RailCOMPLETE® Requirement Specification Editor for Wayside Electrotechnical Assets for Railways

2024-02-15, Rev 005

Subject areas, Processes, Actors and Objectives

Editor for Wayside Electrotechnical Assets for Railways

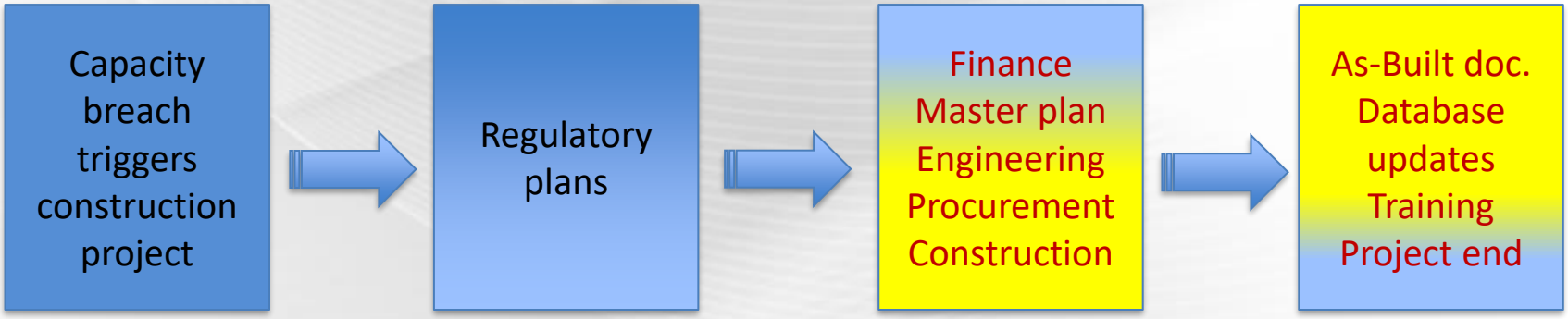
- This document summarizes the functional and specific requirements for an effective editor for railway engineering design.
- Affected disciplines are Substructure, Superstructure, Contact line, Low voltage, Signal, Telecom and Earthing.
- Affected parts of the process are Master plan, Detailed plan, Construction plan, FDV (facility documentation for management, operation and maintenance).
- Affected actors are:
- Administration (requirement setter and orderer)- Consultants (design unit)- Suppliers- Approval authority- Facility owner and its operating organization

Top level process map – railway engineering

- From capacity breach to new operational situation

Traditional process, ad-hoc tools

RailCOMPLETE® supported process



- Population growth
- Demography change
- Political pressure
- RU identified needs
- Dept. process
- National concerns

- Define project goals
- Engage consultant
- Suggest corridors
- Make initial drawings and assessments
- Start political processes

- Terrain, track, civil works
- Traction power system
- Signaling system
- Cabling, ducts, tubing
- Capacity assessment
- 3D
- Tables, lists, drawings
- Proc. / Constr. support

- Import survey data
- Update drawings
- Update 3D models
- Update operational databases
- Update archives
- Training, simulators
- Billing certificates

The players' overall goal with the editor

TIME: EFFICIENCY

- Centrally managed symbol and model libraries are developed only once
- Centrally managed methods are developed only once
- All repetitive data processing is automated

COST: SAVINGS IN PRODUCTION

- High level of detail and accuracy in the design documentation results in few errors during construction and eliminates "penalty rounds"

QUALITY: BIM and CONCURRENT DESIGN

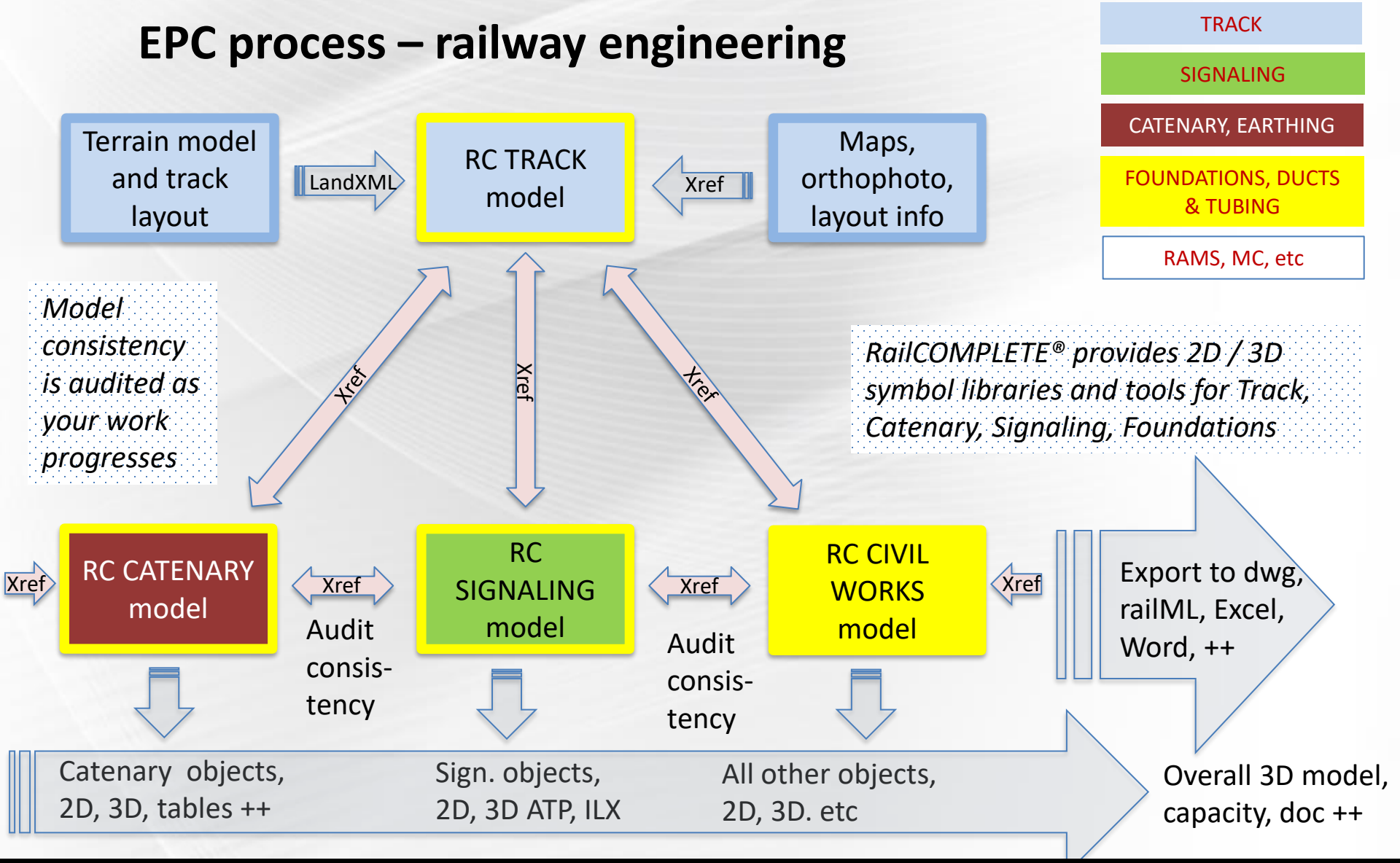
- Quality is ensured through interdisciplinary simultaneous coordinate-based design in a common model space, common data format
- "Single source of truth", i.e. data should be entered only once- Reuse of data at all stages- Seamless integration with existing tools and methods

Editor Main functions, Drawing types, Reports

Railway editor - Main functions

- Be a BIM (Building Information Modeling) collaboration tool to link planners, clients, consultants, contractors and end users closely together with a common data representation for track-bound technical facilities (infrastructure and its operation)
- Subject area support for Substructure, Superstructure, HV, Low voltage, Signal, Telecom
- Compatible with established workflows, data tools and data formats
- Expandable to new ways of working, data tools and data formats introduced in connection with ERTMS
- Support normal process where "the map is constantly changing", i.e. be able to effectively keep up with generic object descriptions and rules for engineering constantly changing (until weekly distribution of new definition)

EPC process – railway engineering



Railway editor - Main functions (2)

Visualize concrete and abstract data connections

Known symbolism
Objects must be represented in the tool with the symbols, models, concepts and designations that are already in use in the administration

Handle **multiple representations for object placement**
- Metric coordinate-based representation with mm accuracy
- 2D/3D position for a given map base (XYZ coordinate survey data)
- Alternative position representation in the form of track name, kilometer, lateral offset, height relative to the track (i.e. "rail positioning")

Visualization
- 2D symbolic view (microscopic level)
- 2D top view (nanoscopic level - construction drawings in plan)
- 3D schematic (*)
- Visualization of properties and relationships between objects

*) Partially in 2023.x

Railway editor - Drawing types and reports

The editor should be able to deliver:

- **2D drawings** at selectable scale with object tables, integrated with commonly used AutoCAD layouts (viewport, PUBLISH, etc.)
- **Schematic drawings** (*) that take their data from the BIM model
- **DWG interlocking tables**, Contact Wire pole table
- **Excel** Bill-of-quantity, Track data, Signage plan table, ATP Code table (**)
- Various **XML file formats**
- **3D geometry** with text annotations
- Report from **model validation** against rule sets

*) Partially in 2023.x
**) To be implemented in applicable DNA

Railway editor – Product structure

- **Generic Product** works the same for all administrations, packaged as one install file.
- **Generic Application** is the set of customization data, separated from the Generic Product. It contains data declarations and functions for a specific infrastructure manager, e.g. 2D symbols, 3D geometries, algorithms for calculation of braking curves, commonly referred to as DNA – **Definition of Network Assets**. DNAs are provided by Railcomplete Agents, which may be independent of Railcomplete AS.
- **Specific Application** data denotes a specific railway line design, using the Generic Product and a suitable DNA, saved as DWG files with embedded parts of the DNA in them, «the DNA molecule».

Editor Platform and file format

Railway editor - AutoCAD® platform and DWG

- The editor in its basic version shall be an AutoCAD® plug-in.
- Variants may become available: BricsCAD plugin, Ares Commander OEM version etc.
- The basic file storage format shall be DWG 2018 or later.
- BIM data shall be stored embedded in the DWG file format.
- All existing AutoCAD fonts, styles and API can be used.
- Newer versions of the editor shall be able to open, edit and re-save DWG files made with older versions of RailCOMPLETE, possibly involving conversion of deprecated items into their new formulations as BIM data.

Railway editor – Import and export, data storage

- Import and export **of alignment data** (geometry, vertical profile, cant profile, speed profile, mileage profile) from/to **LandXML**®.

- Import of geometry and vertical profile from DWG as 2D or 3D polylines.
- Import and export of **reference alignment definitions** from/to an open XML format.
- Import and export of **database filters** from/to an XML format.
- Import and export of DNA from/to an open XML format.

- The central part of a DNA is stored in the document's (DWG file's) **Global Extension Dictionary.**

- Object info (**BIM data**) is stored per object as Extension Dictionary containing serialized XML code.

Railway editor - Formats, import and export (2)

- Export of modelling data to railML® Infrastructure 2.3 (*).
- Export of modelling data to railML® Infrastructure 3.x (**).
- Export of modelling data to railML® Interlocking 3.x (**).
- Export modelling data to Excel, import from Excel.
- Export of annotated 3D models to DWG file format to AutoCAD compatible products:
  - **NavisWorks / Virtual Map / Trimble Connect**
- Export of annotated 3D models to IFC 2x3, 4.0 and 4.3 file format.
- BIM data shall be stored as an aggregate or as a distributed model, i.e., can be composed of multiple DWG files that include each other (XRef), with and without BIM data.
- Data may be spread across multiple files, viewing tools "pick up" objects' BIM data even if they are in an XRef.

*) Not all railML object types are available. The export function has not been certified at railml.org.
**) Planned function

Editor
Data objects

Railway editor – Data objects - Overview

- **Alignment objects**: Tracks, contact wires, guy wires, roads, etc.
- **Point objects**: Wayside railway equipment.
- **Area objects**: OCP areas, platforms, Temporary Shunting Areas etc.
- **Table objects**: Tables with embedded automation, either predefined in DNA or custom-built using the table editor.

Railway editor – Alignment object: Track

Track – inserted with dedicated commands, further edited with the Alignment Manager (AM), the Object Manager (OM) and the Property Manager (PM).

- Display alignment data (geometry, vertical profile, headroom), in table and graphically.
- Create executable track connections, even with "half-finished" track plan, per phase.
- Specify: track name, description, phase affiliation, layer affiliation.
- Kilometration, milestones and chain breaks: Value at start, indicate direction of increasing kilometers, specify “jumps” in mileage value.
- Set a track’s kilometration reference affiliation: Use a railway track, or any alignment type, as reference.
- Specify track system: Rotation axis when cant is non-zero: Rotate around center track or lowest rail, specify lift with cant or not. (*)
- Absolute Linear Addressing: Alignment has start value and increases monotonically between milestones / chain breaks.
- Relative Linear Addressing: Positions are stated as offsets from the closest preceding referent (e.g., kilometration boards along the alignment) (**)
- Specify structure gauges (DNA may contain A-85, A-96, A-96T, A-C, etc.) (*)
- Specify the track's immediate surroundings (platform, bridge, tunnel, etc.) (*)
- Specify the track usage type (service track, main track, switching track, etc.)
- Specify rail type, sleeper type, rail fastening, guide rails, rail inclination, ballast type (*)
- Other alignment object types - wires, cable ducts, roads etc shall be possible (*)

*) Depending on DNA **) Not yet in 2023.x

Railway editor – Alignment object: Track (2)

- In **mathematical calculations**, stored alignment data are used, not the visible graphics.
- Draw new track geometry directly in the drawing (i.e. draw as 2D graphics, save the track as an alignment in the model).
- Convert 2D and 3D polylines to alignments, import track geometry from DWG polylines.
- Convert an alignment to another type of alignment.
- Break, Join or Reverse alignments.
- Changes to track geometry in modelspace are immediately reflected in stored metadata.
- Import alignments from LandXML 1.2 (*)
- Export alignments and reference alignments to LandXML
- Update existing alignments with new data from LandXML
- A Pattern-Recognition algorithm recognizes continuations, bifurcations and crossings and inserts suitable topological connection objects. The connector's object type depends on the type of alignment being treated, as per declarations in the applicable DNA.

(*) There are many editors that produce LandXML 1.2, but each one has its own "dialect" which may require adaptions in RailCOMPLETE to retrieve kilometration data, tags / annotations, etc.

Railway editor - Point object (2)

For each 2D symbol in the library (*), the editor shall handle:

- 2D display symbol in the drag-and-drop dialogue before and after insertion.
- "Snap" to defined track distance (e.g. axle counter=0.75m, Ai and balises=track centre, driving machine=1.75m, signal=3.5m, KL mast=4.0m) (*).
- Whether the object's orientation can be flipped/rotated with a hotkey during insertion.
- Which visible attributes (in modelling space) the object has, and their default values (*).
- What other attributes the object has, which are edited in Properties / Object Manager, and the default values for these when inserting a new object (*).
- Default layer assignment for objects, their graphics and their attributes.
- Ask-for-attachment to other object’s 3D insertion point (*).
- Ask-for-relation to other object(s) before terminating the insertion phase (*).

(*) Depending on DNA

Railway editor – Point objects: Railway equipment

- Point objects are INSERTs in AutoCAD® that are made robust against "tampering“: using AutoCAD anonymous blocks.
- Create new point objects such as balises, signals, buffer stops, points, turnouts, approach beams, catenary poles, cantilevers, section insulators, guy wires. 2D and 3D preview of object type (*).
- Each object is defined in DNA with a category affiliation that can later be sorted on in the viewing tools, and with an insertion dialog category to make it easier to find the right symbol when inserting (*).
- Automatic setting of selected attributes to typical values at insertion (e.g. height relative to Top Of Rail, associated 3D geometry, preferred layer name in AutoCAD) etc. (*).
- Automatic pick-up of alignment affiliation at object creation (*).

(*) Depending on DNA

Railway editor - Point object (3)

- All point objects shall have general properties such as name, description, phase, XYZ position for the insertion point, alignment affiliation (‘own alignment’), greyed-out display of own alignment's reference alignment, distance to own alignment / relative height in relation to TOR / linear address relative to own and reference alignment, 3D representation(s), 2D projection(s).
- The 3D representation shall include offset and rotation for each graphical component in relation to the overall object’s insertion point: XYZ offsets, Pitch / Roll / Yaw, selection of algorithm for automatic pole routing in 3D.
- 2D projections shall include a definition of a surrounding box and from what viewing angle the 2D virtual “photo” of the objects is to be taken.
- Point objects may have DNA-dependent custom properties (*). For example, a "Signal" object may feature several thousand variants (combinations of 1-, 2-, 3-light Hs, Fs, Ds, Mid, Ls, FKs, Z, enter/exit/block/inside).
- Similar point objects may in menus be divided into groups and any number of levels of subgroups.

(*) Depending on DNA

Railway editor - Point object (4): Virtual

- Point objects can be virtual, in the sense that they can have visible graphics without representing actual railway equipment.
- Example 1: End point and via points for train route / shunting route.
- Example 2: Driveable connection between tracks.
- Example 3: Label pointing to a position and indicating some selected properties, such as position and object name.
- Example 4: Table placed in modelspace or in a layout containing a filtered extract of objects with selected attributes.

Railway editor - Area object: Areas (*)

- Drawing of areas directly in modelling space, in the XY plane.
- Point objects and tracks are automatically assigned to all areas that "capture" the object completely (considering XY coordinates only).
- Areas have general properties such as name, description, phase affiliation, XRef, etc.
- Area type is selected from a DNA-defined drop-down list (e.g., contract parcel boundary, station boundary upervision, local release area, work area, radio coverage area, ATC surveillance degree area, viewport location in modelpace for production of pdf etc). Line types are defined in DNA.
- Area boundaries can be manipulated (add/remove/move vertex, switch per segment between straight line and circle curve).

(*) Depending on DNA

Editor User interface, Command Browser and DNA

Railway editor – Command Browser – User Interface

- **A Command Browser** shall allow for easy access to all commands and to the Help system explaining these commands.
- **Command Names** shall follow a consistent naming, all starting with '**RC-**' followed by an **Action** (a verb) and an **Object** (a noun)
- The **User Interface** shall be provided in **multiple languages** (*)
- Offer **selection of Infrastructure Manager and** one or more **DNAs** (**)
- Offer **migration of BIM models** created using legacy DNA to a newer DNA.

(*) English, German and French in RC version 2023.x

(**) Depending on DNA

Railway editor - User interface - Command, Script, GUI

- Editor shall have a Command Browser that facilitates retrieval of rarely used commands, displays help information, and from which the user can launch commands.
- Most common commands shall be presented graphically in a ribbon (a row of buttons appearing on the computer screen). The GUI should be consistent and intuitive.
- Ribbon shall give easy access to the Generic Product's version, the user's license level and valifity, and applicable version information for the current open document's DNA.
- Integrated **Help, FAQ, Weblinks** and **Tutorials** shall be easily accessed from the ribbon
- Major commands shall be adapted for execution inside a script running under either AutoCAD's scripting system or under RailCOMPLETE's scripting system.
- The editor shall log key information to the CAD system's log window.
- Detailed info shall be logged to the end user's machine and be read remotely in connection with remote support

Railway editor - User interface - Graphical

- The editor shall be graphically based and shall display BIM data.
- Data that is changed in one sub-tool shall provide immediate updates in other sub-tools.
- Full undo capability with ctrl-Z, back to the time the file was opened.
- Data relationships are displayed in permanent and transient graphics in modelspace (e.g. tracer). Graphs are to be displayed in pop-up windows.
- Transient graphics ("there and then" graphics) shall be permanently copied to the drawing using a dedicated “make permanent” command.
- Copy-Paste functionality shall work both at object level and on individual properties inside objects.

Railway editor - Project info, selection of DNA

- There shall be a command for «Start New RC Document».
- The editor shall allow the user to select than administration when starting a new document (*) (e.g., SNCF, Bane NOR, DB Netz, etc.) and save project-specific information (assignment number, phase, alternative, etc.) in the model.
- This loads management-specific definitions of network assets (DNA), i.e., detailed descriptions of how objects look and behave, their relationship to other object types, their default values and methods, as well as consistency / tolerance checking rules, specifically targeting the applicable object type at hand (*).
- The user shall be able to update the current RC document’s built-in network asset definitions (switch to a new version of DNA), or choose not to, when the editor detects that newer DNA is available.

(*) Depending on DNA

Railway editor - Migration of data to new DNA

- Each DNA version shall be labelled with a unique DNA IRI (international resource identifier) indicating the target infrastructure manager, the DNA's displayname, the DNA agent's ID and date/time.
- There shall be an embedded tool for defining a **DNA mapping** between any two DNA versions (being reasonably similar).
- The DNA mapping tool is intended to be used by a superuser or by a trusted agent who prepares new DNA versions on behalf of the applicable infrastructure manager, when needed.
- The DNA mapping tool shall define a from-to relationship that specifies how legacy object attribute values shall be migrated if its attribute changes name, and even if the object type changes name/category/variant/etc.
- DNA mappings should become available to all applicable end users as a new DNA is made available. It shall be possible to map from several legacy DNA versions to the new one, or users can migrate their BIM model's data in multiple steps from the file's saved DNA via an intermediate DNA version and then to the latest DNA version. On demand, a DNA agent or superuser can always create a mapping between any two DNA versions, should such a mapping be missing.

Editor
User interface
Main tools
(Managers)

Railway editor - User interface - Main tools

Main tools – the Managers

- Properties Manager
- Object Manager
- Filter Manager
- Table Manager
- Alignment Manager

Railway editor – Properties Manager

- The editor shall have a **Properties Manager** (PM), a window where the user can view and most attributes for one or more selected objects, somewhat similar to AutoCAD's Properties.
- Object(s) displayed in PM **match the selection set in modelspace**.
- If several objects are selected and they all have a common property (attribute), the name of this attribute is displayed. If all the selected objects have the same value for this attribute, the value is displayed. If they are not all equal in value, ***VARIES*** is displayed, as known from AutoCAD's GUI.
- **Numeric values can be adjusted** with +<value> and -<value>, individually or multiple objects at the same time. Also "*VARIES*+3" etc. are allowed (all adjusted +3).
- Combined with Ctrl,Shift,Alt will **manipulate in powers of 10** ranging from 10^{-3} to 10^{2}.
- In most places, **LUA formula can be used instead of a hard-coded value** (*).
- **Attribute categories can be hidden** / brought forward again to increase readability in PM.

- *) LUA is a compact, fast, powerful, functional language, see more about LUA in Table Manager.

Railway editor – Object Manager

- The editor shall have an **Object Manager** (OM), with a table-based view where attributes are displayed in columns and objects are displayed as rows.
- **Selected** objects in modelspace are clearly indicated in the OM.
- **Marked** objects in OM can be manipulated in many ways without having to select them in modelspace first.
- **Zoom-to-object(s)-in-modelspace**, Select object(s), **Single-step** through the list of selected objects with **auto-zoom** in modelspace.
- From modelspace, select object(s) and select "**Show in OM**".
- Auto-focus and zoom-to in modelspace when marking objects in OM.
- Modify the objects' properties directly in the table.
- **Copy/paste** of value, formula, with/without key in OM and to/from Excel.
- **Multiple Copy** - copy the same value to multiple selected cells.
- **Fill Series** - Copy with auto-incrementing of the data series.
- **Edit layer properties** (switch on/off, lock/unlock layers in modelspace) from OM.

Railway editor – Object Manager (2)

- View all objects or **filter** by main category and subcategory.
- Click on column header to set **sorting** and sort direction.
- Object-dependent display of **columns**:
  - **Union**: All attributes that one or more object rows have
  - **Intersection**: Show only attributes that all object rows have.
- Show/hide columns (**click away/retrieve** from drop-down list).
- General **search field**, search in all attributes (with or without wildcards).
- **Column filters**, define a separate search criterion per attribute.
- **Area filter,** set an area that thus "catches" objects.
- **Restriction filter**, select a range of cells and "zoom in" in OM.
- Use **advanced filter** (created with Filter Manager).
- **Export to Excel** based on filtered view in OM.
- **Insert table** in modelspace based on filtered view in OM.

Railway editor – Filter Manager

- Right-click in the category column in OM to launch **the Filter Manager** (FM) where you can **create new/edit existing object filters.**
- Define which objects to **include** and which to **exclude** from the filter search.
- Each **filter row** expresses an "and" association between the filter criterion per attribute (among attributes that are in the filter, this corresponds to a composite column search in OM).
- An object is included in the OM view if it fulfils at least one **inclusion row** and is not covered by any **exclusion rows**.
- Filters can be **saved to file** (proprietary XML format).
- Saved filter can be **retrieved from file**.
- **Preview the filter** result directly in Filter Manager.

Railway editor – Table Editor

- There shall be a **Table Editor** that is started from the ribbon, or by selecting an existing table object and choosing the “Edit Table” command or start a new table from a “Table” icon in the Object Manager.
- In the Table Editor, the user shall be able to select from **predefined tables** (from a DNA) or specify his/her own table.
- Any table can act as a **parent table** (a template for other tables) or a **child table** derived from a parent table.
- The table editor shall offer:
  - Table **layout** (header, data celles, numberingm, fonts, colors)
  - Table **wrapping** at a given number of rows
  - **Add, delete, move columns**
  - Specify **complex row sorting** criteria
  - Set column names, with **multiple header levels** and header cell merging
- Define **automated cell content** using the Lua programming language (*)
- **Select source objects for the table** based on available object types (from DNA), write a custom database object type filter, or create a custom data structure (list of records) to be used as the table row source.

*) See description of LUA elsewhere.

Railway editor – Table Editor (2)

- **LUA** is a **powerful computer language** that is used everywhere in the editor where a data value can be specified as a formula, i.e. a miniature computer program to extract a value from the model. LUA was originally defined by the Brazilian Petrobras R&D department.
- **Example:** "Distance from <x> to <y> = <z>", where x, y and z are retrieved e.g. as *x=Self.Name, y=Attachment.Name* and *z=math.abs((Attachment.RefTrackMileage - Self.RefTrackMileage))*
- Click on an existing table in modelspace/layout and select "Edit" to change the table's appearance or content using Table Manager.
- Click on the table and select **"Refresh" to re-evaluate the tabl**e. The table's column definition is unchanged, but the rows are deleted, computed and the table content is reformatted based on the current BIM data in the model (all cells' LUA formulas are re-evaluated).
- An unwanted table is **deleted as an ordinary AutoCAD** object and purged

Railway editor – Alignment Manager

- The editor shall have an **Alignment Manager** (AM).
- AM shows which alignments (tracks) are available in the model, both those in the model file and those referenced in.
- Edit and view of **horizontal geometry, vertical profile, cant**.
- Display and editing of **kilometration and reference alignments**.
- Display and detection of **topology** (possible track connections).
- Auto-zoom to alignment or to alignment segment in modelspace.
- **Pick alignments** from table in AM or directly in modelspace with pointing tool.
- **Copy** from table **to clipboard**.

Railway editor – Track Manager (2)

- Perform **track design analysis** (*).
- Set **tolerance limits** for track design - at an early stage of track design, e.g. vertical profile (height above sea level, gradients) and cant (track roll) are not defined for all tracks, and tangent directions and accurate placement of switches may leave much to be desired initially (*).
- **Default tolerance limits** for track design may be be given in DNA (*).
- **Ignore selected parts** of the track design, e.g. ignore profile mismatch since early tracks have not always been completely designed yet.
- Analyse **velocity dependent track design** rules (0.15*V etc) (*).
- Detect **segment sequence errors** (FKP vs radii, re-buffering etc) (*).

*) Only partially in 2023.x / depends on DNA.

Railway editor – Stage Manager

- The editor shall have a **Stage Manager** (SM).
- SM allows the user to specify a search pattern to find stage information in AutoCAD layer names, e.g.: "***MyObjectType_Phase_10.20-30.50***".
- SM **scans** the drawing and creates a **list of all Entering / Current / Leaving stage** options.
- Clicking on one or more stages in SM **switches the associated layers on/off** in modelspace.
- Layers in **XREF are also affected** when changing stage.
- SM can also be used in any **viewport** (*).
- **Settings** are remembered for modelspace and per viewport (*)

(*) Not in 2023.x / dependent on DNA

Editor
User interface
Visualizations
(Viewers)

Railway editor – User interface – Visualizations

Visualizations – ‘Show’ tools and ‘Display tools’

- Position – orient yourself in a model, show linear placement details
- Alignment Info (custom tooltip)
- Distance (custom tooltip)
- Annotations (geometry, vertical profile, alignment names, +++)
- Relation Lines
- Paths
- Signal Sighting (line, region, beam, volume)
- Loading Gauges – simulated train runs in 2D or 3D
- Envelope – traced outline of 2-axle coach running all possible paths
- Layer Groups (axle counters, long sleepers, insertion points, +++)
- 2D Projections
- 3D preview
- 3D Export (DWG, IFC)

- **SHOW** = immediate response / **DISPLAY** = parameters needed

Railway editor – Visualizations – Position

- Show **position** as a transient graphic in modelspace, a cross line perpendicular to a selected track, which moves with the CAD cursor and shows alignment name and kilometres for your own track at all times, possibly also for the reference alignment.
- Copy kilometre **position to clipboard** in Excel-friendly format.
- **Copy 3D position**, object orientation and object info to clipboard (*).
- Select **Snap** To Alignment mode from **Nearest or Bind**.
- In **Nearest mode**, info is shown for the alignment that is "picked up" closest to the CAD cursor.
- In **Bind mode**, info is shown for the alignment that you last selected in modelspace (or in other tools).
- **Alignment mileage direction** is displayed transiently
- From 2023.1, RC implements ISO 19148:2012 concerning dead zones.

*) Not in 2023.x / dependent on DNA

Railway editor – Visualizations – Alignment Info

- Show **static information about the track (Alignment Info)** in a window that updates according to the position of the CAD cursor.
- Choose between **Nearest mode, Bind mode** or follow a named path (see explanations elsewhere).
- Display of **position in own track**, instantaneous height, gradient, curvature, cant, wind skew, maximum permitted cant for slow moving trains (ref. DNA).
- Possibility to display the track and the two nearest tracks with transient indication in model space of the clearance gauges incl. inner/outer curve extensions, without cant extension (*).
- Display in profile of the track and two closest tracks with their instantaneous clearance profiles where the CAD cursor is located, including superelevations, curve extensions, superelevation extensions and added shunting gauge (*).
- When the track's superstructure quality class and other quality parameters are set, do continuous calculation whether the track is within the requirements, as a green / yellow / red field in Alignment Info (*) If the design speed is given in the alignment data, the resulting lateral force, maximum permissible cant, missing cant, jerk, ramp ramp speed are displayed (*).

*) Not in 2023.x / dependent on DNA

Railway editor – Visualizations – Distance

- Display **distance** as a transient graphic in modelspace, in the form of a hover-over tool that you move with the CAD cursor and which shows show at all times the distance from the CAD cursor to the nearest matching object on each side, as well as the distance between the two objects.
- Distance between a selected object and the CAD cursor is to be **measured along the shortest path** in the alignment graph.
- User can set **object type(s) and direction** (e.g. "from Fs to Hs in up direction") used to find matching objects on the low mileage side and on the high mileage side for the selected path.
- Distance is displayed optionally as measured along the path (along the alignment’s centre-line) or as measured in reference alignments

Railway editor – Visualizations – Annotations

- **Show annotations as transient graphics** in modelspace for selected alignment(s).
- **DNA-defined behaviour** for units, number of decimals, zoom behaviour.
- **Geometry annotation**: Characteristic points (SS, R, RL, OB, OE, KP, h).
- **Mileage annotation** – Mileage along own alignment, chainbreaks.
- **Mileage along reference alignment** - Mileage along own alignment only when it is "its own reference“.
- **Vertical profile annotation** - LBP, HBP, SE1, SE2, radius, gradient, etc.
- **Reference alignment annotation** - Shows parent (solid line) and child (dashed) in the same colour.
- **Track Agent Annotation** - RC uses an iterative method to find track agents for arbitrarily complex track geometries.
- **Topology** - Show tracks as branches, open ends, executable connections.
- **Alignment name annotation**
- **2-rail track visualization**.
- Right-click and select ***Copy annotations to drawing*** => creates a CAD block with “dead” graphics.

*) Not in 2023.x / dependent on DNA

Railway editor – Visualizations – Relation Lines

- See more on binary relations elsewhere in this specification.
- When the **Relation Lines annotation to**ol is active, clicking on an object will make **nice, curved lines** to appear, between the selected object(s) and the objects that it has been related to.
- A relation is **hidden** whenever the source or the target or both are invisible (on a frozen or OFF layer).
- Relation lines are **continuous** lines in the forward relation direction, and **dashed** in the reverse relation direction, as seen from the selected object(s).
- Relation lines are technically to be drawn as three tangentially connected splines in 3D.
- The relation line’s angle of exit / angle of entry reflects which alignment the source / the target belongs to.

*) Not in 2023.x / dependent on DNA

Railway editor – Visualizations – Paths

- Show **path as transient transparent graphics** over parts of the track network, while the CAD cursor is moved over the tracks.
- Paths may be used for **illustrative purposes**.
- Paths are useful as **input to a braking curve tool** or to an **Offset Alignment Profiles** command, from a source path to a target alignment.
- **Paths** may be used as the basis **for Alignment Creation**.
- **Generate paths from powerful API functions**, as part of graph-searching custom algorithms.
- **Paths can be named, saved** and used in Wizards (*).
- Possibility to **copy path to drawing** (*).
- **Copy or move** selected objects **along a path**.

*) Not in 2023.x / dependent on DNA

Railway editor – Visualizations – Signal Sighting

- **Sight line** draws a **flat, straight line from a wayside object** (signal etc.) **to the centre track** at a specified sighting distance.
- **Sight region** creates a **closed, flat, polygon** that appears as the **sum of all sight lines** from a maximum sighting distance to a specified minimum distance.
- **Sight beam** creates a **3D SOLID beam** from the **train driver's windscreen to the significant part of a target object**, from a specified sighting distance.
- **Sight volume** is the **3D sum of all sight beams** from a specified sighting distance.
- Useful in 3D viewers for a very accurate clash control and signal sighting assessments.
- A particular **“sighting target of interest"** (e.g., the optical lanterns in a larger signal rendering) **can be specified**.

Railway editor – Visualizations – Loading Gauges

- Represent **predefined loading gauges in DNA**.
- **Run a 2D virtual train coach**, a rectangular 2-axle body, though the track network and see the effects of inner and outer curve extensions.
- Select from **custom loading gauges (UIC-GC, A96T etc)** and run a 3D simulation of a moving 2-axle coach through your network. See it **split in two** when there is a **cant ramp**.
- Display Gauge tool can be run **in manual "mouse" mode** or in autonomous **"keyboard" mode**.
- Possibility to **copy virtual coach(es) to drawing** (*) – to illustrate parked trains of multiple types in an illustration – good for **platform planning**.

*) Not in 2023.x / dependent on DNA

Railway editor – Visualizations – Envelopes

- The editor **traverses all possible paths** using a **virtual 2D train coach**, a rectangular 2-axle body, though the track network and traces the resulting outline – the envelope – resulting from the coach's inner and outer curve extensions as it moves everywhere.
- Use envelopes to determine **where objects can be placed without disturbing train traffic's loading gauge** requirements.
- **Selectable trade-off** between accuracy and computing speed.

*) Not in 2023.x / dependent on DNA

Railway editor – Visualizations – Show Layer Groups

- There shall be a **Show Layer Groups** tool that allows the user to switch on/off named groups of object layers, or groups of layers for graphic sub-elements contained in objects' graphics.
- **Axle counter sensor example**:
  - Show / hide a shaded area in where the **nearest conflicting rail strand is 20 cm apart**.
  - Show / hide a reserved area for "mushroom" tuning unit.
- **Switch (turnout) example:**
  - Show / hide **Long sleeper** part of switch / Show **short sleeper** part of switch.
  - Show / hide straight leg's and deviating leg's **centerlines**.
- **Balise example:**
  - Show / hide **Metal-free areas**.
  - Show / hide 3m / 8m / 12m **help-lines to preceding / following balise**.
- **General usage**:
  - Show / hide **label** objects / object's **insertion points** / **auxiliary lines**
  - Show / hide Model check results: **ERRORS / WARNINGS / UNFINISHED / HISTORY / DEROGATIONS** (*)

*) Dependent on DNA

Railway editor – Visualizations – 2D Projections

- **Define 2D Projection parameters individually per object**:
  - Add any number of 2D projections as dynamic properties to a point object.
- Create **illustrations of an object** in its natural 3D context, the ensemble being **projected onto a plane** defined by the 2D Projection parameters contained in the source object.
- The 3D context to be specified as a **tilted 3D box around the source object** – everything that falls within the box is included in the projection.
- Fine-adjust box contents with **inclusion set** and **exclusion set.**
- **Automate 2D projections**, couple with layouts to create projections intended for repeatedly and conveniently updated **PDF installation drawings**.

Railway editor – Visualizations – 3D Preview

- Every point object or alignment object can have **any number of associated 3D geometries**.
- **Point objects** have point geometry, typically a single block / DWG file with one or more solids (such as an **optical signal**).
- **Alignment objects** can be used as basis for sweeping a 2D profile (such as **rails**) or for iterating a 3D geometry along the alignment (such as **sleepers**).
- When the **3D Preview mode** has been activated, each object will upon selection appear in 3D on top of its 2D symbol in AutoCAD modelspace.
- 3D preview is useful for **rapid assessment of local clashes**, e.g., a cable duct clashing with a catenary pole or signal.

*) Not in 2023.x / dependent on DNA

Railway editor – Visualizations – 3D Export

- **Define 3D modeling parameters individually per object**:
  - **Add geometries dynamically** / **cascade of coordinate systems** possible (e.g., place board-on-mast, then standing-text-on-board).
- **Automatic pole routing**: No need to design poles for signs and boards with bends, RC does that for you.
- Full control over block structure, layer structure, colors, block naming, layer naming.
- 3D export is used to **build large BIM models**, typically one discipline (signalling, cable ducts, tracks, catenary, telecom…) per partial 3D model.
- Output format selectable to **DWG2018, IFC 2x3 / 4.0 / 4x3**.
- Property Set Editor (*) and Spatial Structure Editor for IFC output.

*) Not in 2023.x / Dependent on DNA

Railway editor – Database – Automation

- **Attachments**
- **Dock points**
- **Binary Relations, Relation Refresh Depth**
- **Model checks**
- **The Lua Debugger**
- **Scripting with Lua**
- **External Libraries (DLLs)**

Railway editor – Database – Attachments

- In a DNA, each object type can have a (customizable) **AttachmentCategory**.
- For each object type, you define which other object types the object can be attached to (its attachment categories).
- The **Attachment** mechanism means that the attached object moves to the insertion point of its parent and assumes the same Own Alignment as its parent.
- Attachment is typically used for **objects that are to "stick" to the same 3D coordinates as their parent**, e.g. a sign that attaches to an optical signal, or a footplate that attaches to a concrete foundation.
- When **creating** a point object, the object may **ask for attachment** right away. This makes the user dialogue intuitive and efficient.
- A well-designed hierarchy of attachment categories is key to **good DNA modeling**. E.g., a given speed sign object type can be connected to a foundation or placed on top of an additional speed sign, but it may not be placed on a signal and definitively nor on top of an axle counter, etc.

Railway editor – Database – Dock Points

- Point objects shall be able to offer **dock points** to other, compatible, object types.
- **Dock-point mode of operation**: Whenever an object of type X is to be created, objects of type Y offering docking points (*) for objects of type X will display a circle at each offered location. Clicking inside such a circle makes the newly created object snap into position at the exact center of that dock point.

*) Dependent on DNA

Railway editor – Database – Binary Relations

- In DNA, each object type can have a SourceSpace. As soon as such a SourceSpace is declared, it can be used as a TargetSpace in binary relationships between objects.
- In DNA, it must be possible to define an arbitrary number of relationships for each object type.
- A "prompt" (query statement) must be defined for each relationship, whether you click on the Source object or the Target object.
- The relations should know how many instances it can have for a given object, as well as what should happen when the user "oversaturates" the object.
- When creating an object (see this), it should be configurable per object to ask to be related to another object there and then, or it can be done later. For example, a balise group can be related to "its" balises and to a position-defining balise, to an object that provides control information, etc.
- Transient display of relations should be available to be switched on/off and copied to the drawing.
- Relations can be utilised when searching in the database, in tables, in Lua formulas.

Railway editor – Database – Relations / Refresh Depth

- In the DNA you may express any number of binary relations as (*):

  a R b :: A x B |→ {true,false}

  where each of A and B are sets of one or more predefined object types, and where the relation, as seen from a towards b, defines whether a is related to b or not.

- Establish a relation by selecting object a, choose the applicable relation type 'R', then select object b to finalise aRb.

- **Examples:** *Pole1 HasNextPole Pole2* or *Sig1 IsDistantSignalFor Sig2*

- For each relation R there is automatically a **reverse relation** R^{-1} such that aRb ⇔ $bR^{-1}a$

- A reverse relation should be given a natural **reverse relation name**, e.g.: *HasPreviousPole* or *HasDistantSignal*. Note that some relations are by nature **its own inverse**: *IsCoupledWith* etc.

- Set the **Relation Refresh Depth** (RRD) from to 0 to 4. The RRD decides how far out, following a relation's relations' relations' … , an object refresh action will reach. High RRDs keep your model conveniently up-to-date but is computationally more demanding / results generally in more waiting times during editing. Turn RRD up and refresh your model well before delivery to customer!

*) Dependent on DNA – it is a massively customizable feature!

Railway editor – Automation – Model Checks

- An object declaration can contain any number of **Model Checks.**
- Each model check is **a Lua Program** that checks certain aspects of your model (which is the current database), starting out from the object containing the model check, and returning a **text string**, a **symbol** (_error / _warning / _ok) and a '**ping collection**'.
- The text string explains what the **test results** are.
- The symbol leads to a **visible graphical symbol** of the object in error (thick red ring) / warning (thinner orange ring) / ok (nothing or a checkmark) (*)
- The **Ping Collection** points to **peer objects that shall be model checked**.
- **Model checks**:
  - **Minimum 21 meter** distance between neighbor axle counters (*).
  - _**Unfinished** (light blue ring): Object is not fully parameterized yet (*)
  - etc

*) Dependent on DNA – it is a massively customizable feature!

Railway editor – Automation – Lua Debugger

- The **Lua editor appears in many situations**:
  - Editing a property for one or more objects using **Property Manager**.
  - Editing a property for one or more objects using **Object Manager**.
  - Insepecting one or more objects – or the drawing database itself – from the **Query Object** tool.
  - Editing a text attribute property (relative 2D position, text rotation) from the **Edit Text Attribute tool**.
  - Editing a **filter Lua criterion** in the Object Manager's filter editor.
  - Editing an **object type filter** or a **custom collection filter** or a **column computation** / presentation (filter-map) function inside the **Table Editor**.

- For Alignment objects, the Lua editor features a **sampled formula slide tester**, which runs the alignment from start to end, evaluating the lua formula for each position along the slider. The reserved identifier ***_position*** feeds the sampled formula editor with the slider's current value.

- In all these situations, the **powerful Lua Debugger** can be activated. It allows the user to:

  - Single-step / Step into / Step over / Step out / Set breakpoint / Run to breakpoint
  - Inspect local data / Inspect call stack
  - Define multiple Watch expressions / Break on Watch change / Suspend Watch
  - Copy Watch settings to clipboard / restore Watches from cliboard

Railway editor – Automation – Scripting with Lua

- **Edit script** – and run it – or just **Run an existing script**
- Scripts are Lua programs stored in one or more text files, running in a global context in the current drawing's database.
- A script can do whatever a user can do manually:

  - Create objects / Edit objects / Delete objects in modelspace
  - Run any RailCOMPLETE user attainable command (*)
  - Run any Lisp expression using the RC-RunCommand(…) API function.
- A script has access to more than 100 API functions for database traversal and modification.
- Use scripts to create alignments, do format conversions, import objects, export objects+++
- Create your own favourite scripts, store as regular text files.

- (*) Some of RailCOMPLETE's commands may operate in non-window mode (in command line mode), to make them callable from scripts.

Railway editor – Automation – External Libraries (DLLs)

- Add an external library – a DLL file containing C# classes with attributes and methods – to your drawing.
- Create a DLL folder for the DLL under the RailCOMPLETE bundle installation folder on your computer. RC auto-detects its presence.
- Declare in your DNA an association between RC object type and the external library's class that it shall connect to.
- Create an import script that imports and deserializes an external data file shaped according to the external library'si underlying UML model.
- Instantiate by reflection the corresponding properties inside an RC object.
- Create an export script that prepares and writes RC object's external library model data to file.

Railway editor – User interface – Assistants

- Assist Create Connection Objects
- Assist Offset Profile
- Assist Assign Vertical Profile
- Assist Create Alignments Using Regression
- Assist Create Geometry and Vertical Profile Annotation Objects
- Assist Create Interlocking Control Tables / Create Interlocking Data
- Assist Create Earthing
- Assist Create Schematic Drawing

Railway editor – Assistants – Create Connection Objects

- Alignments may be connected to **form a connected graph**
- Connector objects are one of **Continuation / Bifurcation / Crossing**
- The Connection Assistant uses pattern recognition and DNA-defined tolerances to auto-detect and connect the graph (*)
- Switches (turnouts) are bifurcation objects for track type alignments.
- When breaking an alignment, RC will auto-insert a suitable continuation object.
- Crossings are simple crossings, single-slip switches (**) or double-slip switches (**)
- Joining two alignments will remove the connector and splice the two alignments into one.
- Analyse and report on track graphs / track detail design using a combination of model checks in connection objects and predefined tables which display these model check results.
- Audit tool available for in-depth validation of alignment data in a larger model: Detect and overlapping alignments, kilometration jumps at connectors etc. «Audit and auto-fix» is available.

*) Dependent on DNA
**) Not in 2023.x

Railway editor – Assistants – Offset Profile

- The assistant command for offsetting vertical profile lets the user select a source alignment (or a path through a connected graph) and then select a target alignment. The elevations in the source are projected sideways onto the target,
- Note that this function will fail should the two alignments be quite different in horizontal geometry.
- When alignments are fairly parallel, such as cable ducts following a track (but maybe taking a detour around each catenary pole), then the function does a good job and saves considerable time.
- A constant may be added during offset.
- Both vertical profile (elevation) and cant may be offset.

Railway editor – Assistants – Assign Vertical Profile

- Switches (turnouts) often connect two longer track with a shorter connecting track.
- The track designer often omits to assign vertical profile to such small items of the track plan. However, a workable BIM model in RC needs vertical profile in all tracks.
- The Assign Vertical Profile function lets the user select among three methods for assigning vertical profile to the connecting track:
  - **Straight**: Straight-line interpolation between the elevations of the first and second switches.
  - **Gradient**: Smooth transition from the source switch's gradient to the target switch's gradient.
  - **Normal**: Follow the gradient for the main track at the source switch up to its rear end, do likewise at the target switch. Make a smooth connection using two vertical profile Point of Vertical intersection betweenthe two rear-end-of-switch
- The function saves a lot of computation or trial-and-error.

Railway editor – Assistants – Create Alignments Using Regression

- Input: Sampled alignments, survey data, 2D polylines, 3D polylines.
- Use the regression tool (*) to **auto-detect straight segments, circular curve segments and clothoid segments** in **horizontal geometry**.
- Auto-detect **straight segments and circular curve segments** in **vertical profile**.
- Automated search for optimum solution.
- Calculation of mean square error and max delta between survey data vs regression curve.
- Write regression curve back to the source alignment, replacing its geometry and vertical profile.

(*) Partially in 2023.x

Railway editor – Assistants – Create Interlocking Control Tables

- Define – in DNA or in user interface menus – the definition of a Start point (a Signal or a ERTMS board) and a End point (Signal or ERTMS marker board, or other signalling means), for train routes. Do likewise for shunting routes.
- Define via-points, define excluded paths.
- Define the presentation of a train route, a shunting route, a switch condition, a track vacancy proving condition.
- Auto-generate an interlocking control table data structure containing the resulting train routes, shunting routes, extended routes, extended / shunting routes, citing switch conditions and TVP conditions.
- Export to Excel.
- Pick up data structure in a dedicated Interlocking Table object, format that table using the table editor. (*)
- Use Lua coding to auto-create overlaps, flank protections etc. (*)
- (*) Depending on DNA.

Railway editor – Assistants – Create Earthing

- Most railway object data types feature intrisic properties that describing earthing of the object.
- Select the preferred **earthing method** for your objects (*)
- Set suitable **formula** on the **earthing label / color / description** attributes (*)
- Run the Earthing Assistant on selected objects.
  - The objects have now configured themselves, selecting the best earth providing object in their vicinity, based on the selected earthing metod: Rail / Earthing conductor / Other object / None.
- Annotate the **earthing blobs** using the associate ShowEarthing tool.
- Create a suitable **Earthing Table** using the table editor and the objects' earthing properties. Let label text etc auto-compute per object, pick up in table.
- (*) Depending on DNA.

Railway editor – Assistants – Create Schematic Drawing

Signalling engineers need towork on schematic drawings in addition to the geographically correct BIM model.

- There shall be a function that creates a good-looking schematic drawing from the track model (or other alignments that form a connected graph). (*)
- Optimization twists and tweaks shall be available.
- A clone feature shall be available, where geo objects get a schematic drawing clone object. The other moves as you move one of them. They are really one object,but woth two clickable representations in a drawing. (*)
- Inerpolated kilometration shall be used in the schematic drawing (*).
- A reference alignment concept shall be used for the schematic drawing (*).

(*) Partially in 2023.x

Editor
DNA and DNA Tools

Railway editor – Productivity – Overview of needs

- All disciplines: 3D export, Quantity takeoffs, Object lists
  Cable plan (*), Asset database delivery (**)
- Civil works: Objects, Foundation plan, Flexible tubing (*)
- Track: Objects, Schematic track plan (*), Cant table (*)
- Contact Wire: Objects, Support pole table, Earthing, Cables
- Low-voltage: Objects, Cabinets, Cables
- Signaling: Objects, Geographical signal placement plan,
  Signage plan, Signage table,
  Schematic plan (*), Signal- and balise placement (*),
  ATP code table (*)

  Telecom: Objects

*) Not in 2023.x / dependent on DNA

**) Merge data using Object Manager and the copy-with-key feature.

Railway editor – Productivity – ATP Encoding Table (*)

- The editor shall be able to handle all data requests needed to automatically compute the data for an ATP encoding table (*)

*) Not in 2023.x / dependent on DNA

Railway editor – Productivity – 3D libraries - LOD

- There may be one or more 3D libraries available with the DNA:
  - The railway administration (Infrastructure Manager)
  - The consultant engineer
  - The contractor who builds
  - The end user, the operating company
- It shall be possible to reduce the level of detail when exporting to 3D (LOD, Level Of Detail) to create less heavy exported models.
- Geometry in the 3D library can be complex, the user decides where the level of detail should be.
- Avoid "data explosion" with slow BIM applications when the models become detailed - keep a minimum representation in the BIM model and keep the 3D model in separate files - export to another file.
- It shall be possible to construct the layer names, coloring and block composition in exported data per export job, to satisfy varying requirements in BIM requirements for projects.
- Metadata in3D geometricalobjects (library DWG files) may contain meta-data that RC will exploit to guide pole routing, sighting, text on signs etc.

Railway editor – Productivity – DNA Validator

- The editor shall feature a function that analyses a given DNA file and checks for:
  - Inconsistencies
  - Double declarations
  - Missing resources
  - Bad syntax (DotLiquid, XML, Lua)
  - Lua coding errors
  - Bad use of identifiers
  - Use of deprecated identifiers
  - Inexistent attributes in XML elements

*) Not in 2023.x / dependent on DNA

Railway editor – Productivity – DNA Tool

- The editor shall feature an editor for **creating and maintaining DNA**. (*)
- This **DNA Tool** lets the **DNA agent** concentrate on modeling issues and not on bringing all the small details into correctness.
- The former manual creation of XML files with XPPq (**) macros shall be replaced by a versatile and (fairly) user-friendly interface for the DNA agent.
- The DNA tool shall be able to produce DNA mapping files, which describe how an existing data model built using DNA version X shall be salvaged and ported to DNA version Y. (*)
- The DNA Tools shall produce cross-reference / indexed documentation for identifiers in use, Lua functions – presenting all information in a human-readable way as a document and / or as a HTML web page.

*) Not in 2023.x / dependent on DNA

**) A French tool, the EPEIOS open software project, courtesy of Claude SIMON

The Distant Future Roadmap

Railway editor – The Distant Future – Roadmap

Functions that may be developed:

- **Runtime 2D-symbols**: Parametrical design of 2D DWG blocks using Lua stored in DNA, i.e., create double-curvature switches etc.
- **Ballons**, to label items for Bill-Of-Quantity and annotated drawings
- **Cable data type** as supplement to the alignment data type.
- **Cable routing** functions.
- «**Foxtail**» - flush metadata out of drawing before delivery.
- **railML** 3.x export / import
- IFC import?
- Desktop **capacity simulator**.
- Stand-alone version of RailCOMPLETE (Ares Commander?).
- Free viewer version.
- Lower-cost table / data extractor version (objects are locked).

None of the above are part of 2023.x. Time to introduction is not known.

RailCOMPLETE