Ontology Construction Approach

The development of the ontology for integrating Digital Product Passports (DPPs) into Enterprise Asset Management (EAM) systems started by identifying the key data and concepts needed. This was done through literature review, interviews with industry professionals, and a survey aimed at understanding current practices. The findings highlighted asset information is often scattered across multiple systems, stored in various formats, and difficult to access or integrate efficiently. There is also a lack of common standards for how product data should be structured and shared. The insights from the review, interviews, and survey helped define the important elements and relationships that the ontology should include. This work forms the basis for building a structured, flexible model that can improve data integration, support better maintenance planning, and help organizations move toward more sustainable and circular asset management.

Table 1: Interview and SurveyFindings

Interview Findings	**Survey Findings**
Asset data is fragmented across spreadsheets, PDFs, and internal systems like Maximo.	Most organizations use manual methods (e.g., spreadsheets) for asset tracking.
Lack of standardized structure and validation limits interoperability.	Top challenges: lack of standardized data, difficulty tracking maintenance history.
QR codes are already used in some products for reordering and identification.	DPPs are seen as beneficial for traceability, compliance, and sustainability.
There are concerns about data confidentiality and intellectual property when sharing detailed product information.	Main barriers: cost of upgrades, lack of standards, resistance to new technology.
Semantic technologies not yet used but recognized for their potential.	Regulatory compliance and standardization are seen as key adoption drivers.
Confidentiality concerns limit data sharing, especially from manufacturers.	Most valued DPP data: maintenance history, energy consumption, environmental impact.

MOMo methodology Workflow

The Modular Ontology Modeling (MOMo) workflow methodology developed by (Shimizu et al. 2023) is applied within the DSRM's design stage to develop the modular ontology in a structured and reusable manner, aligned with stakeholder needs and expectations. At this stage the MOMo workflow plays a vital role in guiding how to address the reuse or adapting ontologies for a new use case or purpose. The MOMo methodology defines processes for developing modular ontologies which in context of ontology development for DPPs is a more suitable approach since it contains a large amount of data (Kebede et al. 2024)

Table 1: The MOMo methodology workflow adapted according to ***(Shimizu et al. 2023)***

Step	Responsible	Output
1. Describe use cases & data sources	Entire team	Use case descriptions
2. Gather competency questions	Entire team	List of CQs
3. Identify key notions	Entire team	List of key notions
4. Identify existing ODPs	Ontology engineers	Selected ODP(s) for each key notion.
5. Create module diagrams	Entire team	Diagrammatic representation of the solution module.
6. Document modules & axioms	Ontology engineers & domain experts	Module documentation with embedded schema diagrams, axiomatization, etc. (e.g., in LaTeX, Word, HTML format).
7. Create ontology diagram	Ontology engineers	Diagrammatic representation of the whole composed ontology.
8. Add spanning axioms	Ontology engineers	Documentation of the entire ontology with embedded schema diagrams, axiomatization, etc. (e.g., in LaTeX, Word, HTML format).
9. Review naming & axioms	Ontology engineers	Updated module and ontology documentation.
10. Create OWL file & axioms	Ontology engineers	An OWL file for publication and use.

1. Describe use cases & data sources

In industries that depend on physical assets such as manufacturing, infrastructure, and construction, efficient **lifecycle** management is critical to ensuring reliability, cost-efficiency, and sustainability. However, many **organizations** still rely on fragmented systems, spreadsheets, and manual documentation, resulting in poor traceability and limited data interoperability. Survey responses in this study indicated heavy reliance on non-standardized asset tracking methods, while interviews with domain professionals highlighted challenges related to data integration and inconsistency across asset management platforms. The **Digital Product Passport (DPP)**, mandated by the European Union's Eco-design for Sustainable Products Regulation (ESPR), presents an opportunity to address these issues by providing machine-readable, structured data about a **product's origin**, **composition**, **usage, maintenance history, and environmental impact**. This use case focuses on integrating **DPP data** into an **Enterprise Asset Management system** (IBM Maximo) using semantic technologies to improve **predictive maintenance** and sustainability tracking. For example, each **lighting fixture** is equipped with a **QR code** linked to its DPP, which contains semantically structured data (e.g., **technical specification**, **carbon footprint**, **lifespan** etc.). When scanned in the field, the data is accessed via a mobile interface, allowing **maintenance staff** to retrieve **instructions**, **log issues**, and trigger feedback loops. The ontology will be designed to support integration of heterogeneous sources such as **manufacturer** details, **maintenance activities**, **Materials** and **regulatory documents** etc., enabling a modular, standard-aligned, and reusable structure for **asset information** management within EAM environments.

2. Gather competency questions

1. What Digital Product Passport (DPP) is associated with a product?
2. What repair instructions are available for a given product?
3. What are the characteristics and composition of a product as per its DPP?
4. Who is the responsible actor for a product's DPP?
5. What is the model and serial number of a product?
6. What materials and components make up a product?
7. What certifications does a product have?
8. Who manufactured a product?
9. What are the physical and chemical properties of a material?
10. Is a component made of hazardous material?
11. What is the expected lifetime of a component?
12. Who manufactured a material or component?
13. Who is the manufacturer of a product or component?
14. What role does an agent play in manufacturing or distribution?
15. What is the provenance of a product? Which certifications do a product hold?
16. Who is the certifying authority for a product?
17. What documents are associated with a certification?
18. What lifecycle stage is the product currently in?
19. What environmental impacts are associated with a product?
20. Which agent performed the design or manufacturing of a product?
21. What maintenance tasks are associated with a product?
22. What is the cost or duration of a maintenance activity?
23. Who performed a given maintenance task?
24. Who is the sender or receiver in a transportation event?
25. What is the identifier of a supply chain event?
26. Where and when did a supply chain event occur?

3. Identify key notions

Table 3: Identified Keynotions

Product	Access Control	Failure Event
Digital Product Passport (DPP)	Repair Instruction	Data Format
Component	Installation	QR Code
Usage	Energy Consumption	RFID Tag
Lifecycle	Environmental Impact	System Identifier
Maintenance Activity	Material Composition	log issues
Maintenance History	Service Log	Process
Data Carrier	Location	Data
User Role	Disassembly Information	Operational Status
Instructions	Technical Specification	Type
Supplier	Carbon Footprint	Certification
Manufacturer	Ownership History	Task
Regulatory Standard	End-of-Life Process	Asset information
Carbon footprint	Maintenance staff	organizations

4. Identify existing ODPs

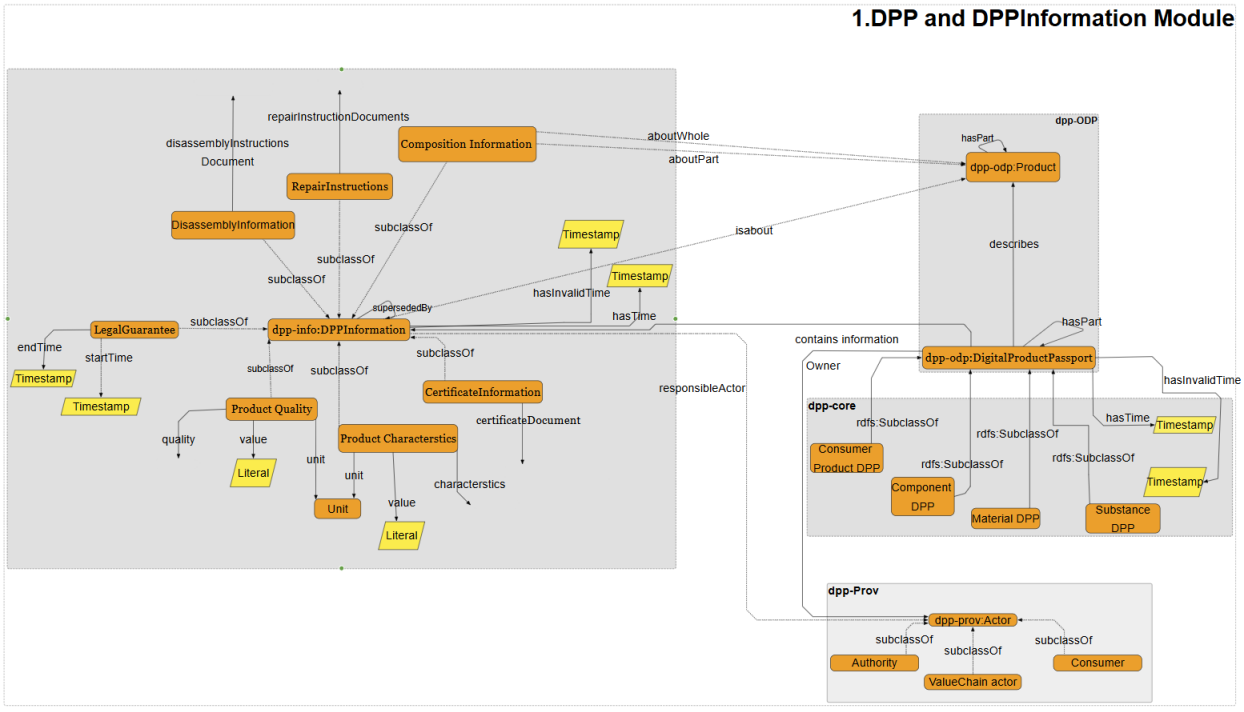
To promote interoperability and avoid redundancy, the proposed ontology strategically reuses existing ontological constructs from well-established vocabularies. Table 4 summarizes the reused elements. Core domain concepts such as *Quantity*, *QuantityKind*, *QuantityValue*, *Unit*, *Product*, *Material*, *Agent*, *Place*, and *Manufacturer* are integrated primarily from QUDT, Schema.org, and PROV-O, ensuring alignment with widely adopted Linked Data standards.Object properties such as *used*, *wasGeneratedBy*, *atTime*, and *attributedTo* are reused from **PROV-O** to describe processes, roles, and how data is connected over time. The *numericValue* property from **QUDT** is used to represent numbers related to quantities in a consistent way.By reusing these standard terms, the ontology becomes easier to integrate with other systems and supports better data sharing across different use cases.

Table 4: concept and properties that are used in DPP4EAM Ontology

Ontology Part	Element Names	Ontology
Concepts	Quantity, QuantityKind, QuantityValue, Unit, Role, Entity, Activity, Product, Manufacturer, Agent, Material, Name, Identifier, Place	QUDT, PROV-O, Schema.org
Object Properties	used, wasGeneratedBy, attributedTo, derivedFrom, hasPart, supersededBy, atTime	PROV-O, Schema.org
Datatype Properties	numericValue	QUDT

5-6. Create module diagrams - Document modules & axioms

1.DPP and DPP info Module



Axiom:

Product ⊑ ∃hasDPP.DigitalProductPassport

➔ **Explanation:** A product must be linked to its Digital Product Passport (DPP).

DigitalProductPassport ⊑ ∃containsInformation.DPPInformation

➔ **Explanation:** A DPP must contain relevant product information.

DigitalProductPassport ⊑ ∃hasPart.ComponentDPP

➔ **Explanation:** A DPP must describe at least one component.

ComponentDPP ⊑ ∃hasTime.Time

➔ **Explanation:** Every component's lifecycle in the DPP must include a timestamp.

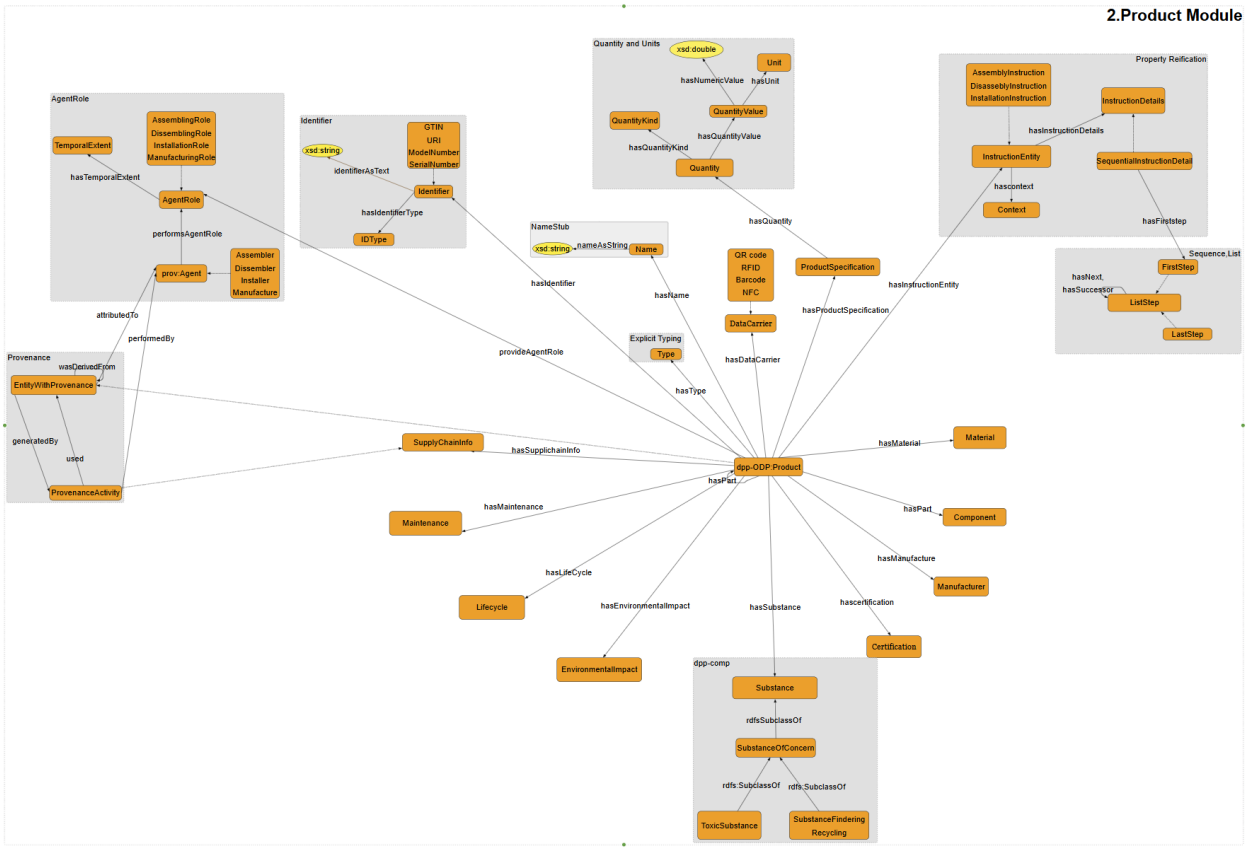
CertificateInformation ⊑ ∃certificateDocument.Document

➔ **Explanation:** Certification information must reference an official document.

SPARQL Query:

```
SELECT ?dpp ?product
WHERE {
  ?dpp dpp:describes ?product .
}
SELECT ?dpp ?actor
WHERE {
  ?dpp dpp:responsibleActor ?actor .
}
SELECT ?product ?certificate
WHERE {
  ?product dpp:certificateDocument ?certificate .
}
```

2.Product Module



Axiom:

Product ⊑ ∃hasIdentifier.Identifier

➔ **Explanation:** Every product must be associated with a unique identifier.

Product ⊑ ∃hasName.Name

➔ **Explanation:** Each product must have a name to be identified by humans.

Product ⊑ ∃hasPart.Component

➔ **Explanation:** A product is composed of one or more components.

Product ⊑ ∃hasSpecification.Specification

➔ **Explanation:** Products must have associated specifications.

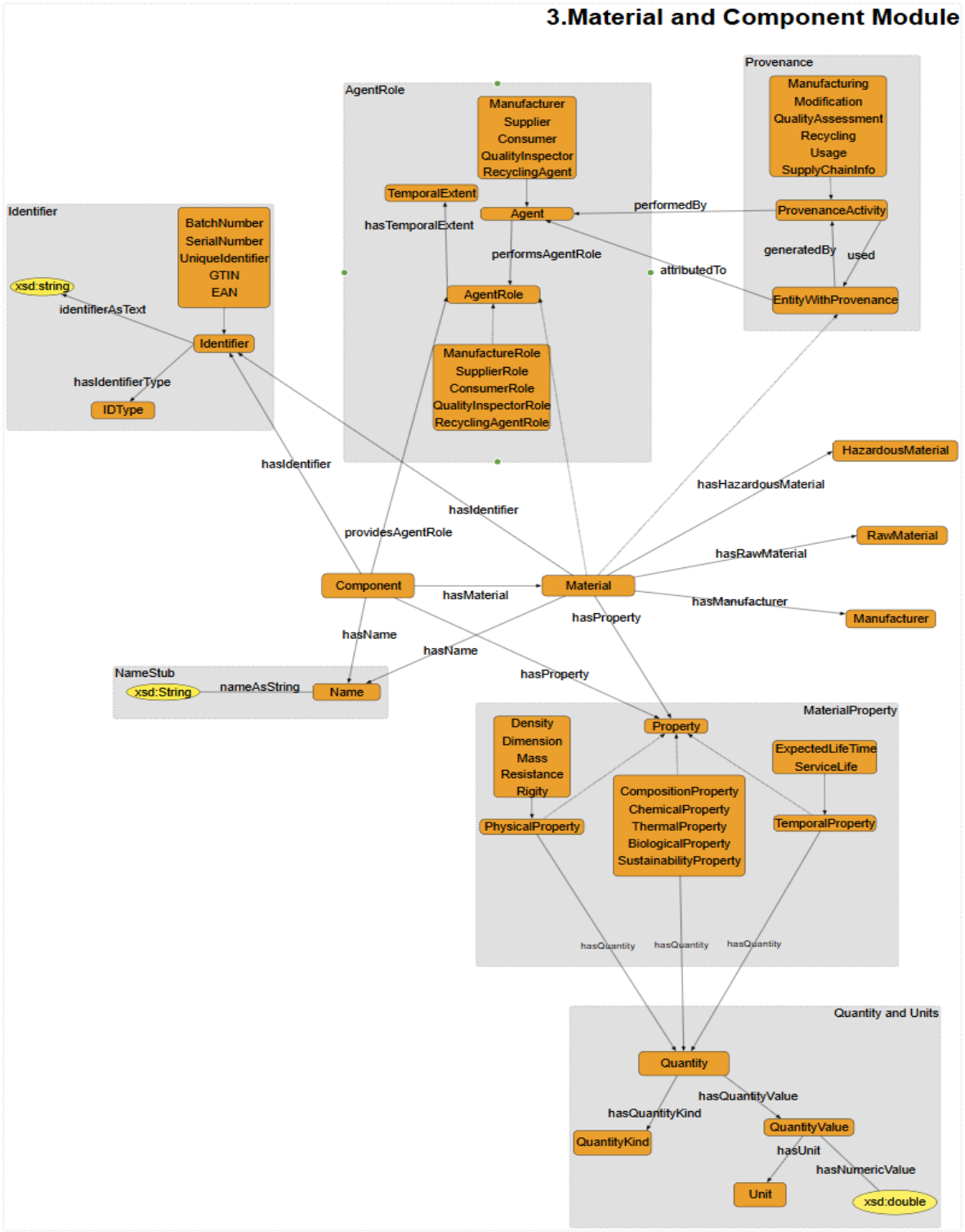
Product ⊑ ∃hasType.Type

➔ **Explanation:** Each product must be classified under a type.

SPARQL Query:

```
SELECT ?product ?name
WHERE {
  ?product prod:hasName ?name .
}
SELECT ?identifier ?type
WHERE {
  ?identifier prod:hasIdentifierType ?type .
  OPTIONAL { ?identifier a schema:identifier . }
}
SELECT ?product ?spec
WHERE {
  ?product prod:hasSpecification ?spec .
}
SELECT ?product ?instruction
WHERE {
  ?product prod:hasInstructionDetail ?instruction .
}
```

3.Material and Componet Module



Axiom:

Component ⊑ ∃hasMaterial.Material

→ **Explanation:** A component must be associated with some material.

Material ⊑ ∃hasProperty.PhysicalProperty

→ **Explanation:** A material must have at least one physical property.

Product ⊑ ∃hasComponent.Component

→ **Explanation:** Every product must be composed of at least one component.

Component ⊑ ∃hasIdentifier.Identifier

→ **Explanation:** Each component must have a unique identifier.

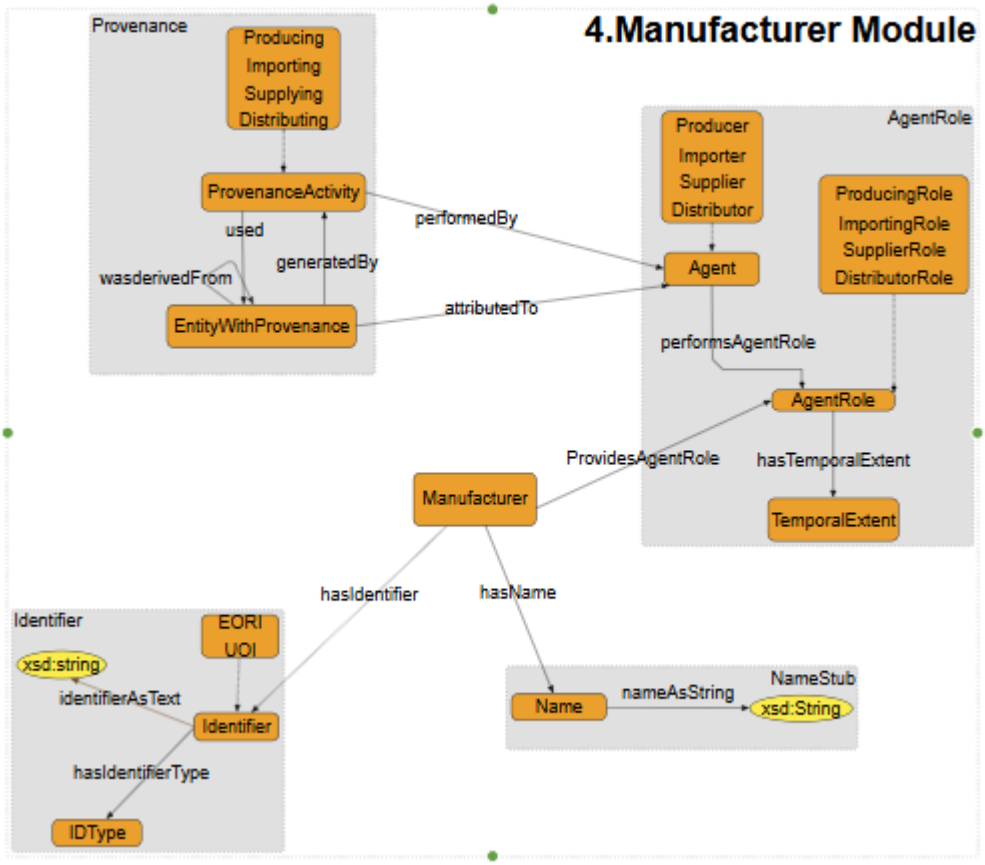
Material ⊑ ∃hasQuantity.Quantity

→ **Explanation:** Materials must be linked to a measurable quantity.

SPARQL Query:

```
SELECT ?product ?material
WHERE {
  ?product mc:hasMaterial ?material .
}
SELECT ?product ?component
WHERE {
  ?product mc:hasComponent ?component .
}
SELECT ?entity ?identifier ?name
WHERE {
  ?entity mc:identifierAsText ?identifier .
  OPTIONAL { ?entity mc:hasName ?name . }
}
```

4.Manufacturer Module



Axiom:

Product ⊑ ∃hasManufacturer.Manufacturer

→ **Explanation:** Every product must be linked to a manufacturer.

Manufacturer ⊑ ∃hasIdentifier.Identifier

→ **Explanation:** Each manufacturer must be uniquely identifiable.

Manufacturer ⊑ ∃providesAgentRole.ManufacturingRole

→ **Explanation:** Manufacturers provide the role of manufacturing agents.

ManufacturingRole ⊑ ∃performedBy.Manufacturer

→ **Explanation:** Each manufacturing role must be linked to the manufacturer that performs it.

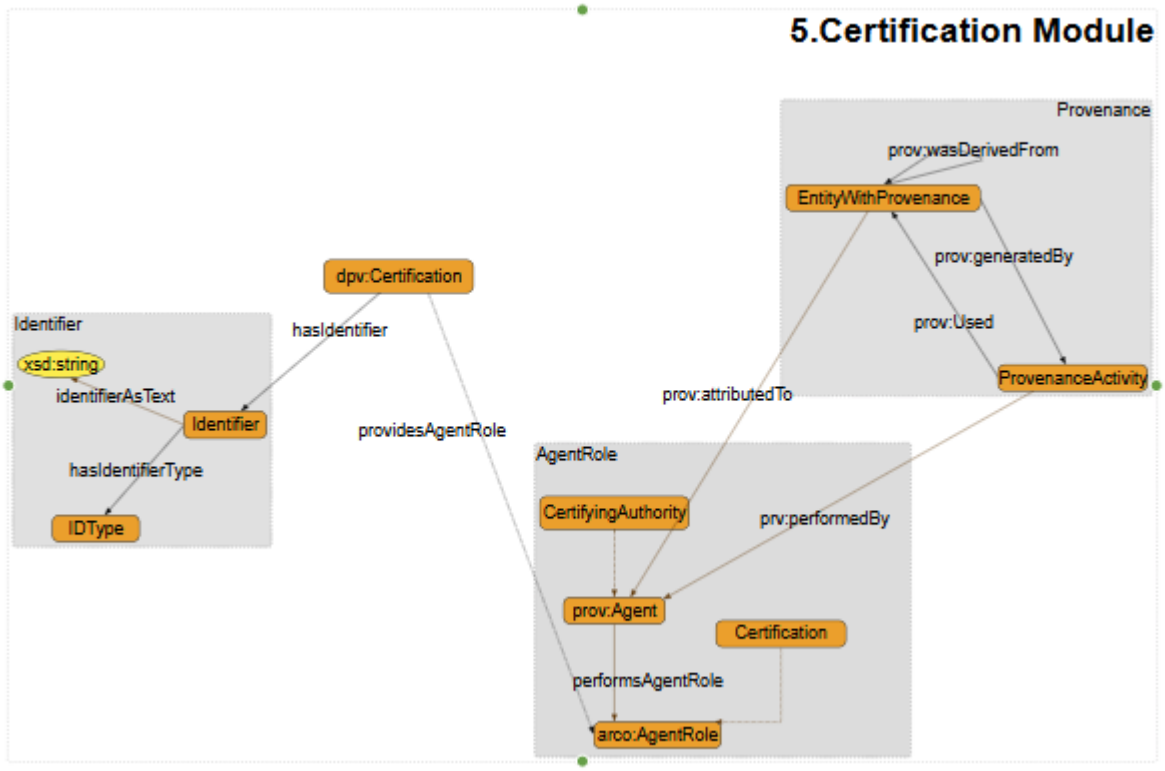
Manufacturer ⊑ ∃hasName.Name

→ **Explanation:** A manufacturer must have a name as an identifying attribute.

SPARQL Query:

```
SELECT ?product ?manufacturer
WHERE {
  ?product manuf:hasManufacturer ?manufacturer .
}
SELECT ?identifier ?idType
WHERE {
  ?identifier manuf:hasIdentifierType ?idType .
}
SELECT ?entity ?idText
WHERE {
  ?entity manuf:identifierAsText ?idText .
}
```

5.Certification Module



Axiom:

Product ⊑ ∃hasCertification.Certification

→ **Explanation:** Every product must be linked to its certification details.

Certification ⊑ ∃issuedBy.Authority

→ **Explanation:** Each certification must be issued by an authority.

Certification ⊑ ∃certificateDocument.Document

→ **Explanation:** Certifications should be accompanied by documentation.

Certification ⊑ ∃hasIdentifier.Identifier

→ **Explanation:** Each certification must have a unique identifier.

Certification ⊑ ∃hasTime.Time

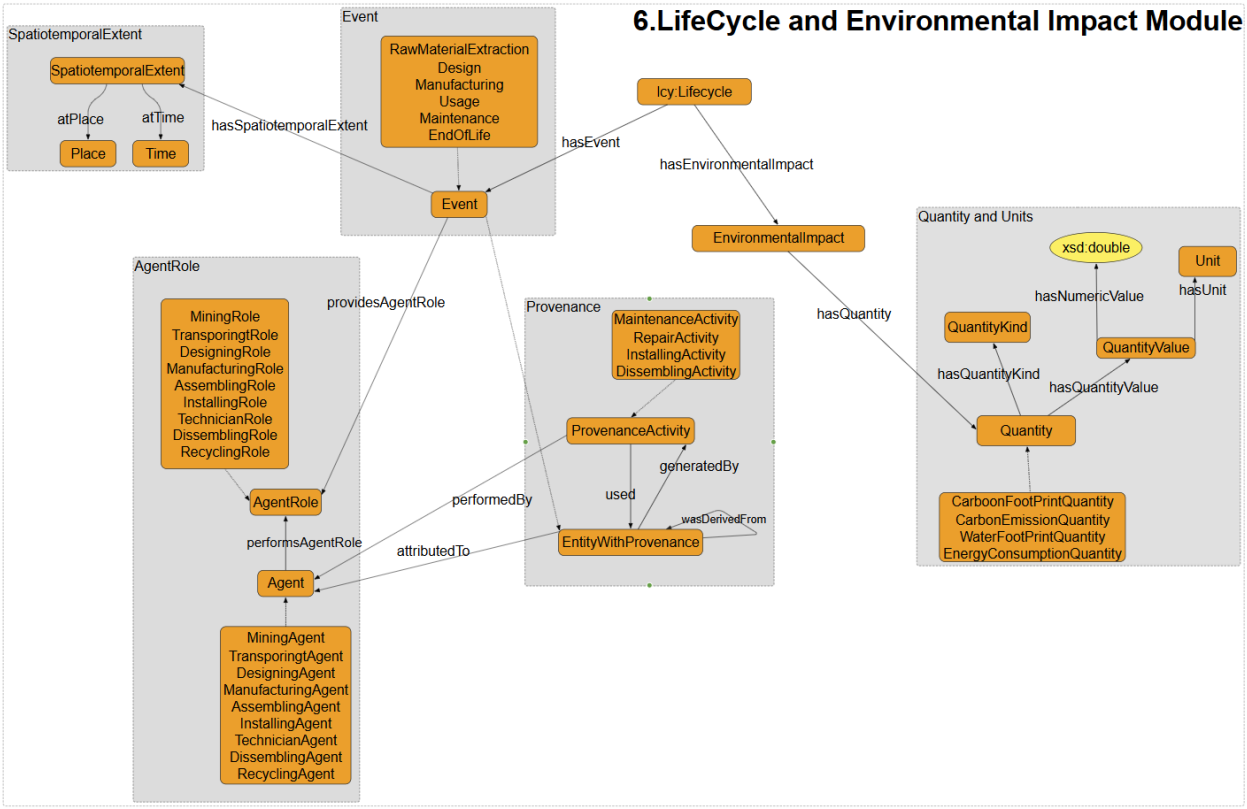
→ **Explanation:** Certifications should include issuance or expiration time.

SPARQL Queries:

```
SELECT ?identifier ?idType
WHERE {
  ?identifier a cert:IDType .
}

SELECT ?product ?cert
WHERE {
  ?product cert:hasCertification ?cert .
}
```

6.LifeCycle and Environmental Impact Module



Axiom:

Product ⊑ ∃hasLifeCycle.LifeCycle

➔ **Explanation:** Each product must be linked to a life cycle.

LifeCycle ⊑ ∃hasEnvironmentalImpact.EnvironmentalImpact

➔ **Explanation:** A lifecycle must be associated with environmental impact information.

EnvironmentalImpact ⊑ ∃hasQuantityValue.QuantityValue

➔ **Explanation:** Environmental impacts are measured and expressed as quantity values.

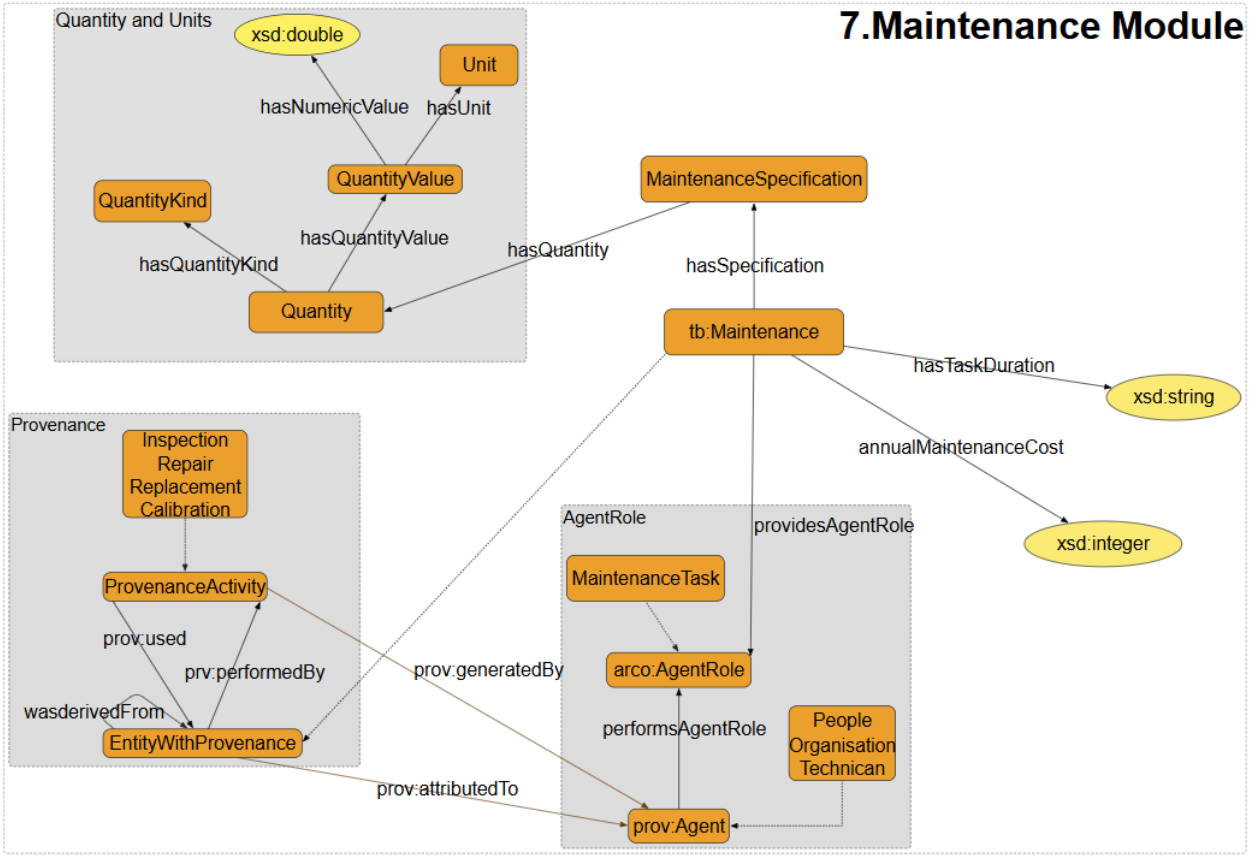
QuantityValue ⊑ ∃hasUnit.Unit

➔ **Explanation:** A quantity value must be expressed using a specific unit.

SPARQL Query:

```
SELECT ?lifecycle ?event
WHERE {
  ?lifecycle lcei:hasEvent ?event .
}
SELECT ?entity ?impact
WHERE {
  ?entity lcei:hasEnvironmentalImpact ?impact .
}
SELECT ?event ?place
WHERE {
  ?event lcei:atPlace ?place .
}
```

7.Maintenace Module



Axiom:

Product ⊑ ∃hasMaintenance.Maintenance

➔ **Explanation:** Each product must be associated with maintenance information.

Maintenance ⊑ ∃performedBy.Agent

➔ **Explanation:** Maintenance must be performed by an agent (e.g., technician).

Maintenance ⊑ ∃hasTemporalExtent.Time

➔ **Explanation:** Maintenance includes temporal data like start or due dates.

Maintenance ⊑ ∃hasIdentifier.Identifier

➔ **Explanation:** Maintenance procedures should have unique identifiers.

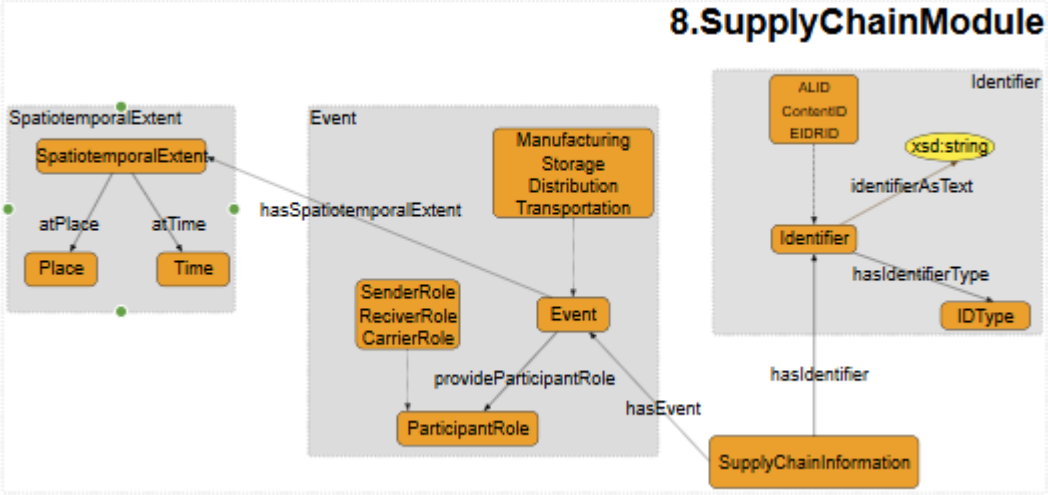
Agent ⊑ ∃performsActivity.Maintenance

➔ **Explanation:** An agent (e.g., technician or service provider) performs maintenance activities.

SPARQL Query:

```
SELECT ?entity ?maintenance
WHERE {
  ?entity maint:hasMaintenance ?maintenance .
}
SELECT ?task ?duration
WHERE {
  ?task maint:taskDuration ?duration .
}
SELECT ?task ?agent
WHERE {
  ?task maint:PerformedBy ?agent .
}
SELECT ?maintenanceEntity ?unit
WHERE {
  ?maintenanceEntity maint:hasUnit ?unit .
}
```

8.SupplyChain Module



Axiom:

Product ⊑ ∃hasSupplyChainInfo.SupplyChainInformation

➔ **Explanation:** Each product must be associated with supply chain information.

Supplier ⊑ ∃hasName.Name

➔ **Explanation:** A supplier should have an identifiable name.

SupplyChainInformation ⊑ ∃hasIdentifier.Identifier

➔ **Explanation:** Supply chain data should be uniquely identified.

Supplier ⊑ ∃providesAgentRole.SupplierRole

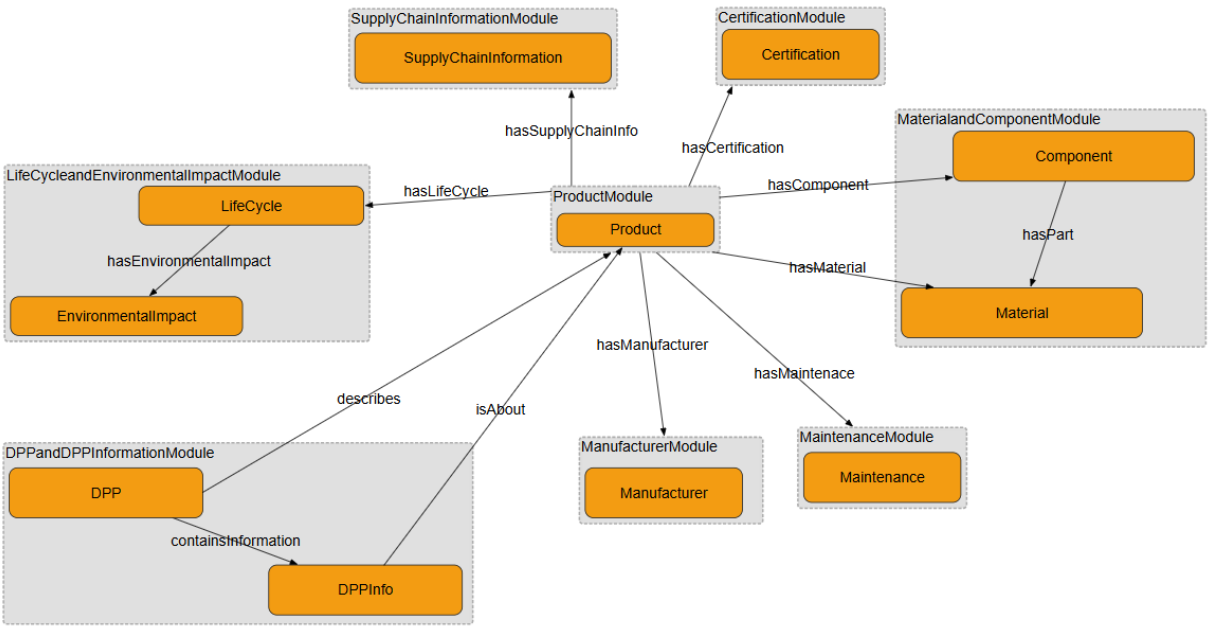
➔ **Explanation:** Suppliers are defined by the roles they provide within the supply chain context.

SPARQL Query

```
SELECT ?product ?event
WHERE {
  ?product sci:hasEvent ?event .
}

SELECT ?event ?time
WHERE {
  ?event a sci:Event .
  ?extent prov:atTime ?time .
}
```

7. Create ontology diagram



8&9.Add spanning axioms - Review naming & axioms

DPPandDPPInfo ⊑ ∃describes.Product

→ **Explanation:** The DPP describes the product.

Product ⊑ ∃hasManufacturer.Manufacturer

Explanation: Every product must have a defined manufacturer entity.

Product ⊑ ∃hasLifeCycleandEnvironmentalImpact.LifeCycleandEnvironmentalImpact

Explanation: A product has lifecycle and impact data.

Product ⊑ ∃hasMaterialandComponent.MaterialandComponent

Explanation: A product is made of materials or components

Product ⊑ ∃hasCertification.Certification

Explanation: A product has at least one certification.

Product ⊑ ∃hasSupplyChainInfo.SupplyChainInformation

Explanation: A product includes supply chain information.

Product ⊑ ∃hasMaintenance.Maintenance

Explanation: A product may need maintenance.

SPARQL Query:

```
SELECT ?product ?component ?material ?manufacturer ?certification
WHERE {
  OPTIONAL { ?product core:hasComponent ?component . }
  OPTIONAL { ?product core:hasMaterial ?material . }
  OPTIONAL { ?product core:hasManufacturer ?manufacturer . }
  OPTIONAL { ?product core:hasCertification ?certification . }
}
```

10. Create OWL file & axioms

modules\DPPmodule.owl

modules\Productmodule .owl

modules\Material and Component module.owl

modules\ManufacturerModule.owl

modules\CertificationModule.owl

modules\MaintenanceModule.owl

modules\LifeCycle and EnvironmentalImpact Module .owl

modules\SupplychainModule.owl