

TCCS - Data Model_00_Guide

1 Table of Contents

1	Table of Contents	1
2	Terms	2
3	Motivation for CCS/TMS Data Modelling	2
4	CCS/TMS in Extended ERA Ontology	3
5	Development History and Next Steps	4
6	Structure of CCS/TMS Model	10
7	Application of the Data Model	12
7.1	Identification & Referencing Objects	12
7.2	Referencing between split objects	13
8	Modelling Methodolgy	14

2 Terms


CCS/TMS Extended ERA Ontology (CCS/TMS Data Model)

The CCS/TMS Extended ERA Ontology and its derived CCS/TMS Data Model defines the harmonised data language for configuration, communication and diagnostics interfaces within ERJU System Pillar architecture. The Transversal CCS domain is responsible for the specification of the CCS/TMS Data Model in collaboration with

- the System Pillar domains, which apply the defined data structures in interface specifications
- the Innovation Pillar, which proves the applicability of the data model by demonstrators.

3 Motivation for CCS/TMS Data Modelling

SPT2TS-2037 - A harmonised digital CCS/TMS system requires a shared data language applied at all relevant interfaces with similar exchange items. Furthermore, it requires comprehensive configuration data for

implementing systems according to the System Pillar architecture (i.e. radio-based ETCS only). With the  **SPT2TS-2040 - CCS/TMS Extended ERA Ontology (CCS/TMS Data Model)**

the System Pillar provides a data structure for standardised engineering and to align a data structure for standardised interface specifications within CCS including CCS-TMS. This data structure shall be suitable across all relevant use cases of the System Pillar such as engineering and communication for traffic control or automated train operation.

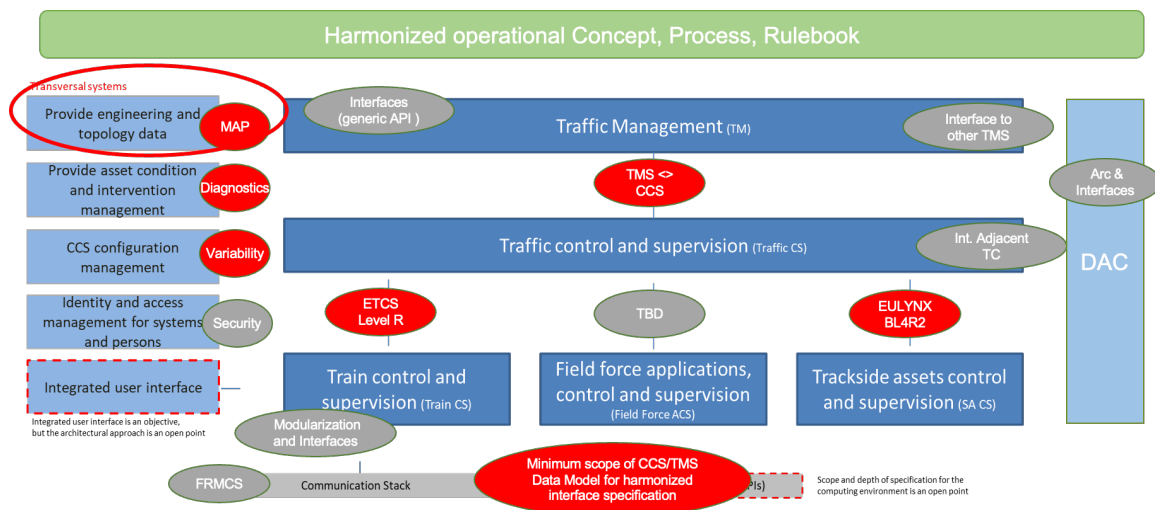


Figure 1 Interfaces between domains and minimum scope of CCS/TMS data model

The Data Model and interface specification must be comprehensive for SERA and therefore sufficiently detailed (not on a high conceptual level), at least on the Functional Interface Specifications (FIS) level within the System Pillar architecture. The development must be according to SEMP. The modelling approach is selected to support these goals.

While the TCCS domain is located in Task 2 of the System Pillar, it offers the inclusion of other tasks with the same and consistently applied CCS/TMS Data Model, as it is already applied for the interface between Task 2 and Task 3 TM.

[ Open]

For more details, see [TCCS - Data Model_01_Approach](#).

4 CCS/TMS in Extended ERA Ontology

SPT2TS-2052 - The CCS/TMS data language will be the base at least for ERJU specifications for communication interfaces, diagnostics and configuration. The CCS/TMS model specification is highly integrated into the System Pillar system engineering tool chain (Capella, Polarion,...).

Furthermore, the integration into the [ERA Vocabulary](#) as CCS/TMS extension is decided and performed for the purpose of legislation, long-term maintenance and semantical linking to further data models. Besides reusing semantic definitions and obtaining consistent use of terms within the railway domain, this approach also offers the possibility to benefit from existing links of other data models, which already are or will be connected with ERA vocabulary.

As a result of the integration, all further developments in the System Pillar regarding data models are reflected in ERA vocabulary. Official releases of the data model will always be derived from ERA vocabulary by an automatized process, as shown by the following process overview:

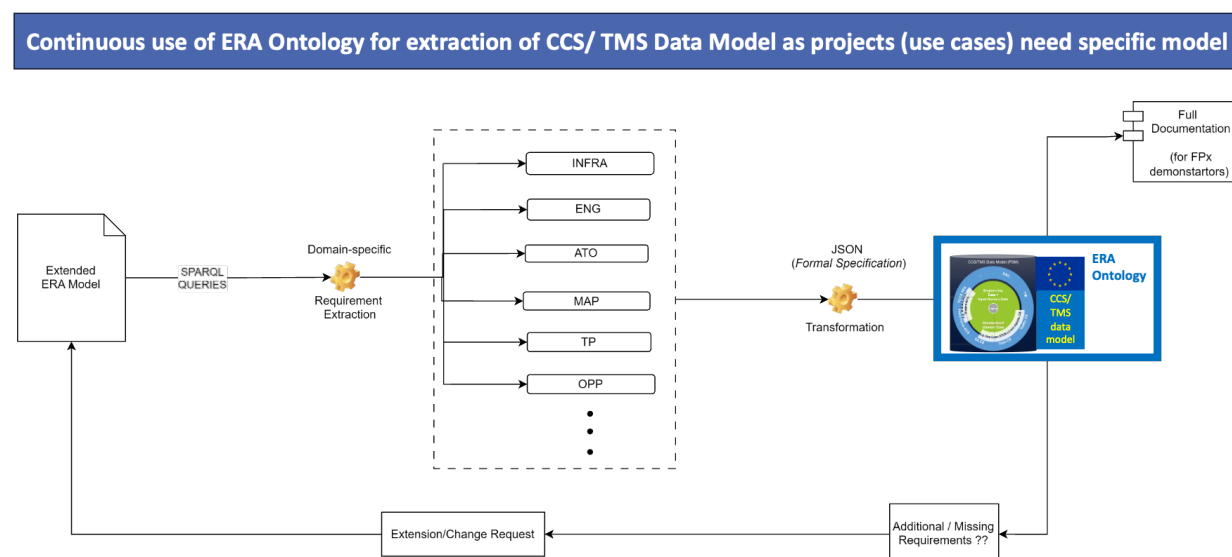


Figure 2 CCS/TMS Data Model derived from ERA Vocabulary to serve interface specification and demonstration needs

To support the specification process, the representation of the CCS/TMS data model in the toolchain of System Pillar (Polarion and Capella) is maintained by an automated toolchain, which keeps the connection to ERA vocabulary.

[ Open]

SPT2TS-130685 - 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-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]

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]






For more details regarding the CCS/TMS Data Model as ERA Ontology Extension it is referred to  [TCCS - Data Model_01_ERA Ontology Extension](#)











5 Development History and Next Steps

SPT2TS-122465 - Revisions of CCS/TMS Extended ERA Ontology and CCS/TMS Data Model














In the following table, the revision history of the CCS/TMS modelling developments (first as standalone ERJU development, now integral part of ERA Ontology) is documented, containing details about the release notes and the review activities. The feedback for one release leads to improvements within the next iteration of the data model and its documentation. Also, schemas are added or extended to cover more use cases of ERJU System Pillar. Besides the history, the planned releases are listed to give an overview of the next steps.










Data Model Version	Release Date	Release Notes	Publish and Review



Data Model Version	Release Date	Release Notes	Publish and Review
0.1	31.03.2023	<p>By the following documentation, we would like to inform you about the defined scope and mode of work of SP TCCS SD1, i.e. regarding the derived  SPT2TS-2040 - CCS/TMS Extended ERA Ontology (CCS/TMS Data Model) and the intended collaboration with the other domains/FAs:</p> <p> TCCS SD1 - Scope and Approach for Collaboration and Specification</p> <p>Attachments:</p> <p> TCCS SD1 - Data Model</p> <p> TCCS SD1 - Data Model Schema</p> <p>We kindly ask you to take note and provide comments in Polarion (or exported word file).</p> <p>Some additional contexts regarding the  SPT2TS-2040 - CCS/TMS Extended ERA Ontology (CCS/TMS Data Model): The data model initially focuses on the first layers (topology, geometry) and shall demonstrate the SD1 approach in a practical way. This paves the way and defines a base for further extensions in collaboration with you. Besides just providing clarification about the scope and approach of SD1, we also encourage (i.e. the FAs) to give feedback based on their implementation experiences.</p>	<p>The release (Polarion, PDF, JSON) has been shared via email and presentation with:</p> <ul style="list-style-type: none"> • All System Pillar Domain Leads for further distribution to responsible domain members • Innovation Pillar FA1 (incl. CDM) • Innovation Pillar FA2 (WP27) <p>The documentation was used for general scope & approach alignment rather than detailed model content discussion. The received comments (written form, discussions) are used to improve the explanations and structure of documents.</p>

Data Model Version	Release Date	Release Notes	Publish and Review
0.2	05.07.2023	<p>Version 0.2 of the  SPT2TS-2040 - CCS/TMS Extended ERA Ontology (CCS/TMS Data Model) has been updated with incoming needs from first use cases like ETCS, ATO, and Train Protection. In addition, the document structure has been modified to provide data model separation between different domains with sufficient linking between the domains.</p> <p>The  SPT2TS-2040 - CCS/TMS Extended ERA Ontology (CCS/TMS Data Model) version 0.2 contains the following set of documents:</p> <ul style="list-style-type: none">  TCCS SD1 - Scope and Approach for Collaboration and Specification (for additional information)  TCCS SD1 - Data Model_00_Release Notes (this document)  TCCS SD1 - Data Model_01_Introduction  TCCS SD1 - Data Model_02_Schema  TCCS SD1 - Data Model_10_INFRA  TCCS SD1 - Data Model_11_ATO  TCCS SD1 - Data Model_11_ENG  TCCS SD1 - Data Model_11_TP <p>We ask you to take note and provide comments in Polarion (or exported word file).</p>	<p>The release (Polarion, PDF, JSON, XML) has been shared with:</p> <ul style="list-style-type: none"> • All System Pillar Domain Leads for further distribution to responsible domain members • Innovation Pillar FA1 (incl. CDM) • Innovation Pillar FA2 (i.e. WP27, WP13, WP44) <p>Amongst others, the release with extended model content was used for the following collaborations:</p> <ul style="list-style-type: none"> • Discussion and model improvement together with FA2 WP13 and WP27 experts regarding data model needs for Moving Block demonstrator (TP, INFRA schema) • Discussion of needs regarding the onboard system (e.g. localisation) data needs with FA2 WP27 experts (INFRA Schema) • Discussion with engineering and tool development experts regarding data and model needs, i.e. for ETCS, Interlocking, and Infrastructure; including practical feedback from tool development supporting the data model for import/export. All feedback comments are used for model improvement (ENG/INFRA schema) to ensure applicability for implementation. • Check of ATO schema against actual ATO configuration data (i.e. Segment Profiles) <p>All relevant data model schemata have been commented on within Polarion (or by PDF comments).</p>

Data Model Version	Release Date	Release Notes	Publish and Review
0.3	31.08.23	<p>Correction release based on all comments of previous release. In addition, more clarification is given in the TCCS - Data Model_02_Methodology document regarding the structure and linking of data model schemas, identification and referencing (between data model parts and between instances).</p> <p>The SPT2TS-2040 - CCS/TMS Extended ERA Ontology (CCS/TMS Data Model) version 0.3 contains the following set of documents:</p> <ul style="list-style-type: none"> TCCS/TMS Data Model - Scope and Approach for Collaboration and Specification: updated i.e. regarding top-down / bottom-up development process TCCS SD1 - Data Model_00_Release Notes: this document TCCS SD1 - Data Model_01_Introduction: more details regarding modelling language, structure, linking, identification TCCS SD1 - Data Model_02_Schema: error corrections TCCS SD1 - Data Model_10_INFRA: processed feedback from model usage and review TCCS SD1 - Data Model_11_ATO: processed feedback from model usage and review) TCCS SD1 - Data Model_11_ENG: processed feedback from model usage and review TCCS SD1 - Data Model_11_TP: processed feedback from model review <p>In addition to the documentation, the data model for implementation in the formats JSON, XML and the JSON Schema will be published.</p> <p>The following use cases are in the focus of this release regarding test case scenarios (e.g. in collaboration with Innovation Pillar)</p> <ul style="list-style-type: none"> - ETCS/Balise Engineering - ATO Engineering 	<p>The release was used for SP Steering Group information in Sep 23, which led to an official sharing of the data model to Innovation Pillar for specification and demonstration purposes.</p> <p>In addition, the release was shared and discussed with ERA to clarify the connection to ERA vocabulary.</p>

Data Model Version	Release Date	Release Notes	Publish and Review
0.4	22.12.2023	<p>The  SPT2TS-2040 - CCS/TMS Extended ERA Ontology (CCS/TMS Data Model) is now extended with Data Model for CCS - TMS Interface (SCI-OP) and semantic linking for ERA. In addition, corrections based on all comments of the previous release were performed.</p> <p>The  SPT2TS-2040 - CCS/TMS Extended ERA Ontology (CCS/TMS Data Model) version 0.4 contains the following set of documents:</p> <p> TCCS - Data Model_01_Approach: processed feedback from model review</p> <p> TCCS - Data Model_00_Release Notes: this document</p> <p> TCCS - Data Model_02_Methodology: processed feedback from model review</p> <p> TCCS - Data Model_02_Schema: processed feedback from model review</p> <p> TCCS - Data Model_10_INFRA: processed feedback from model usage and review; added ERA Linking</p> <p> TCCS - Data Model_11_OI: processed feedback from model usage and review</p> <p> TCCS - Data Model_11_ENG: processed feedback from practical model usage, comparison against other engineering model, and review; added Fouling Point and other details required for engineering phase.</p> <p> TCCS - Data Model_11_TP: processed feedback from model review ; added Field Object Controller with first details as required by first demonstrators</p> <p> TCCS - Data Model_11_OPP : new schema to cover needs of CCS-TMS interface. The current state is aligned with Concept_Interface_TMS_CCS_V1_2 and will be further developed in collaboration with TMS (and Traffic CS) in 2024</p> <p> TCCS - Data Model_11_MAP: Map data model (coordinates) has been extracted out of INFRA as a new domain.</p> <p>Not included: the connection to the diagnostic data model as developed by the responsible taskforce has not been included yet due to needed alignment with other model parts. The plan is to include it in the next release.</p>	<p>Besides the release of documents (Polarion and PDF in SP Open Share), the automatically created data model files (json, xml/xsd,...) are provided for further review and usage within specification and demonstration.</p> <p>Links to SP Open Share:</p> <ul style="list-style-type: none"> • Data Model Files (json) • JSON Schemata • UML diagrams / UML pdf version • XSD Files <p>In addition, based on the alignment process started with ERA, a semantic version (TTL files) of  SPT2TS-2040 - CCS/TMS Extended ERA Ontology (CCS/TMS Data Model) is provided for this release, including linking to ERA vocabulary:</p> <ul style="list-style-type: none"> • CCS TMS Data Model Semantic (ttl) <p>The model will further evolve during the further collaboration with ERA.</p>

Data Model Version	Release Date	Release Notes	Publish and Review
0.4.1	04.04.2024	<p>This is a minor release of the  SPT2TS-2040 - CCS/TMS Extended ERA Ontology (CCS/TMS Data Model) with an update of the data model for CCS - TMS Interface as well as further improvements based on inputs from Innovation Pillar FA2 Demonstrators.</p> <p>Updated documents:</p> <ul style="list-style-type: none">  TCCS - Data Model_10_INFRA: integrated feedback from IP FA2 Moving Block Demonstrator  TCCS - Data Model_11_OI: minor changes to the data model  TCCS - Data Model_11_ENG: Replace mileage by Km-Signs.  TCCS - Data Model_11_TP: integrated feedback from IP FA2 Moving Block Demonstrator  TCCS - Data Model_11_OPP : integrated feedback from Task 3 CMS/TMS and added upstream data model objects. 	<p>Besides the release of documents (Polarion and PDF in SP Open Share), the automatically created data model files (json, xml/xsd,...) are provided for further review and usage within specifications and demonstrators (IP FA1, FA2, FA3, FA5, and FA6).</p> <p>Links to SP Open Share for XML, UML, JSON, and PlantUML exports: results</p>
0.4.2	28.06.2024	<p>This is a minor release of the  SPT2TS-2040 - CCS/TMS Extended ERA Ontology (CCS/TMS Data Model) with following changes:</p> <ul style="list-style-type: none">  TCCS - Data Model_11_OPP : integrated feedback from Task 3 CMS/TMS and IP FA2 Moving Block Demonstrator  TCCS - Data Model_11_TP: integrated feedback from IP FA2 Moving Block Demonstrator <p>Amended certain naming conventions in the data model to make the data model compatible for ERA integration.</p>	<p>Besides the release of documents (Polarion and PDF in SP Open Share), the automatically created data model files (json, xml/xsd,...) are provided for further review and usage within specifications and demonstrators (IP FA1, FA2, FA3, FA5, and FA6). In addition the ontology version for ERA integration is published.</p> <p>Links to SP Open Share for Ontology, XML, UML, JSON, and PlantUML exports: https://eeigertms.sharepoint.com/sites/SPOpenShare/Gedeelde%20documenten/Forms/AllItems.aspx?FolderCTID=0x0120009759543C5D980E4786FE7B845BBBAF5A&id=%2Fsites%2FSPOpenShare%2FGedeelde%20documenten%2FGeneral%2F24%2D06%2D28%20TCCS%20CCS%20TMS%20Data%20Model%20v0%2E4%2E2&viewid=7d2094bc%2D4ed7%2D4bd4%2D8055%2Dcda6a753bac3</p>

Data Model Version	Release Date	Release Notes	Publish and Review
1.0	04.09.2024	<p>Final release of CCS/TMS Data Model for remit period SC2.3</p> <ul style="list-style-type: none"> • Improvements based on Innovation Pillar demonstrator feedback • Alignment with domains and already defined interfaces - including feedback from domain approval • Improvements after Mirror Group Approval • added schema  TCCS - Data Model_20_SDI_Generic, imported from already approved Equipment Model from EULYNX (partial transfer based on needs, to be extended by other aspects according user needs during next remit period, i.e. for OC configuration and diagnosis) 	As usual, all model files are created for this release. The files are published on ERJU GitHub and Polarion.
1.1	21.03.2025	<p>Minor release of the CCS/TMS Data Model in framework of remit SC2.4</p> <ul style="list-style-type: none"> • First data model release that is aligned with ERA Ontology Extension, see extended-era-ontology Includes naming changes to all domains keeping the naming consistent with ERA Vocabulary. • New Domain On-Board Infrastructure (OI) has been introduced. OI now the static on-board infrastructure data needs for systems like Perception, ASTP, ATO, etc. • Includes updates data model classes in TP and OI domain based on needs coming Innovation Pillar 	As usual, all model files are created for this release. The files are published on ERJU GitHub and Polarion  CC S/TMS Data Model .

[ Open]

6 Structure of CCS/TMS Model

SPT2TS-122470 - The CCS/TMS data model follows a compositional tree structure, wherein the root of the model is a composition of several packages that constitute the CCS/TMS data model (e.g., Infrastructure, Restrictions, Rolling Stock, etc.), as shown in the figure below. Every package can be divided into sub-packages (e.g., TopologyArea, GeometryArea, etc.) depending on its needs. The figure below depicts the CCS/TMS model structure with the current included content:

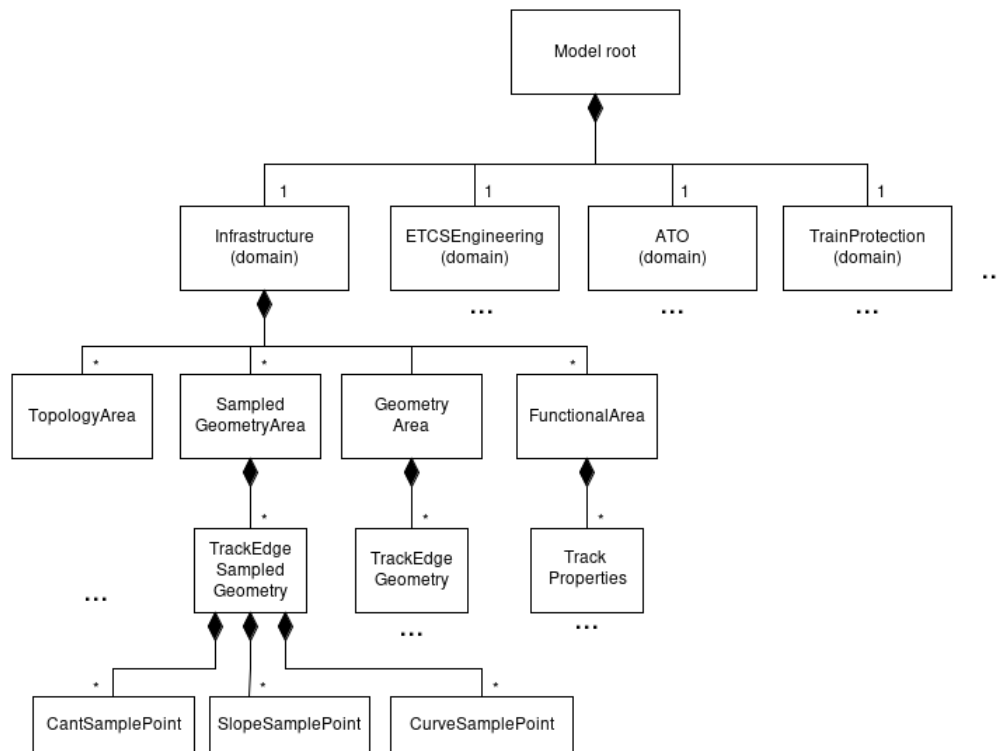


Figure 3 Overview of a compositional tree in data model.

[Content to be approved]

Rationale for compositional tree: [SPT2TS-123593 - Discussion regarding selected model structure](#)


SPT2TS-122471 - Constructing model packages should avoid cyclic dependencies, so the packages build a layer-hierarchy: upper layer-packages use (refer to) classes and enumerations specified in lower-layers-packages. The referencing is done via the point notation: `PackageId.class` or `PackageId.enumeration`. E.g. a `DangerPoint` in package `Engineering` refers to the `TrackEdgePoint` struct defined in package `Infrastructure`:

```
{
  "name": "DangerPoint",
  "attrs": [
    { "intId": 1, "name": "id", "dataType": "string", "key": "local" },
    { "intId": 2, "name": "name", "dataType": "string" },
    { "intId": 3, "name": "trackEdgePoint", "composition": "infra.TrackEdgePoint" }
  ]
}
```

[Content to be approved]

Regarding more details and discussions it is referred to [TCCS - Data Model_02_Methodology](#).

7 Application of the Data Model

This chapter gives guidance how to identify and reference objects as instances of the  SPT2TS-2040 - CCS/TMS Extended ERA Ontology (CCS/TMS Data Model). After describing the general referencing/identification principles, the referencing between objects that carry information from different packages is explained.

Regarding the underlying discussions it is referred to  TCCS - Data Model_02_Methodology

7.1 Identification & Referencing Objects

SPT2TS-122473 - Recommendation: Instead of referencing objects via data management containers (document in XML, table in SQL database, namespace in a single OPC-UA-Server) the referencing is done via the data model itself by providing the object-position inside of the objects-tree defined by composition relations:

```
trackEdge="/infra/topoArea[area51]/trackEdges[edge21]"
```

Besides an absolute path the object position can be also referred in relative way (e.g. "trackEdges[edge21]" only) if it is still unambiguous due to pointing within the same area.

All the aspects in the address are parts of the data model itself and are agnostic to the data management technology:

```
{
  "prefix": "infra",
  "intId": 1,
  "structs": [
    {
      "name": "infrastructure", "moduleContainer": true,
      "attrs": [{"name": "topoAreas", "composition": "TopoArea", "intId": 2, "multiplicity": "*", ...}]
    },
    {
      "name": "TopoArea",
      "attrs": [{"name": "trackEdges", "composition": "TrackEdge", "intId": 3, "multiplicity": "*", ...}]
    }
  ]
}
```

Having a layered and flexible model structure requires fine-grained objects with MANY references. It is assumed, that references would consume a considerable amount of the bandwidth. To reduce them a compact reference representation is proposed, where the attribute-names are replaced by integer-IDs also specified inside of the data model in "intId"-attributes:

```
trackEdge="/1/2[area51]/3[edge21]"
```

The last open question is: "how to select an object from the array-attribute?"

Here we follow the XPath approach - it can be selected by one of two possibilities:

- By a zero-based-index: /1/2[#32]/3[#211]
- By the key-attribute-value: /1/2[area51]/3[edge21]. In this case the class to be selected must have one attribute marked as "key": true. Normally it is an "id"-attribute of type string. To allow separation of index-based vs. key-based referencing, the keys must not start with '#'-character.

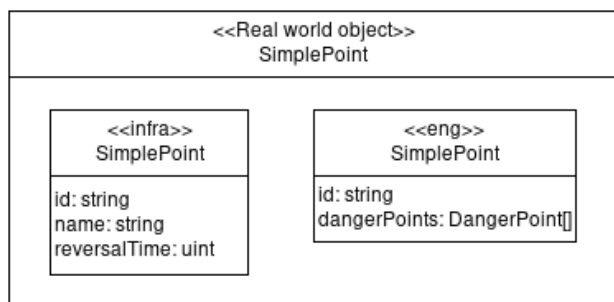
Both methods are applicable. For clear separation by the parser the '#' should only be used vor index based referenced and are forbidden in key values. In addition, the slash '/' is not allowed in references or keys, to allow speration of absolute paths from relative object positions or other ways of referencing (like GUID), which are also supported by the model, but not recommended due to the reasons mentioned above.

[🗑 Deleted]

7.2 Referencing between split objects

SPT2TS-122469 - The data model is structured in packages/layers covering specific aspects of the reality. In this situation one real-world-object is splitted in several data-objects representing various aspects of it, e. g.

- There is a "real-world-object" SimplePoint,
- Infra-package defines a functional data-object infra.SimplePoint covering name and transversal time important for operations,
- Engineering-package defines also a data-object eng.SimplePoint, adding danger-points to it, important for interlocking programming.

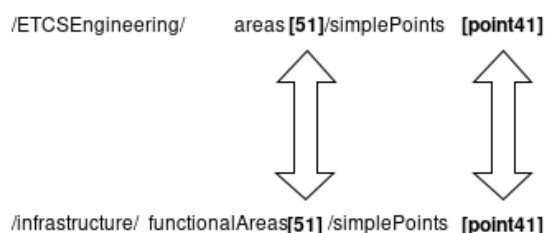


A typical Use-Case for a reading application is to combine several aspects of the physical object in memory. For this purpose it would need to retrieve several data-objects from the data storage and **link** them together. [🗑 Content to be approved]

SPT2TS-122476 - Splitted objects are linked to each other by **double-link-id-notation**. In case of the double-link-id-convention the real-world-object can be split in many parts and be constructed very effectively, as the reading application can deduce other parts from the current data-object automatically, e.g. knowing the reference to eng.SimplePoint it is easy to construct a reference to infra.SimplePoint:

/ETCSEngineering/areas[51]/simplePoints[point41] -> /infrastructure/functionalAreas[51]/simplePoints[point41]

The same is valid for the opposite direction:



This kind of dependencies are annotated in the model inside of key-attributes (here is a simplified notation from two different packages), e.g.

```
[
  {
    "name": "infra.SimplePoint",
    "attrs": [
      { "name": "id", "dataType": "string", "key": "global", "sameKeyAs": "eng.SimplePoint" }
    ]
  },
  {
    "name": "eng.SimplePoint",
    "attrs": [
      { "name": "id", "dataType": "string", "key": "global", "sameKeyAs": "infra.SimplePoint" }
    ]
  }
]
```

The "sameKeyAs"-dependency is transitive: a developer can follow the dependency-chain to identify all aspects of a real-world-object inside of the data-model. [🔒 Content to be approved]

8 Modelling Methodology

SPT2TS-2043 - To provide a model that is unambiguous and very close to implementation, the SD1 decided to use a formal, textual specification including meta-information such as semantics and linking to source data models. Based on the experiences in former modelling or implementation projects, a JSON notation has been selected and described as schema [TCCS - Data Model_02_Schema](#). The data model itself is explained and documented in [TCCS - Data Model_02_Methodology](#). The definitions of this document are automatically exported into technical schema files (e.g. XML/XSD, json,...) that can be used for the first implementation, demonstration and testing purposes to support a high grade of maturity even in early development stages. [🔒 Content to be approved]

The following excerpt from [TCCS - Data Model_02_Methodology](#) gives an impression of the chosen JSON notation (Example: Topology):

SPT2TS-1697 - Formal specification "Topology":

```
{
  "structs": [
    {
      "name": "TrackEdge",
      "info": "A track edge is an uninterrupted stretch of railway track, without divergence or convergence.",
      "belongsToSubPackage": "topology",
      "see": "http://ontorail.org/rsm12/Common/Topology/LinearElementWithLength",
      "attrs": [
        { "intId": 1, "name": "id", "dataType": "string", "key": true, "info": "Identity of the track edge, needed for referencing",
        { "intId": 2, "name": "name", "dataType": "string", "info": "User-friendly name. Empty string, if equal to the id",
        { "intId": 3, "name": "length", "dataType": "uint32", "units": "m", "exp": -3,
          "info": "Distance along the TrackEdge's 3D-alignment. Use zero if not defined."
        ]
      ]
    }
  ],
}
```

```
{
  "name": "TrackEdgeLink",
  "info": "Defines a relation between two track edges along which a train can run.",
  "see": "http://ontorail.org/rsm12/Common/Topology/PositionedRelation",
  "belongsToSubPackage": "topology",
  "attrs": [
    { "intId": 1, "name": "id", "dataType": "string", "key": true, "info": "Identity for referencing, e.g. by points and crossings"},
    { "intId": 2, "name": "trackEdgeA", "reference": "TrackEdge", "info": "Connects to track edge A"},
    { "intId": 3, "name": "trackEdgeB", "reference": "TrackEdge", "info": "Connects to track edge B"},
    { "intId": 4, "name": "startOnA", "dataType": "boolean", "info": "True when linked to the start of track edge A, false when to the end"},
    { "intId": 5, "name": "startOnB", "dataType": "boolean", "info": "True when linked to the start of track edge B, false when to the end"}
  ]
},
{
  "name": "TopoArea",
  "info": "Defines a container for rail network topology",
  "belongsToSubPackage": "topology",
  "attrs": [
    { "intId": 1, "name": "id", "dataType": "string", "key": true},
    { "intId": 2, "name": "trackEdges", "composition": "TrackEdge", "multiplicity": "0..*", "sorted": true},
    { "intId": 3, "name": "trackEdgeLinks", "composition": "TrackEdgeLink", "multiplicity": "0..*", "sorted": true}
  ]
}
] [ Deleted ]
```

SPT2TS-2044 - The formal representation also allows automatic transformation into specific schemata for different applications (protobuf, xml/xsd, or further development/documentation (e.g. UML). The tools for transformations and resulting schemata will be provided as part of the next releases. [🔒 Content to be approved]

SPT2TS-1657 - The model is primarily documented, versioned and stored in Polarion. The model is exported with each release and parsed into the mentioned data schemata. These schemata allow unambiguous standardisation and fast integration into development tools or demonstrators with test data, i.e. Innovation Pillar. This practical usage of the model creates a short feedback loop so that a high level of maturity is quickly attained.

In addition, a translation into UML (e.g. plantUML, XMI) can be used for overall model visualisation, which is automatically created and always consistent with the model itself. The bridge to Capella (if available to Polarion) will be used to achieve a synchronised model view with the System Pillar architecture and all other domain stakeholders working with Capella.

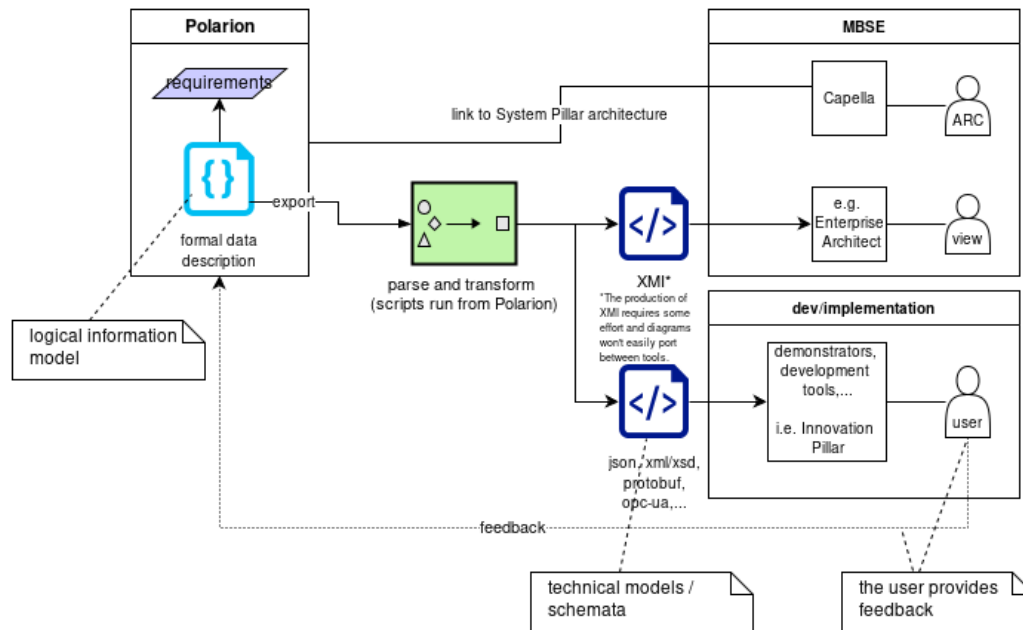


Figure 4: Handling the data model. The data model is a formal description of the object types and their relations. The ensuing technical information models are the schemata for transporting actual data over-the-wire or as files.

[🔒 Content to be approved]