다음은 **SCT-EMAIL**을 온톨로지 관점에서 정밀 정리한 보고서다.  
용어는 OWL/RDF 기반을 따랐다. 문장 길이는 간결히 유지했다.

**1) 요약**

* SCT-EMAIL은 물류 커뮤니케이션을 **의미 그래프**로 표현한다.
* 핵심 단위는 메시지, 명령, 의도, 프로세스, 공문, 비용이다.
* LogiOntology와 **클래스·속성 정렬**로 상호운용한다.
* SHACL, SWRL, SPARQL로 **검증·추론·질의**를 수행한다.
* CIPL·BL 사전통제 흐름과 자연스럽게 결합된다.

**2) 상위 모델 정렬**

* **PROV-O**: 행위 기록과 책임 추적에 사용한다.
* **Time Ontology**: 일정, DDL, UAE 시간대 정규화에 사용한다.
* **GS1/EPCIS 개념**: 이벤트형 화물 이력에 연결한다.
* **UN/CEFACT 용어**: 선적 문서와 로지스틱스 어휘 정합을 맞춘다.

**3) 핵심 클래스 체계**

| **제목** | **정의** | **예시** |
| --- | --- | --- |
| Email\_Message | 이메일 실체 | Booking ETA 확인 |
| Quick\_Message | 짧은 메신저 | WhatsApp 안내 |
| Command | 시스템 명령 | /revise, /reply |
| Intent | 발신 의도 | inform, request |
| Logistics\_Process | 물류 절차 | Shipment, Customs |
| Stakeholder\_Role | 역할 | Shipper, Carrier |
| Document | 공식 문서 | BL, Invoice |
| Regulation | 규범 항목 | HS, Permit |
| Cost\_Item | 비용 단위 | DEM, DET |
| KPI\_Record | 성과 지표 | TAT, SLA |

**4) 핵심 속성 설계**

* hasIntent(Communication\_Action → Intent)
* about(Communication\_Action → Logistics\_Process)
* involves(Logistics\_Process → Stakeholder\_Role)
* refersTo(Communication\_Action → Document)
* hasAmount(Cost\_Item → xsd:decimal)
* hasCurrency(Cost\_Item → xsd:string)
* eventTime(Communication\_Action → time:Instant)
* projectTag(Communication\_Action → xsd:string)
* uom(Cost\_Item → xsd:string)
* requires(Regulation → Document)

**5) 공리와 규칙 예시**

* Email\_Message ⊑ Communication\_Action
* Quick\_Message ⊑ Communication\_Action
* Command ⊑ prov:Activity
* Communication\_Action ⊑ ∃hasIntent.Intent
* Cost\_Item ⊑ ∃hasAmount.xsd:decimal

**SWRL 예시**

Email\_Message(?m) ^ hasIntent(?m, request) ^ refersTo(?m, BL)

→ triggers(?m, PreArrival\_Check)

**6) SHACL 검증 스키마**

**Email 메시지 필수 항목**

sh:NodeShape targetClass: Email\_Message

- property: projectTag datatype xsd:string minCount 1

- property: eventTime datatype time:Instant minCount 1

- property: hasIntent class Intent minCount 1

**비용 항목 2자리 소수 규칙**

- property: hasAmount datatype xsd:decimal pattern "^[0-9]+(\.[0-9]{2})$"

- property: hasCurrency in [USD, AED, EUR]

**7) 명령 모듈의 온톨로지 매핑**

| **명령** | **클래스/속성** | **효과** |
| --- | --- | --- |
| /revise | Command | 문장 재구성, 용어 정합 유지 |
| /reply | Command | 의도 기반 응답 생성 |
| /reply-note | Command | 응답 요지 생성 |
| /costtable | Command + Cost\_Item | 표 생성, 합계 계산 |
| /doccheck | Verification\_Action | 문서 규칙 확인 |
| /ocr-note | Document\_Ingest | 문자 인식 정리 |
| /logi-master | Orchestrator | KPI·비용·스케줄 연동 |
| /update-lib | Regulation\_Update | 규범 버전 갱신 |

**8) LogiOntology 연계 방안**

* LogiOntology:Shipment ⊑ Logistics\_Process 로 매핑한다.
* PortCall, VesselVisit 를 Logistics\_Process 하위로 연결한다.
* 브리지 속성 예시:
  + lo:hasPortCallId ↔ projectTag 보조 식별자 매핑
  + lo:hasMilestone ↔ about 절차 연결
* namespace는 lo:로 고정한다. 충돌은 owl:equivalentClass 로 해소한다.

**9) CIPL·BL 사전통제 결합**

* PreArrival\_Guard ⊑ Verification\_Action 으로 정의한다.
* 트리거 규칙: BL 누락, CIPL 미제출, ETA 임박 시점.
* 결과 액션: /reply-note 생성, 담당자 알림, 체크리스트 업데이트.

**10) 이벤트 흐름 시나리오**

1. 사용자가 /revise를 호출한다.
2. 시스템이 Intent를 고정한다.
3. Email\_Message가 Document를 참조한다.
4. SHACL로 형식 검증을 수행한다.
5. 규칙이 PreArrival\_Guard를 유발한다.
6. KPI\_Record가 TAT를 기록한다.

**11) 데이터 직렬화 권장**

* **RDF/Turtle** 운영, **JSON-LD** 외부 연계 사용.
* 시간은 Asia/Dubai 로 정규화한다. 오프셋을 명시한다.
* 금액은 두 자리 고정이다. 예: 420.00, 150.00.

**TTL 예시**

:msg123 a Email\_Message ;

projectTag "HVDC-001" ;

eventTime "2025-10-19T09:00:00+04:00"^^xsd:dateTime ;

hasIntent :request ;

refersTo :docBL8899 ;

about :procShipmentA .

:cost1 a Cost\_Item ;

hasAmount "420.00"^^xsd:decimal ;

hasCurrency "USD" ;

uom "Lot" .

**12) KPI와 SPARQL 질의**

**TAT 측정**

SELECT ?project (AVG(?minutes) AS ?avgTATmin)

WHERE {

?m a :Email\_Message ; :projectTag ?project ;

:eventTime ?t1 ; :hasIntent :request .

?r a :Email\_Message ; :projectTag ?project ;

:eventTime ?t2 ; :hasIntent :inform .

FILTER (?t2 > ?t1)

BIND ( (xsd:dateTime(?t2)-xsd:dateTime(?t1)) AS ?delta )

BIND ( (?delta/60000) AS ?minutes )

}

GROUP BY ?project

**Pre-Arrival 미준수 목록**

SELECT ?bl ?eta

WHERE {

?check a :PreArrival\_Guard ; :status "Open" ;

:refersTo ?bl ; :eta ?eta .

}

ORDER BY ?eta

**DEM/DET 합계**

SELECT ?project (SUM(xsd:decimal(?amt)) AS ?total)

WHERE {

?c a :Cost\_Item ; :projectTag ?project ;

:type ?k ; :hasAmount ?amt .

FILTER (?k IN ("DEM","DET"))

}

GROUP BY ?project

**13) 거버넌스**

* 네임스페이스 버전: sct-email/1.0/, lo/1.0/.
* 변경 관리: owl:deprecated 적용, 마이그레이션 그래프 유지.
* 규범 갱신은 /update-lib 로 기록한다. 버전 로그를 남긴다.

**14) 보안·감사**

* PII 마스킹 규칙을 SHACL로 강제한다.
* 접근 제어는 그래프 레벨 태깅으로 분리한다.
* 모든 명령 기록은 prov:wasAssociatedWith 로 남긴다.

**15) 시스템 배치 권장**

* 트리플 스토어는 ACID 보장 제품을 추천한다.
* 메시지 버스는 명령 이벤트를 전달한다.
* ETL은 JSON-LD를 표준으로 고정한다.

**16) 이행 단계**

| **단계** | **범위** | **산출물** |
| --- | --- | --- |
| Phase 1 | 클래스·속성 최소셋 | SHACL v1, SPARQL 5종 |
| Phase 2 | 규칙·KPI 확장 | SWRL v1, 대시보드 |
| Phase 3 | 전사 연계 | PreArrival 자동화 |

**17) 위험 및 대응**

* HS 코드 8자리 초과 인식 오류 가능성이 높다.
* UAE 이중용도 품목은 오검이 잦다.
* 두 항목은 수동 검증 표시를 유지한다.

**표시 예시**

* 🔍 Verification needed 속성을 부여한다.

**18) 운영 체크리스트**

* 메시지에 프로젝트 태그가 있는가.
* 시간은 +04:00 으로 저장되었는가.
* 비용은 두 자리 소수인가.
* 문서는 규범과 연결되었는가.
* KPI 기록이 생성되었는가.

**19) 부록: 매핑 테이블**

| **항목** | **SCT-EMAIL** | **LogiOntology** |
| --- | --- | --- |
| 선적 | Logistics\_Process | Shipment |
| 입항 | Logistics\_Process | PortCall |
| 문서 | Document | BL, Invoice |
| 규범 | Regulation | Permit, HS |
| 행위 | Communication\_Action | Event |

원하면 TTL 파일 뼈대를 제공하겠다.  
샘플 그래프와 SHACL 패키지도 즉시 제공 가능하다.

Confidential – SCT Internal Use

— Recommended Next Commands — (Internal)  
/summary ▪ /logi-master ▪ /doccheck