

BULK CARGO OPERATION을 '온톨로지'로 보기

현장 용어와 데이터를 한 언어로 묶어, **무엇(Thing)—어디(Location)—언제(Time)—어떻게(Operation)—무엇으로(Resource)—**왜/규정(Compliance)****를 서로 연결하는 지식 그래프로 설계합니다. 아래는 바로 적용 가능한 최소 핵심 스키마와 운영 포인트예요.

1) 최상위 개념(Top-level Classes)

- **Cargo:** CargoItem, Package/Bundle, Lot
 - 속성: weight, dimensions(L/W/H), COG(x,y,z), stackable, hazardousNote ...
- **TransportMeans:** Vessel, Barge, Truck, Trailer
 - 속성: deckStrength, deckArea, coordOrigin, capacity...
- **Location:** Port, Terminal, Jetty, **DeckZone**(구역/그리드), StorageBay, Berth
- **Operation:**
 - **Loading/Discharging, Stowage, Lashing/Seafastening, Lifting,** Pre-carriage, SeaPassage, Inspection
 - 상태: Planned → Ready → InProgress → Completed → Verified
- **Resource:**
 - **Equipment**(Crane, Forklift, Spreader, RiggingGear: sling/shackle/beam), **Workforce**(Rigger, Banksman, Operator)
- **Document:** StowagePlan, LashingPlan, StabilityReport, LiftingPlan, MS/JSA, P/L, B/L, Permit
- **Condition/M Measurement:** Weather, SeaState, Wind, Motion(accel g), Clearance
- **Organization/Agent:** OFCO, DSV, SCT, Client, Surveyor, Class
- **Constraint/Rule:** DeckLoadLimit, SWL/ WLL, ClearanceRule, RegulatoryRule
- **Time:** Instant/Interval(ETA/ETD, Shift), Milestone

말 그대로, "화물—작업—장비—장소—시간—문서—규정"을 모두 1개의 그래프에서 '연결'해 질문이 통과되게 만듭니다.

2) 핵심 관계(Object Properties)

- cargoLocatedAt(Cargo → DeckZone | StorageBay)
 - assignedTo(Cargo → Operation) / produces(Operation → Document)
 - securedBy(Cargo → RiggingGear) / performedBy(Operation → Workforce|Organization)
 - uses(Operation → Equipment) / occursAt(Operation → Location) / scheduledFor(Operation → TimeInterval)
 - constrainedBy(TransportMeans|Operation → Constraint)
 - hasMeasurement(... → Measurement) / hasStatus(... → StatusConcept)
-

3) 필수 데이터 속성(Data Properties) 예

- weight(kg|t), length/width/height(m), cogX/Y/Z(m)
 - deckStrength(t/m2), radius(m), swl/wll(t)
 - windSpeed(m/s), roll/pitch(deg), accelLong/Trans/Vert(g)
 - startAt/endAt(ISO 8601), docVersion, approvalState
-

4) 표준 연계(Interoperability)

- 단위: QUDT/UCUM (kg, t, m, deg, m/s²)
- 시간: OWL-Time (Instant/Interval)
- 측정/센서: SOSA/SSN (가속도, 풍속)
- 위치/좌표: GeoSPARQL (DeckZone도 폴리곤/그리드로 모델링)
- 어휘/상태표: SKOS (작업상태/허가상태 코드셋)
- 근거성: PROV-O (문서가 어떤 작업/데이터에서 파생됐는지)

표준을 재사용하면 시스템 간 데이터 교환이 편해지고, 단위 오류를 줄입니다.

5) 규칙/검증(Constraints) — SHACL로 예시

- **Deck 접지압:** $\Sigma(\text{cargo.weight} / \text{contactArea}) \leq \text{deckStrength}$
- **Lashing 용량:** $\Sigma(\text{WLL} \times \cos\theta) \geq \text{designLoad} \times \text{safetyFactor}$
- **Crane 반경 SWL:** $\text{SWL}(\text{radius}) \geq \text{liftedWeight} \times \text{factor}$
- **Clearance:** $\text{cargo.height} + \text{grillage} \leq \text{allowableHeight}(\text{zone})$

규칙은 **SHACL**(또는 규칙엔진)로 선언해 “데이터가 들어오는 순간” 자동 검증하게 합니다.

6) 컴피턴시 질문(이 온톨로지가 반드시 답해야 할 질문)

1. 현재 선적안(버전 X)에서 **DeckZone A**의 총 하중과 접지압은 안전한가?
2. **LashingPlan #123**에서 각 슬링의 예상 장력과 WLL 대비 사용률은?
3. 반경 R에서 선정 크레인의 SWL이 리프트에 충분한가?
4. **COG가 높은 화물**만 필터해 추가 시추/보강이 필요한 후보는?
5. 오늘 야간(19:00–07:00) **필요 인력/장비**와 공석은?
6. **SeaState ≥ 5** 조건에서 가속도(g) 가정이 바뀌면 어떤 화물의 라싱이 불합격되는가?
7. 특정 B/L에 포함된 Cargo들의 **Stowage 위치/문서/승인 현황**은?
8. **OFCO/DSV/SCT** 각각 담당 작업과 책임 경계는 어디까지인가?
9. 마지막 승인된 **StabilityReport**와 실제 탑재 데이터(중량/VCG)의 차이는?
10. 적재 순서 변경 시 **크리티컬 경로/대기시간** 변화는?

7) 아주 작은 예시(Turtle)

@prefix bco: <https://example.com/bco#> .

@prefix time: <http://www.w3.org/2006/time#> .

@prefix qudt: <http://qudt.org/schema/qudt/> .

@prefix unit: <http://qudt.org/vocab/unit/> .

```
bco:CARGO_001 a bco:Cargo ;  
  bco:weight "18.5"^^qudt:QuantityValue ;  
  qudt:unit unit:T ;  
  bco:length "12.0" ; bco:width "2.4" ; bco:height "2.8" ;  
  bco:cogX "6.0" ; bco:cogY "1.2" ; bco:cogZ "1.4" ;  
  bco:cargoLocatedAt bco:DeckZone_A ;  
  bco:assignedTo bco:OP_Load_20251019 .
```

```
bco:OP_Load_20251019 a bco:LoadingOperation ;  
  bco:occursAt bco:Vessel_TRUE_Deck ;  
  bco:uses bco:Crane_80T ;  
  bco:scheduledFor bco:Shift_Night_20251019 .
```

```
bco:Shift_Night_20251019 a time:Interval ;  
  time:hasBeginning "2025-10-19T19:00:00+04:00" ;  
  time:hasEnd        "2025-10-20T07:00:00+04:00" .
```

8) SHACL 스케치(간단 아이디어)

```
bco:DeckLoadShape a sh:NodeShape ;  
  sh:targetClass bco:DeckZone ;  
  sh:sparql [  
    sh:message "Deck load exceeds allowable pressure." ;  
    sh:select ""  
      SELECT ?this WHERE {  
        ?this a bco:DeckZone ; bco:deckStrength ?limit .
```

```

{
    SELECT ?this (SUM(?w/?area) AS ?pressure)
    WHERE {
        ?cargo bco:cargoLocatedAt ?this ; bco:weight ?w ; bco:contactArea ?area .
    } GROUP BY ?this
    }
    FILTER (?pressure > ?limit)
}
""";
].

```

9) 운영 설계 팁(현장 맞춤)

- **ID 정책:** VSL_TRUE/ZONE-A/2025-10-19/LOT-xxx처럼 사람과 시스템이 같이 읽히는 URI/ID.
- **DeckZone 그리드화:** 2D 좌표계 기준(Origin, X fwd, Y port)과 격자 크기(예: 1×1 m)를 그래프에 저장.
- **문서-데이터 연결:** LashingPlan, StabilityReport를 **produces/validates** 관계로 작업/데이터와 연결.
- **버전/승인 추적:** PROV-O로 "누가, 언제, 무엇을" 승인/수정했는지 이력 관리.
- **상태어휘(SKOS):** Planned/Ready/InProgress/OnHold/Completed/Rejected 같은 컨트를 타워용 코드셋 고정.
- **규정 계층화:** SOLAS/IMSBC/AD Ports 규정을 Rule 노드로 선언하고, 작업/장비에 constrainedBy로 링크.

10) 시스템 아키텍처(간단 청사진)

- **Triple Store/Graph DB(RDF/OWL) + SHACL Validator**
- **Ingest 파이프라인:** CSV/Excel(화물, 장비, 인력, 스케줄) → 매핑(R2RML/ETL) →

RDF

- **Query API:** SPARQL endpoint + GraphQL façade(현장 앱/대시보드 용)
 - **Rule/Calc:** SHACL(SP), 파이프라인 계산(예: 라싱 각도/장력), 결과를 Measurement로 귀속
 - **문서화:** 그래프에서 최신 상태를 끌어와 Stowage/Lashing/Lifting/Logistics Plan 자동 채움
-

11) 지금 있는 데이터와의 핏

당신이 이미 관리하는 **화물/선박/장비/인력/환경/스케줄** 표는 그대로 쓰되,

- 열(Column)마다 **어떤 클래스/속성**으로 들어갈지 맵핑 테이블만 정하면 됩니다.
 - 이후부터는 "질문"이 곧 "SPARQL 쿼리"가 되고, 검증은 SHACL이 담당합니다.
-

12) 한 줄 요약

온톨로지는 **현장 데이터를 하나의 지식 그래프로 엮어**, "안전·용량·일정·책임" 질문에 즉답하게 합니다. 한 번 골격을 세워두면, 선적 변경·야간 교대·기상 변수 같은 **변동성**에도 빠르게 재검증·재생성할 수 있습니다. 엔진은 단순합니다. ****개념(클래스)**를 작게, 관계는 명확하게, 규칙은 선언적으로.** 그러면 일은 훨씬 덜 복잡해집니다.