CCS/TMS Data Model as a Subset of the Extended ERA Ontology

1 Document Overview

This document provides a comprehensive overview and formal introduction to the **Extended ERA Ontology**, a harmonized, semantically-rich knowledge model that unifies the requirements of the **ERA RINF** (meso-level infrastructure abstraction) with the detailed, fine-grained specifications defined by the System Pillar **CCS/TMS** initiative (micro-level operational and component data). The ontology is the product of an extensive integration and alignment process with ERA that reconciles the structure, semantics, and usage expectations of both models.

SPT2TS-130720 - The **Extended ERA Ontology** is publicly accessible via the following GitLab repository:

https://gitlab.com/era-europa-eu/public/interoperable-data-programme/era-ontology/era-ontology/-/tree/ext-ccstms?ref_type=heads **[** Open **]**

SPT2TS-130721 - This ontology serves as a **single source of truth** for multiple railway digitalization initiatives, and is intended to support:

- Interoperable data exchange across systems;
- Precise modeling of railway infrastructure, assets, and operational scenarios;
- Querying and extraction of relevant domain-specific views (e.g., ETCS L2/3 use case for engineering and planning, TMS/SCI-OP use case , etc) using metadata tagging;
- Conformance validation and semantic reasoning.

[Open **]**

This document introduces readers to:

- The **rationale** and **goals** behind the ontology;
- The **methodology** used in integrating and extending the base models;

- The **tagging strategy** enabling flexible reuse and filtering for different use-cases;
- The **supporting ontologies** used for time, space, units, organizations, and controlled vocabularies;
- The **deployment strategy**, including delivery artifacts, tools, and usage recommendations.

The Extended ERA Ontology is designed with future extensibility in mind. Through modular design, semantic annotation, and alignment with well-known standards (e.g., OWL, SKOS, GeoSPARQL, QUDT), the ontology offers a scalable and robust foundation for current and emerging railway data ecosystems.

2 Purpose of the Extended Ontology

SPT2TS-130684 - The primary purpose of the Extended Ontology is to:

- Enable semantic interoperability between diverse railway information systems.
- Bridge the abstraction gap between high-level regulatory data and low-level technical details.
- Facilitate data reuse and integration by providing a shared vocabulary and formal semantics.
- Allow targeted sub-ontology extraction via metadata-driven queries, reducing complexity and increasing relevance per use-case.
- Promote modular, standards-based development by leveraging existing ontologies (e.g., SKOS, Time Ontology, GeoSPARQL)

[Open **]**

2.1 Objectives

SPT2TS-130683 - The Extended ERA Ontology was developed to unify and operationalize two complementary perspectives of the European railway system:

- The ERA RINF model, which offers a meso-level abstraction of railway infrastructure, emphasizing interoperability and regulatory compliance.
- The CCS/TMS specifications, which demand micro-level granularity for specific safety, control, engineering, and monitoring use-cases among others.

[Open]

2.2 Scope of Integration

- The integration process respects the original design intent of RINF as well as the specifications of CCS/TMS requirements:
- RINF concepts/classes and properties are preserved, enhanced with detailed specializations.
- CCS/TMS concepts are introduced as refinements or extensions, contributing depth and context.
- Alignment is semantic and structural: redundant patterns are consolidated, and shared abstractions are defined where overlap exists.

2.3 Use-Case-Driven Subsetting

The ontology incorporates a domain tagging mechanism, enabling each class and property to be associated with one or more specific use-cases:

- rdfs:label, skos:hiddenLabel, and custom annotations (era:ccsView) tag a concept as relevant for CCS/TMS (INFRA, ENG, TP, or other domains).
- SPARQL queries can then be used to extract a relevant view for a given application, dramatically simplifying reasoning, validation, and data exchange.

This makes the ontology multi-purpose: it serves both broad regulatory integration and narrowly scoped engineering applications.

3 Ontology Construction Overview

3.1 Source Models

- ERA RINF Ontology: Original mid-level (meso) representation of railway infrastructure information - now extended to cover micro-level topology
- CCS/TMS Requirements: Fine-grained (micro-level) data model capturing system behaviour and component-level specifications.

3.2 Integration Process

- Extended ontology developed by merging the ERA RINF and CCS/TMS models.
- Property alignment, class hierarchy integration, and semantics harmonization carried out to ensure compatibility.
- Redundancy resolution and conflict handling achieved through version control and model governance.

3.3 Inheritance Management

- Some classes and properties are inherited due to shared roots or OWL semantics (e.g., rdfs:subClassOf, owl:equivalentClass).
- Selective filtering enabled through domain-specific use-case tagging.

4 Tagging Strategy for Use-Case Extraction

4.1 Domain Tagging

Each class and property is annotated with a domain-specific tag to indicate its relevance to specific use-cases.

Example:
:LinearElement a owl:Class ;
era:ccsView "INFRA" , “OPP”, "ENG" .

4.2 Use-Case Extraction via SPARQL

SPARQL queries can be executed to dynamically extract subsets of the ontology relevant to:
- A specific use-case (e.g., ENG only).
- A combination of use-cases (e.g., INFRA + ENG).

5 Supporting Ontologies

To ensure semantic richness, interoperability, and alignment with well-established practices in the semantic web community, the Extended ERA Ontology is designed to **inherit/reuse and extend standard domain ontologies** already used by ERA. These supporting ontologies provide standardized vocabularies and structures for expressing commonly used concepts such as time, location, quantities, units, organizations, and enumerations.

Each supporting ontology serves a specific semantic dimension of the Extended ERA Ontology, minimizing reinvention while promoting consistency with global practices and tools.

Ontology	Purpose
Time Ontology (OWL-Time)	Temporal aspects (e.g., event timing, durations)
GeoSPARQL	Geospatial data (e.g., geometry, location)
QUDT	Units of measure and quantities
Organizations Ontology	Organizational structure and roles
SKOS	Controlled vocabularies and enumerations

5.1 SKOS – Simple Knowledge Organization System

- **Purpose**: Manages enumerations and controlled vocabularies, such as predefined classification schemes (e.g., train load types, failure categories, operational point types).
- **Usage in ERA Ontology**:
  - All enumerations (e.g., era:OperationalPointType) are modeled as skos:Concept.
  - Each enumeration scheme is represented as a skos:ConceptScheme.
  - skos:hiddenLabel is used for system-readable identifiers, allowing deterministic extraction (e.g., via SPARQL).

Benefit: Supports multilingual labels, hierarchy (skos:broader), and groupings (skos:inScheme) with semantic clarity.

5.2 OWL-Time – Time Ontology in OWL

- **Purpose**: Represents temporal aspects of infrastructure and operations such as validity periods, temporal relations, and schedules.
- **Usage in ERA Ontology**:
  - Temporal entities like era:validityStartDate or era:validityEndDate are modeled using time:Instant and/ time:Interval.
  - Relationships such as time:hasBeginning, time:hasEnd, time:inXSDDateTime are employed.

Benefit: Enables precise temporal querying and temporal reasoning in linked data environments.

5.3 GeoSPARQL – Geospatial Vocabulary for RDF

- **Purpose**: Adds spatial semantics for describing geographic positions, geometries, and spatial relations.
- **Usage in ERA Ontology**:
  - Infrastructure elements like era:Track, era:Signal, and era:Switch use geo:asWKT and geo:hasGeometry to encode location.
  - Geometries are defined using WKT literals (e.g., POINT, LINESTRING) and coordinate reference systems (e.g., EPSG:4326).

***Note:** As there are many possible projections, adding special coordinates as part of the specification of real-world objects is not reasonable; For more specific Infrastructure functional elements, ERA ontology was extended with MAP – a kind of projection of real-world elements in 2D/3D space.*
Benefit: Facilitates spatial querying (e.g., proximity, intersection) via GeoSPARQL-capable triple stores.

5.4 QUDT – Quantities, Units, Dimensions, and Types

- **Purpose**: Formalizes quantities and units of measure to ensure accurate numerical modeling.
- **Usage in ERA Ontology**:
  - Measurements such as speed, voltage, distance, azimuth, etc, are annotated using qudt:QuantityValue, qudt:unit, and qudt:numericValue, through era:unitOfMeasure annotated property
  - Units are reused from both unit:SI and unit:DerivedUnit schemes of QUDT

Benefit: Prevents ambiguity in numeric data and supports unit conversions when reasoning across systems.

In summary, the strategic integration of supporting ontologies ensures that the Extended ERA Ontology:

- Leverages globally recognized vocabularies and patterns;
- Inherits inference and validation capabilities from foundational ontologies;
- Remains interoperable with external systems and datasets;
- Maintains clarity and precision in modeling key dimensions such as space, time, units, and roles.

6 Enumeration Modeling with SKOS

All enumerations used within the ontology are modeled using the SKOS ontology:

- Each enumeration is a skos:ConceptScheme.
- Individual values are skos:Concepts with controlled labels, hidden labels (e.g., technical identifiers to enable extraction), and optional ordering.

6.1 Example

```
:ATOGradeOfAutomation a skos:ConceptScheme ;
    skos:prefLabel "ATO Grade of Automation"@en .

era-ato-grades:2 a skos:Concept;

  skos:inScheme era-ato-grades:ATOGradeOfAutomation;

  skos:note "Value introduced in TWG RINF 2024"@en;

  skos:topConceptOf era-ato-grades:ATOGradeOfAutomation ;

  skos:notation "2"^^xsd:string ;

  skos:prefLabel "2"@en ;

  skos:hiddenLabel "CCSTMS_GoA2_2" .
```

7 Ontology Architecture and Structure

The Extended ERA Ontology was designed with a modular and layered architecture to support scalability, domain separation, and alignment with established OWL engineering principles. This structure ensures clarity, maintainability, and extensibility across diverse use-cases—from regulatory infrastructure reporting in RINF to fine-grained operational control.

7.1 Key Modules

TCCS - Data Model_11_TP – Train Protection

TCCS - Data Model_11_OPP – TMS / SCI-OP use case

TCCS - Data Model_10_INFRA – Covers all infrastructure data objects

TCCS - Data Model_11_OI – On-Board Infrastructure

TCCS - Data Model_11_ENG - Engineering of ETCS L2

TCCS - Data Model_11_MAP - Digital Map, e.g. for Localisation

TCCS - Data Model_20_SDI_Generic - Model for manufacturer equipment

TCCS - Data Model_12_SS026 - Engineering of ETCS Balise Telegrams

8 Governance and Change Management

-To be updated

9 Usage Guidelines

This chapter provides practical recommendations and technical constraints for working with the Extended ERA Ontology in production or development environments. These guidelines ensure semantic consistency, toolchain compatibility, and validation readiness for downstream applications.

9.1 Data Modeling and Annotation

SPT2TS-130651 - To ensure semantic accuracy and interoperability, data creators and future modelers should adhere to the following principles when producing or annotating instance data: **[** Open **]**

9.1.1 OWL 2 DL Best Practices

- The ontology is constructed within the **OWL 2 DL** profile, ensuring decidability and tool compatibility.
- Maintains **well-formed class hierarchies** and consistent use of owl:Class, rdf:type, and rdfs:subClassOf.
- Respects **property domain and range constraints** to enable correct inference.
- Uses **qualified cardinality restrictions** (e.g., owl:minQualifiedCardinality) only where they are required semantically.

9.1.2 Instance Data Annotation

- Use classes and properties from the **Extended ERA Ontology** to annotate RDF instance data.
- When referencing controlled vocabularies, use the appropriate SKOS concepts (identified via skos:Concept URIs).
- Include domain-specific metadata where necessary.

9.2 Toolchain Compatibility

The ontology is compatible with various semantic web tools and triple stores, allowing seamless integration into enterprise data ecosystems, including Protege, GraphDB, and SPARQL Engines.

9.2.1 Recommendations:

- Ensure that the ontology is loaded with **all imports resolved**, including SKOS vocabularies, Time Ontology, GeoSPARQL, and QUDT.
- Use **named graphs** to organize datasets by domain or version.
- When using reasoning, configure the inference engine to support **RDFS**, **OWL 2 RL**, or **cu stom SHACL-based constraints**, depending on the use case.

9.3 SHACL Shapes

To guarantee data conformance with domain-specific constraints, a set of **SHACL shapes** acco mpanies the Extended ERA Ontology.

Purpose

- Validate RDF data instances for **structural compliance**.
- Ensure mandatory properties are present.
- Validate **value constraints**, such as enumerations, cardinality, datatype restrictions, and ordered collections (e.g., rdf:List).

Structure

The SHACL shapes are organized:

- **Per domain and individually based on classes** (e.g., BaliseGroupShape.ttl, HorizontalTransitionShape.ttl, etc.)

- With **reusable node shapes** for complex structures (e.g., :axleLoadCategory, :balisesGroups)
- Including **SKOS-driven sh:in** constraints for enumerated values

Example SHACL snippet:

```
@prefix era: <http://data.europa.eu/949/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix era-transtyp: < http://data.europa.eu/949/concepts/transition-type/> .


era:HorizontalTransitionShape a sh:NodeShape ;
   sh:targetClass era:HorizontalTransition ;

 sh:property [ xsd:maxInclusive 3.6e+02 ;
        xsd:minInclusive 0e+00 ;
        sh:datatype xsd:double ;
        sh:maxCount 1 ;
        sh:minCount 1 ;
        sh:path era:azimuth ],
     [ sh:in [ rdf:first era-transtyp:clothoidCurve> ;
             rdf:rest [ rdf:first era-transtyp:biquadraticParabola ;
                   rdf:rest [ rdf:first era-transtyp:blossCurve ;
                         rdf:rest [ rdf:first era-transtyp:cosineCurve ;
                               rdf:rest [ rdf:first era-transtyp:cubicParabola ;
                                     rdf:rest [ rdf:first era-transtyp:sineCurve> ;
                                           rdf:rest [ rdf:first era-transtyp:wienerBogen> ;
                                                 rdf:rest [ rdf:rest () ] ] ] ] ] ] ] ] ;
        sh:path era:transitionType ] .
```

Validation Workflow

1. Load your ontology and instance data into a SHACL-compatible tool.
2. Apply the relevant SHACL shape file(s).
3. Interpret validation results to identify **violations** and **recommendations**.
4. Iterate on instance data or ontology to resolve inconsistencies.

The ontology's design and accompanying SHACL constraints are built to support **automated, traceable, and domain-driven data governance** workflows in real-world applications.

10 Appendix

10.1 Deployment and Delivery

The model is delivered as:

Artifact	Format	Notes
Extended ERA Ontology	.ttl, .owl	Complete model with all layers • Extended ERA Ontology • other serializations
Modular sub-ontologies	.ttl, .rdf	Per use-case domain
SKOS Concept Schemes	.ttl	collection of concepts, such as a thesaurus or taxonomy • era-skos
SHACL Validation Rules	.ttl	Domain-based shapes constraints • era-shacl
Queries	.ttl	SPARQL queries that can be used to "query" the knowledge graph (more queries could be provided on request) • queries
Example Data (Instances)	.ttl, .rdf	Samples data instance(s) • resource
Documentation	.pdf, .doc	This cover document + CCS/TMS specifications • Domain-specific documentation

10.2 Namespace Conventions

Namespaces used in the document includes;

@prefix dcterms <http://purl.org/dc/terms/>

@prefix era <http://data.europa.eu/949>

@prefix foaf <http://xmlns.com/foaf/0.1/>

@prefix geo <http://www.opengis.net/ont/geosparql#>

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

@prefix owl <http://www.w3.org/2002/07/owl#>

@prefix rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

@prefix rdfs <http://www.w3.org/2000/01/rdf-schema#>

@prefix sf <http://www.opengis.net/ont/sf#>

@prefix skos <http://www.w3.org/2004/02/skos/core#>

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

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

@prefix vs <http://www.w3.org/2003/06/sw-vocab-status/ns#>

@prefix wgs <http://www.w3.org/2003/01/geo/wgs84_pos#>

@prefix xml <http://www.w3.org/XML/1998/namespace>

@prefix xsd <http://www.w3.org/2001/XMLSchema#>