FDL (Factory Design Language) 的詳細說明。這兩種語言是 IDTF 的核心組成部分，用於標準化工業資產的定義和工廠佈局的描述，為數位分身 (Digital Twin) 的建立和管理奠定基礎。透過 IADL 和 FDL，IDTF 旨在實現工業數據模型的一致性、互操作性和可驗證性，從而加速工業數位轉型。

## 2. 工業資產定義語言 (IADL)

IADL 是一種用於定義工業資產類型 (AssetType) 及其相關屬性、行為、接口和關係的語言。它提供了一種標準化的方式來描述資產的「是什麼」和「能做什麼」，而不涉及其在實際工廠中的具體部署位置。IADL 的設計目標是實現資產定義的模組化、可重用性和可擴展性。

### 2.1 核心概念與特性

- 資產類型 (AssetType):** IADL 的基本單元，代表一類具有共同特徵和行為的工業設備或組件，例如「離心泵」、「馬達」或「閥門」。
- 屬性 (Properties):** 描述資產類型的靜態或動態特徵，如「額定功率」、「最大流量」或「製造商」。每個屬性都有名稱、數據類型和可選的單位。
- 組件 (Components):** 資產類型可以由其他資產類型作為組件構成，形成階層關係。例如，「離心泵」可能包含「馬達」和「泵體」作為其組件。
- 接口 (Interfaces):** 定義資產類型與外部系統或服務互動的方式，如「OPC UA」、「MQTT」或「REST」。接口描述了連接點和數據映射規則。
- 遙測點 (Telemetry Points):** 資產類型可以發布的實時數據點，如「當前流量」、「馬達溫度」。

- **命令 (Commands)**: 資產類型可以接收並執行的操作，如「啟動泵」、「停止泵」。
- **繼承 (Inheritance)**: 資產類型可以從其他資產類型繼承屬性和行為，實現定義的重用。

## 2.2 IADL 數據模型 (JSON Schema)

IADL 的數據模型基於 JSON Schema 定義，確保了其語法正確性和機器可讀性。以下是 IADL 核心結構的簡化表示：

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Industrial Asset Definition Language (IADL) Schema",
  "description": "Schema for defining industrial asset types and their characteristics.",
  "type": "object",
  "properties": {
    "iadlVersion": {"type": "string"},
    "assetTypes": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": {"type": "string", "description": "Unique identifier for the asset type"},
          "name": {"type": "string", "description": "Human-readable name"},
          "description": {"type": "string"},
          "baseType": {"type": "string", "description": "ID of base asset type"},
          "properties": {"type": "array", "items": {"$ref": "#/definitions/Property"}},
          "components": {"type": "array", "items": {"$ref": "#/definitions/ComponentReference"}},
          "interfaces": {"type": "array", "items": {"$ref": "#/definitions/Interface"}},
          "telemetry": {"type": "array", "items": {"$ref": "#/definitions/TelemetryPoint"}},
          "commands": {"type": "array", "items": {"$ref": "#/definitions/Command"}}
        },
        "required": ["id", "name"]
      }
    },
    "required": ["iadlVersion", "assetTypes"],
    "definitions": {
      "Property": { /* ... 屬性定義 ... */ },
      "ComponentReference": { /* ... 組件引用定義 ... */ },
      "Interface": { /* ... 接口定義 ... */ },
      "TelemetryPoint": { /* ... 遙測點定義 ... */ },
      "Command": { /* ... 命令定義 ... */ }
    }
  }
}
```

## 2.3 IADL 範例：離心泵 (Centrifugal Pump)

以下是一個定義「離心泵」資產類型的 IADL 範例，展示了其屬性、組件、遙測點和命令：

```
{
  "iadlVersion": "1.0.0",
  "assetTypes": [
    {
      "id": "CentrifugalPump_V1",
      "name": "Centrifugal Pump",
      "description": "A standard centrifugal pump for fluid transfer.",
      "properties": [
        {
          "name": "MaxFlowRate",
          "dataType": "number",
          "unit": "m3/h",
          "description": "Maximum fluid flow rate."
        },
        {
          "name": "RatedPower",
          "dataType": "number",
          "unit": "kW",
          "description": "Rated electrical power."
        }
      ],
      "components": [
        {
          "componentTypeId": "Motor_AC_V2",
          "quantity": 1,
          "name": "MainMotor"
        }
      ],
      "telemetry": [
        {
          "name": "CurrentFlowRate",
          "dataType": "number",
          "unit": "m3/h",
          "description": "Current fluid flow rate."
        },
        {
          "name": "MotorTemperature",
          "dataType": "number",
          "unit": "°C",
          "description": "Motor winding temperature."
        }
      ],
      "commands": [
        {
          "name": "StartPump",
          "description": "Starts the pump."
        },
        {
          "name": "StopPump",
          "description": "Stops the pump."
        }
      ]
    }
  ]
}
```

## 3. 工廠設計語言 (FDL)

FDL 是一種用於描述工廠的物理佈局、資產實例 (AssetInstance) 及其之間階層關係的語言。它將 IADL 定義的資產類型實例化到具體的工廠環境中，構建出工廠的數位分身模型。FDL 的設計目標是實現工廠佈局的精確描述、資產實例的唯一識別和階層關係的清晰表達。

### 3.1 核心概念與特性

- **工廠 (Factory)**: FDL 的頂層單元，代表一個完整的生產設施。
- **站點 (Site)**: 工廠內的一個地理區域，可以包含多個區域。
- **區域 (Area)**: 站點內的一個功能性分區，如「組裝區」、「加工區」。
- **製程單元 (Process Cell)**: 區域內的一個獨立生產單元，執行特定的製程。
- **單元 (Unit)**: 製程單元內的一個操作單元，包含多個資產實例。
- **資產實例 (AssetInstance)**: IADL 中定義的 `AssetType` 在實際工廠中的具體實例。每個實例都有唯一的 ID、名稱，並引用其對應的 `assetTypeId`。
- **階層關係 (Hierarchical Relationships)**: FDL 透過嵌套結構來表達工廠中資產和區域之間的物理和邏輯階層關係，例如工廠包含站點，站點包含區域，區域包含製程單元，製程單元包含單元，單元包含資產實例。
- **位置資訊 (Location)**: 資產實例可以包含詳細的物理位置資訊，如座標和所在房間。
- **操作狀態 (Operational Status)**: 資產實例的當前運行狀態，如「運行中」、「停止」、「維護中」。

### 3.2 FDL 數據模型 (JSON Schema)

FDL 的數據模型同樣基於 JSON Schema 定義，確保了其語法正確性和機器可讀性。以下是 FDL 核心結構的簡化表示：

```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Factory Design Language (FDL) Schema",
  "description": "Schema for defining factory layout and asset instances.",
  "type": "object",
  "properties": {
    "fdlVersion": {"type": "string"},
    "factory": {
      "type": "object",
      "properties": {
        "id": {"type": "string", "description": "Unique identifier for the factory"},
        "name": {"type": "string", "description": "Human-readable name"},
        "description": {"type": "string"},
        "sites": {"type": "array", "items": {"$ref": "#/definitions/Site"}}
      },
      "required": ["id", "name"]
    }
  },
  "required": ["fdlVersion", "factory"],
  "definitions": {
    "Site": { /* ... 站點定義 ... */ },
    "Area": { /* ... 區域定義 ... */ },
    "ProcessCell": { /* ... 製程單元定義 ... */ },
    "Unit": { /* ... 單元定義 ... */ },
    "AssetInstance": { /* ... 資產實例定義 ... */ }
  }
}

```

### 3.3 FDL 範例：工廠佈局 (Factory Layout)

以下是一個描述工廠佈局的 FDL 範例，展示了資產實例如何在工廠階層中被組織：

```

{
  "fdlVersion": "1.0.0",
  "factory": {
    "id": "Factory_A",
    "name": "Main Production Plant A",
    "description": "Primary manufacturing facility.",
    "sites": [
      {
        "id": "Site_01",
        "name": "Site One",
        "areas": [
          {
            "id": "Area_Assembly",
            "name": "Assembly Line 1",
            "processCells": [
              {
                "id": "ProcessCell_PumpAssembly",
                "name": "Pump Assembly Process Cell",
                "units": [
                  {
                    "id": "Unit_Mixing",
                    "name": "Mixing Unit",
                    "assetInstances": [
                      {
                        "id": "Pump_001",
                        "name": "Mixing Pump 1",
                        "assetTypeId": "CentrifugalPump_V1",
                        "serialNumber": "SN-PUMP-001-XYZ",
                        "location": {
                          "x": 10.5,
                          "y": 20.1,
                          "z": 5.0,
                          "room": "Mixing Room A"
                        },
                        "operationalStatus": "Running"
                      }
                    ]
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  }
}

```

## 4. 結論

IADL 和 FDL 作為 IDTF 的基石，提供了強大且靈活的機制來定義和描述工業資產及其在工廠環境中的部署。透過標準化的 JSON Schema 和清晰的階層結構，它們確保了數位分身數據模型的一致性、互操作性和可驗證性。這不僅簡化了工業數位分身的建立和管理，也為進一步的數據分析、AI 應用和智慧製造奠定了堅實的基礎。