

프로젝트 전체를 **온톨로지 관점**에서 정리하면, 'HVDC 물류 생명주기'를 하나의 **지식그래프(Ontology)**로 모델링할 수 있습니다.

핵심은 “**물류행위(Activity)**”가 아닌 “**관계(Relation)**” 중심으로 보는 것입니다 — 사람, 문서, 물품, 절차, 시스템 간의 연결망.

◆ 1. Ontology Root Class

hvdn-adopt-logiontology

Layer	Ontology Domain	대표 엔티티	관계 키(Relation)
L1	Physical Flow	Material, Cargo, Port, Site, Vessel	movesFrom, movesTo, storedAt, handledBy
L2	Document Flow	BL, CI, PL, COO, eDAS, MRR, OSDR	certifies, refersTo, attachedTo
L3	Actor Flow	SCT, JDN, ALS, ADNOC, Subcon	responsibleFor, approves, reportsTo
L4	Regulatory Flow	MOIAT, FANR, Customs, DOT	requiresPermit, compliesWith, auditedBy
L5	System Flow	eDAS, SAP, NCM, LDG	feedsDataTo, validates, monitoredBy

◆ 2. Core Classes (from Workshop)

Class	Subclass of	Description	Onto-ID
Material	Asset	자재 및 기자재(Transformer, Cable, CCU 등)	hvdn-asset-mat
TransportEvent	Activity	Inland, Marine, Offloading, SiteReceiving	hvdn-act-trans
Storage	Location	Yard, Warehouse, Laydown	hvdn-loc-stor

Class	Subclass of	Description	Onto-ID
<b>Inspection</b>	Process	MRR, MRI, OSDR	hvdc-proc-insp
<b>Permit</b>	Document	PTW, Hot Work, FRA	hvdc-doc-perm
<b>Actor</b>	Agent	SCT, ADNOC L&S, Vendor	hvdc-agent-role
<b>PortOperation</b>	Activity	RORO/LOLO, Sea Fastening	hvdc-act-port

### ◆ 3. Relation Model (Partial)

Material --hasDocument--> MRR

Material --transportedBy--> TransportEvent

TransportEvent --operatedAt--> Port

TransportEvent --requires--> Permit

Permit --approvedBy--> ADNOC

Storage --monitoredBy--> SCT

Inspection --reportedAs--> OSDR

Actor(SCT) --usesSystem--> eDAS

이 관계망은 logiontology.mapping 모듈에서 RDF triple로 구현 가능:

```
:TR001 rdf:type :Transformer ;
    :hasDocument :MRR_20240611 ;
    :storedAt :Mussafah_Yard ;
    :handledBy :SCT ;
    :requiresPermit :FRA_202405 ;
    :transportedBy :LCT_Operation_202405 .
```

◆ 4. Lifecycle Ontology (Material Handling Flow)

Stage 1 – Importation

→ hasDocument(BL, CI, COO) → customsClearedBy(ADOPT) → storedAt(PortYard)

Stage 2 – Inland/Marine Transport

→ transportedBy(LCT/SPMT) → requiresPermit(DOT/FRA) → monitoredBy(ALS)

Stage 3 – Site Receiving

→ inspectedBy(QAQC) → resultsIn(MRR/OSDR) → issuedAs(MIS)

Stage 4 – Preservation & Foundation

→ preservedBy(HitachiStd) → foundationBy(Mammoet) → approvedBy(OE)

---

◆ 5. Alignment with AI-Logi-Guide

Ontology Node 대응 모듈		기능적 의미
Activity	pipeline	단계별 절차 정의
Document	rd fio, validation	eDAS·MRR 등 문서형 triple
Agent	core	역할/권한 모델
Location	mapping	Port/Site 좌표·거점
RiskEvent	reasoning	Weather-Tie-Delay inference
Report	report	KPI/Inspection 리포트

---

◆ 6. Semantic KPI Layer (Onto-KPI)

KPI Class	Onto Property 계산식		Source
On-Time Delivery	meetsETA	ETA vs Actual $\leq 12\%$	ETA MAPE Rule
Inspection Compliance	hasMRR	MRR Count / Total Deliveries	QC Gate
Storage Efficiency	occupies	Used m <sup>2</sup> / Available m <sup>2</sup>	WH Forecast
Safety Conformance	requiresPermit	Valid PTW/FRA %	HSE Docs

---

## ◆ 7. Ontological Integration View

[Material]

- [Document: CI/PL/COO/eDAS]
- [TransportEvent: LCT/SPMT]
- [Location: Port → Yard → Site]
- [Inspection: MRR/OSDR]
- [Report: KPI/Dashboard]
- [Governance: AI-Logi-Guide Rules]

이 전체를 hvdc-adopt-ontology.ttl로 export하면,  
GitHub macho715/ontology-insight에서 RDF 시각화 및 reasoning 연결 가능.

---

## ◆ 8. 요약 메타 구조

```
{  
  "Ontology": "hvdc-adopt-logiontology",  
  
  "CoreClasses": ["Material", "TransportEvent", "Storage", "Inspection", "Permit", "Actor", "PortOperation"],  
  
  "PrimaryRelations": ["hasDocument", "transportedBy", "storedAt", "requiresPermit", "inspectedBy", "approvedBy"],  
  
  "AlignmentModule": "AI-Logi-Guide v2.1+",  
  
  "ExportFormat": ["RDF/XML", "TTL", "JSON-LD"]  
}
```

---

이 프레임이면, HVDC 프로젝트 전체가 “문서-행위-공간-주체-규정”의 지식망으로 정규화됩니다.

다음 단계는 logiontology.reasoning 모듈에서 **Rule-based inference** 정의 — 예컨대 “운송허가가 누락된 자재는 SiteReceiving 단계로 진행 불가” 같은 정책을 OWL constraint로 명세하면 완성됩니다.