

**E³.HarnessAnalyzer**

**Maintenance Package**

**7.0.0**

**Created by:** Rüsseler

**Phone:** +49 9131 / 691430

**Email:** martin.ruesseler@zuken.com

**Created at:** 21.03.2017

© 2014 Zuken E3 GmbH All rights reserved

**Specification**

**Index**

[1 Open issues 3](#_Toc472935249)

[2 Revision history 3](#_Toc472935250)

[3 Glossary 3](#_Toc472935251)

[4 Globalization & Localization (Rappel) 4](#_Toc472935252)

[5 Entire vehicle and inliner consistency reporting (Rappel) 4](#_Toc472935253)

[6 Redlining modification 6](#_Toc472935254)

[6.1 Quality Stamping (Hahn) 6](#_Toc472935255)

[6.2 Packaging respectively grouping (Rappel) 12](#_Toc472935256)

[7 Weight calculation (Hahn) 12](#_Toc472935257)

[8 “What if” analysis on bundles (Rappel) 15](#_Toc472935258)

[9 Visualization of QM issue reporting (Hahn) 17](#_Toc472935259)

[10 Windows 10 support (All) 19](#_Toc472935260)

[11 Tablet application (Rappel) 19](#_Toc472935261)

[12 Analysis view protection (Hahn) 19](#_Toc472935262)

[13 Bundle calculation according “Daimler” formula (Rappel) 19](#_Toc472935263)

[14 Center curves as splines (Rüsseler) 20](#_Toc472935264)

[15 Offset control in graphical compare (Rappel) 21](#_Toc472935265)

[16 Enhancements of graphical compare to reduce move differences (Rappel) 21](#_Toc472935266)

[17 Additional small issues (All) 22](#_Toc472935267)

[17.1 Hourglass or progress in KBL compare 22](#_Toc472935268)

# Open issues

|  |  |  |
| --- | --- | --- |
| Designation | Para. | Description |
|  |  |  |

# Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Name | Company | Description |
| 1.0 | 02.01.2017 | Rüsseler |  | Initial |
| 1.1 | 04.01.2017 | Rappel |  | Some enhancements |

# Glossary

|  |  |
| --- | --- |
| Term | Commentary |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Globalization & Localization (Rappel)

We need to globalize the application first properly, before we can localize it to English and German.

It needs to be decided, if we localize the help file, too. This will be a bigger package then.

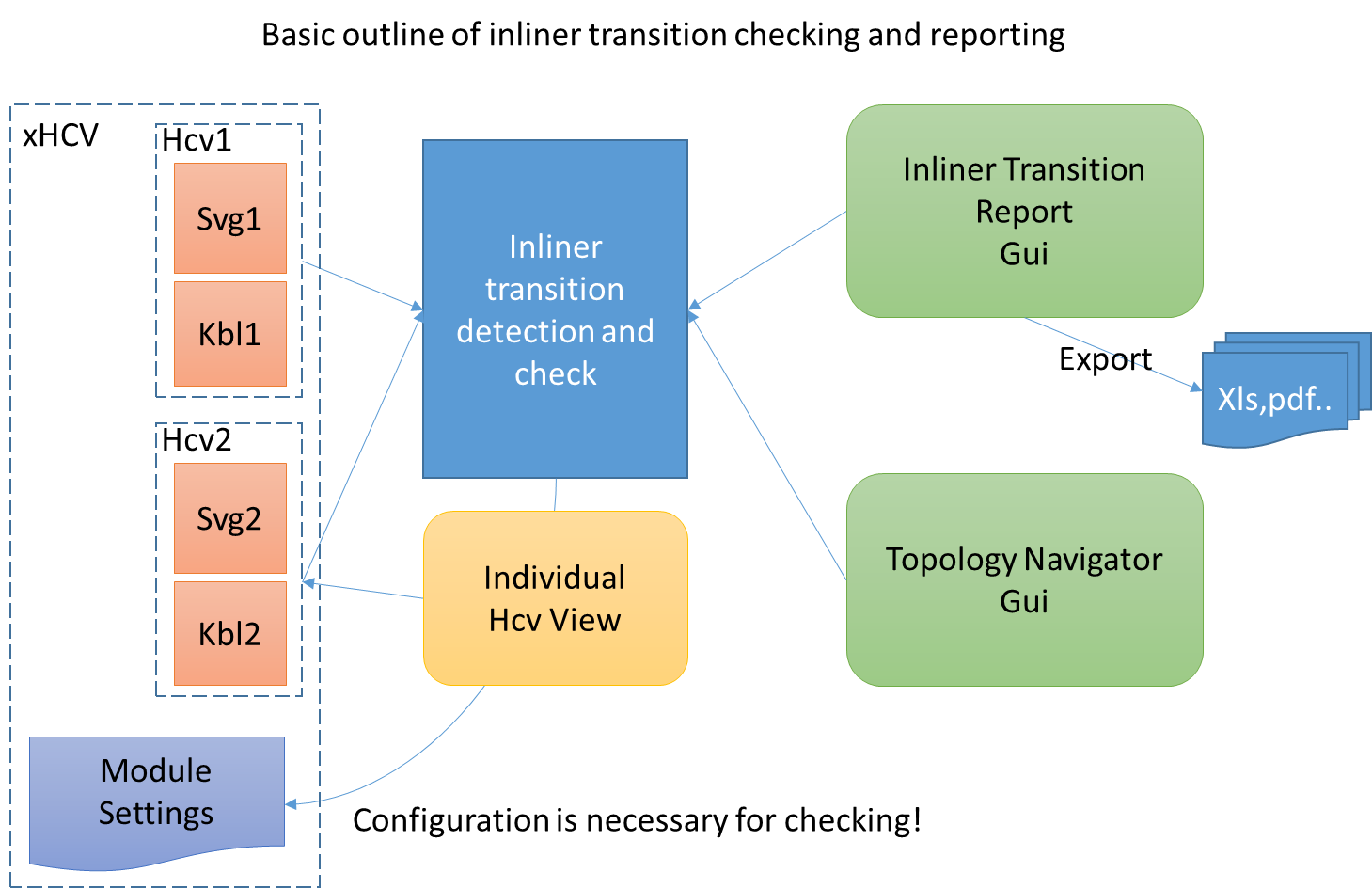
# Entire vehicle and inliner consistency reporting (Rappel)

Today we have an interactive solution to trace the inliner transitions of a given wire using the xHCVs. We can start the analysis from the individual wire using “Show entire routing…”.

The final solution needs to generate reports over all inliners in **one** step. How do we put this together?

For the real reporting solution, only the KBL content is necessary and the loading of all HCVs with the graphics is time and memory consuming.

A solution could be to have a switch for xHCV, which controls the loading of the SVG files? If not set, only the KBL and the module settings are loaded. The basic reporting process could be the same, regardless of the additional graphic functionality if there are SVGs available.



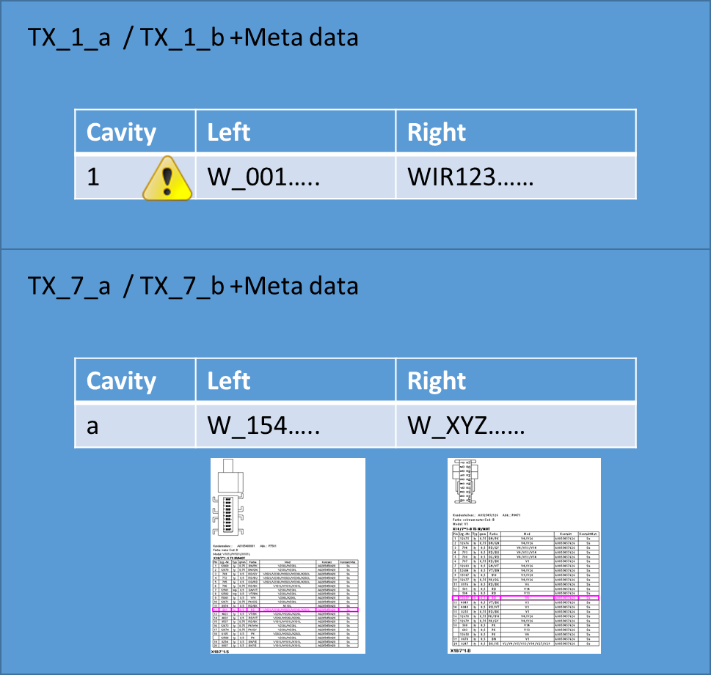
Basic issues:

* Configuration handling: each KBL has a set of modules, but with different module names, and for a valid transition, the modules on both sides of the transition need to be switched correctly! The xHCV has a special XML structure which can keep the module setting for each internal HCV.
* The topology overview used today needs to be modified. Today the car background picture is mandatory for generating the topology sketch. This has to be relaxed so that the user is able to place compartments without a background picture.
* A load switching on opening of xHCVs for graphics would be needed.

The process of detection of all inconsistencies could be invoked on opening a specific transition view. It seems to be reasonable to make this dialog not modal?

This specific view (form with grid / tree structure or similar) is to be implemented to visualize the results. Exporting to Excel or PDF is invoked from this view. Cross probing should be supported. There could be a tree with a node for each inliner pair. Under each node we might use a grid and combine the information to show the cavity content of both sides. Instead of the tree, a repeater like the one we used in Vesado for the conflict resolving would be even nicer.

Basic view outline:



In case we have the graphics loaded, it would be great to have both connector face/ table graphics available in this view for each of the inliner pairs. So we have the logical table from the KBL and optionally the table and face views from the SVG.

The necessary checks for each transition within a cavity are basically defined:

*CSA (Error)*

*Wire type (Warning)*

*Color (Warning)*

*Signal (Warning)*

*Plating of the terminals (Error)*

*Module configuration setting regarding routed wires (Information)*

It would be useful to classify the checks like stated above. This conflict classifications should be stored as setting in our application settings XML, where the user is able to change the predefined settings, but we do not offer an UI for this.

The inliner pairs are the root structure and each pair contains a grid sorted by cavity numbers. We have to relax the hard-coded restriction of cavity count equality check while determining inliner pairs on xHCV load to also display potentially inliners which may belongs together but have invalid cavity counts. In addition to this we have to consider that even cavity count of inliner pair is identical, the numbering might be different. So we have to have a row for each cavity number and either on left or right or both sides we can have wire content an contact material in grid cells. For this merged cells displayed in grid would be great. This also affects cavities with more than one wire assigned (module configuration dependent).

We have to consider the defined module configuration within this view. Inactive wire and housing content has to be displayed greyed-out like we do this in the drawing. It has to be discussed how we handle the different wire content regarding the conflict checks in relation to the defined module configuration.

# Redlining modification

We have to think about compatibility as we have to change the structure of the redlining content! How to handle existing redlining information from older versions?

* Conversion
* Show old content without new functionality
* Version number

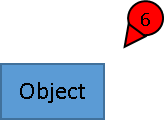
## Quality Stamping (Hahn)

This functionality is not an Analysis View, it needs to be supplied within each opened hcv quite similar to the redlinings.

A new data structure principally similar to redlining (it will **not** be integrated in the redlining) is to be introduced for the quality management. Each entry in this data structure keeps information about a test point (quality stamp): is has an internal identifier, a reference number, some test specification and meta data about the editor as well as date/time information. The link between an entry and the objects within the hcv is established by the internal kbl id.

Each entry is to be visualized in the drawing by a predefined hardcoded symbol displaying the reference number of the test point (3 digits should be sufficient; it needs to be in a way that it is readable when printed!).

We may scale the symbol a bit to match the number of digits to save some space on the drawing.



The symbol should be placed automatically near the related object and should be rotated automatically so that the edge points to the object, which is in most cases the center of the bounding box (exception: segments). Nevertheless the user must be able to move the symbol to a location where it does **not overlay** other important drawing information. Therefore we need to keep the relative coordinates of the symbol origin to the corresponding objects insertion point. This allows to place the stamp symbol on import to a drawing where the related object is, even if the location changed somewhat. (if the handling with absolute coordinates is much simpler, we could discuss this too)

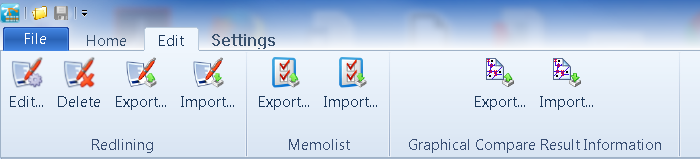
A dedicated layer needs to be introduced for these stamps, so that the user can switch off the display on request. It might be helpful to draw a light dashed reference line (switchable?) from the edge of the marker to the object somehow.

*The move of the markers need to be implemented. A marker is to be selected and may show the grip at its origin. A command action to move a grip could be used, we need to check whether this fits in the existing selection logic or statemaschine.*

For the segments, we have already some trial implementations to improve the detection of the centerlines. Today the bounding box center is used for segments, but this gives not a good result to place a segment marker on bent or edged segments. These trials might be needed and implemented properly on our group objects and filled only for segment objects once.

* SHALL THIS ALL BE POSSIBLE FOR MULTIPLE DOCUMENTS, if yes: the behavior is not clear when the user switches between the active document.

Access to the functionality could be from here:

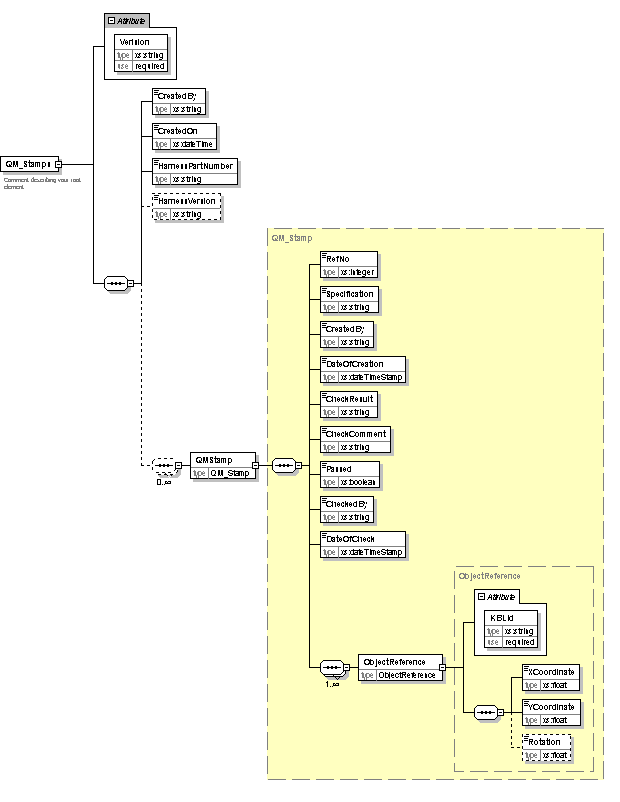


A **new** section “Quality stamping” similar to the “Redlining” section seems to be reasonable. The activation/deactivation of the stamp layer could be integrated also in the menu group, as we will not have more layers in the future.

On “Edit” the editor form comes up.

The data itself will be held in a xml structure within the hcv.

XML Schema (preliminary):



Most of the test points are related to physical objects like connectors, but there are also entries which do not relate directly an individual object (i.e. test point marker for “full electrical check” somewhere on the drawing). Basically these objects could be linked to the one and only harness object. These markers may consider their coordinates relative to some origin of the drawing.

A form or pane (better?) with a grid is needed to show and edit these test point entries. The display of the entries and their symbols must reflect the module configuration: If the related objects are not in the current module configuration, they will be greyed out (see information hub).

The numbering of the entries should propose the highest number but needs to be changeable by the user. A warning must be issued if a duplicate number in entered.

The test point data will be exportable and importable to a hcv just like redlinings.

This can only be done for the same harness with the same part number and version.

(see redlinings)

(due to the kbl id linkage which we have to rely on that). This functionality can be used to transfer QM stamps within the same HCV.

Additionally an export for further processing in other tools is needed. This can be done based on the grid information and transferred to Excel.

There must be functionality to delete the entire test point data or individual entries from the grid.

Cross highlight from the marker symbol to the grid and vice versa is necessary.

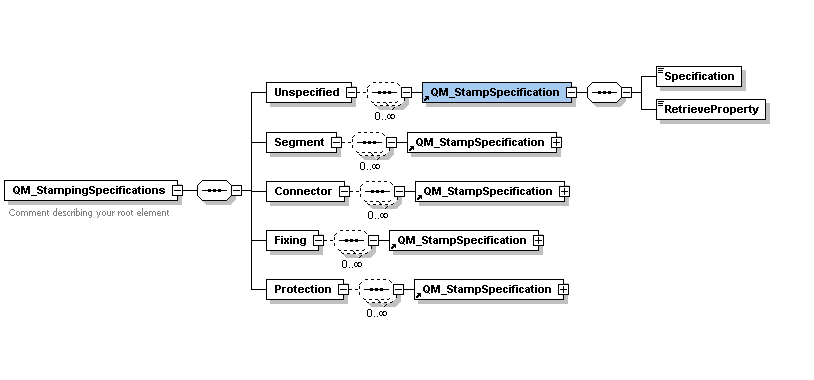
Adding these test points is crucial as there can be easily 500 entries on a large drawing.

There should be several ways to add these entries: from the drawing by selecting an object as well as by selecting an object in the information hub similar to the redlining creation. A keyboard shortcut(s?) is definitely necessary to create these entries fast.

Adding entries without relation to an object can be done in the test point editor. The resulting marker needs to be placed then possibly in the origin of the drawing and must be moved afterwards by the user. Alternatively a context on the drawing might be employed here, which would make the handling easier as the location of the marker could be determined here in on step.

Regarding i.e. test entries which relate to measurement data (segment length or the like), a lookup to propose the tolerance values additionally to the segment length would be nice. We have to define where to keep this lookup, possibly in the general settings xml. Furthermore the idea exists, that there can be a lookup allowing to easily pick typical specifications from, depending on the type of the object. If there is no object type given, a set of typical test specs can also be given here. The test specification proposal should be overridable by the user, in case something different is needed. This selection should also be supported by some keyboard action to speed up the entry.

Idea of lookup (preliminary):

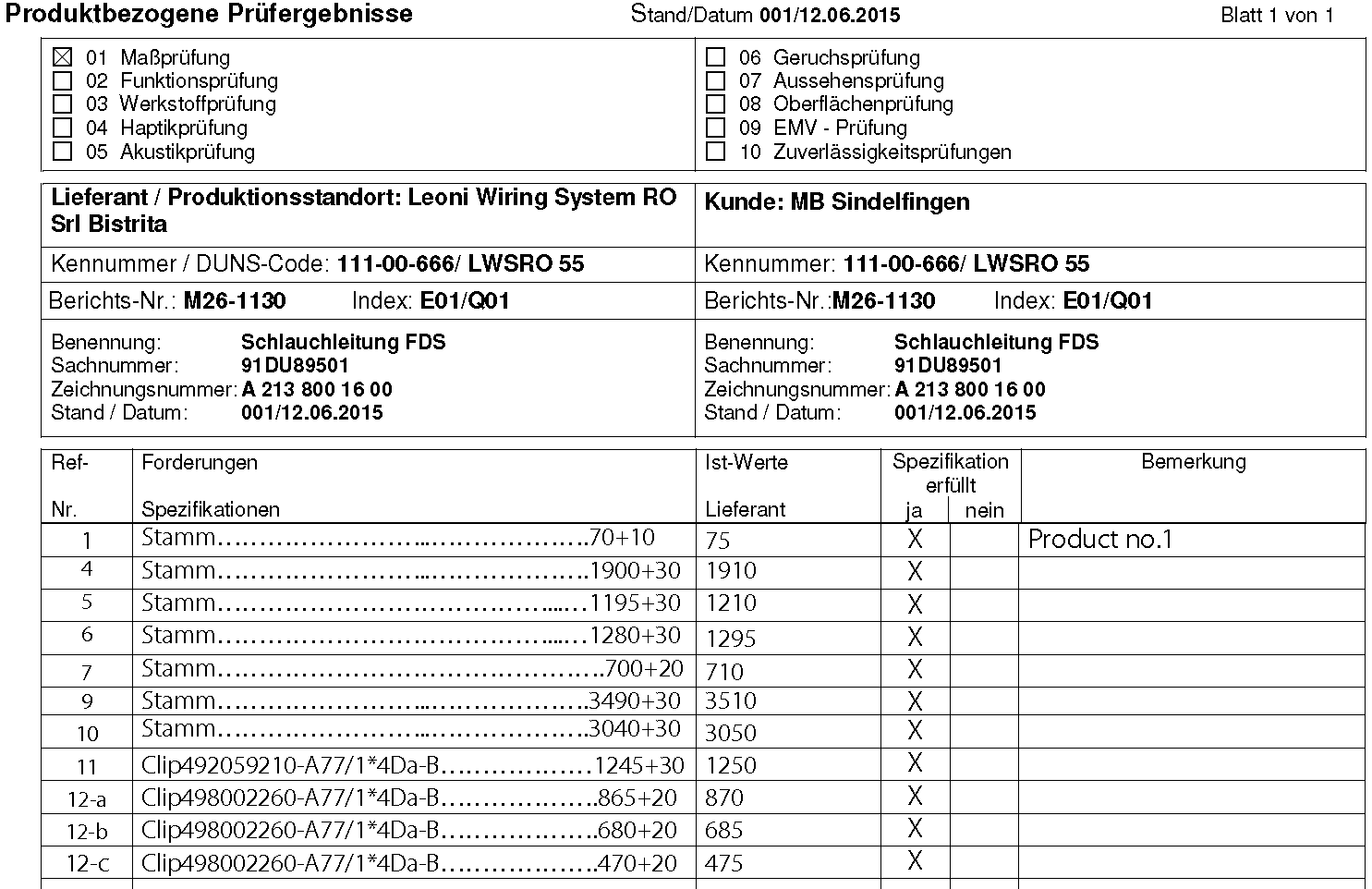


This test point data should allow for entering/editing of the final test results for each test point, too. Therefore we need also metadata of the user who entered these information and a passed flag.

The grid content will be exportable to Excel and several sorting / filter functions from the grid are to be employed before i.e. exporting.

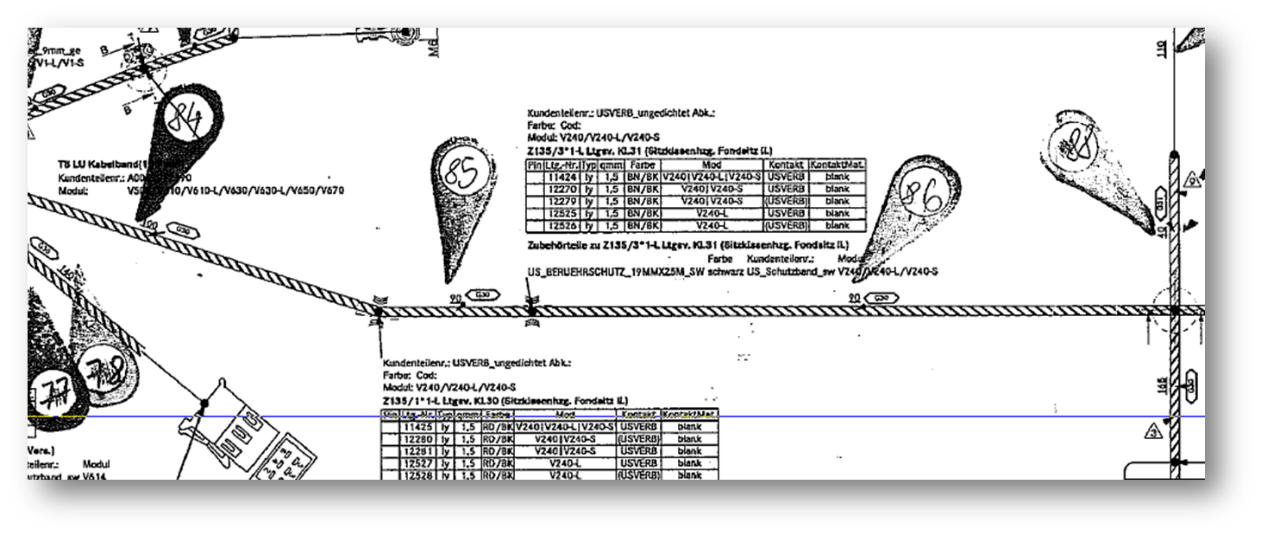
* Today’s algorithm for position of the symbol needs to be reworked; especially on bend segments it is placed far away from the point of interest (Bounding box problem). If there is no other chance, we might need to use some reference lines to visualize the connection somehow.

Sample of a final report which will be created by user manually out of the test point report:



Typical stampings and special cases:

* Segment direction where the marker should point to the clock symbol, but where we can only link this stamp to the segment. Rotation might be a problem here, the marker can be moved manually.
* Clip orientation is similar to the above mentioned, but here we can link to the clip.
* Segment length to a connector will be linked to the segment, but the connector name is displayed in the report.
* Cable duct, could be placed on the duct or on the segment.
* Clip to clip dimension, is typically placed on the segment.
* Special case?? Multiple markers on different segments which should depict one dimension (“cumulated quota”) results today in different markers with different numbers but having one tolerance (100+90+90). Multi-selection of the segments result in one stamp entry with markers for each segment showing the same number. This means we need to map multiple objects and markers to one stamp entry (See xml structure and the multiple object references)!
* Mounting hint on the drawing: stamp might be linked to the object but should be moved to the depicting graphics and need possibly the manual rotation.
* Electrical test: has no link to the object? Or will be linked to the harness id and needs to be placed and rotated manually.



## Packaging respectively grouping (Rappel)

A super structure is needed to group redlining entries. Each of these packages will have several meta data, which are still to be defined. There will be some identifier or change number at least, along with the user name and a timestamp.

(see now first schema draft)

One question is to be discussed: do we stay orthogonal and have always a group or do we allow to have redlining also without groups? For the QM redlining, this might get a bit cumbersome with all these groups. An alternate approach might be to have a QM group or a junk yard group, where all “ungrouped” redlining entries are kept.

We go for independent redlinings and groups

It seems reasonable to keep these things orthogonal. This will involve changes on the corresponding GUI. We have to think about move functionality of redlining entries from and to other groups.

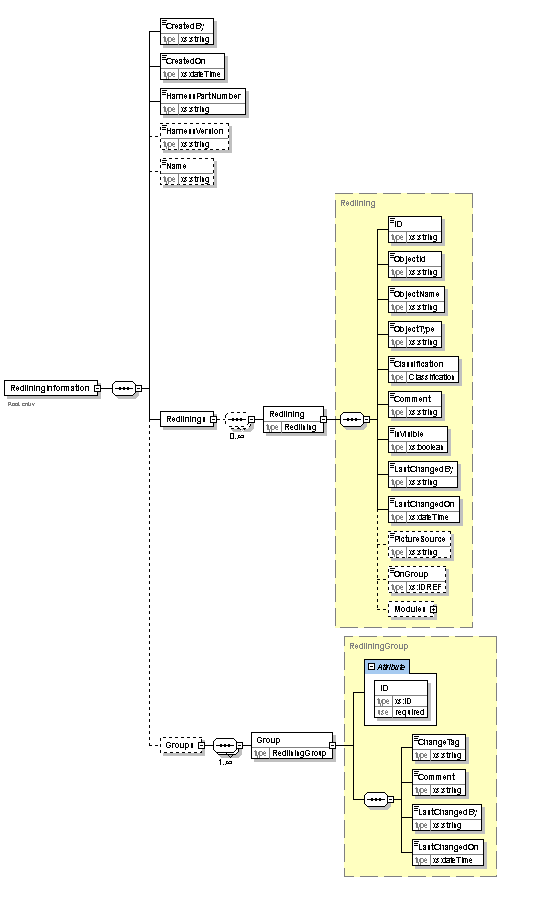
The redlining entries along with the grouping must be exportable to an XML structure for use by other external tools. As there are also graphical redlining entries, the pictures should be exported as PNG and need to be linked to the XML entry. Therefore a ZIP container should be created on export which contains the XML structure and all necessary pictures named by an identifier, so that the link to the entry can be established.

Additionally the existing Excel export should be enhanced with the group information and the PNG files of graphical redlining can be put into Excel cells as embedded object.

There is a request from Daimler that we add the corresponding modules (partnumber and name) to the redlining entry for the corresponding object. The zgs / kem information for the module should also be added (just the way it is retrieved in the grid)

The final export will be a zip container which contains the rdl with the new schema and the png files. On import in Harnessanalyzer, we need to check if we get an old rdl file or a zip container where we have to extract the internal rdl first to import this.

The picture content is not used for the Harness analyzer loop.



# Weight calculation (Hahn)

We need additional functionality to get the weight also by own calculation.

The mass value for wires is only existent on the general wire definition and must be a weight information per length.

There is no instance weight of a wire occurrence. This should be calculated from the information of the general wire and the appropriate length information from the occurrence.

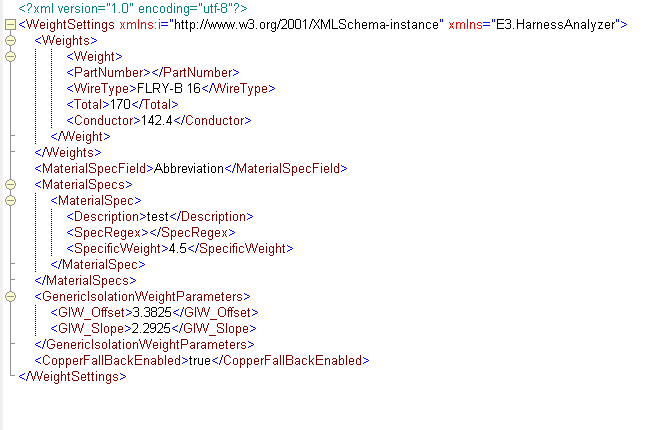
In case we have a core we could only take the weight information from the cable and divide it by the number of cores to give a rough information.

The material of the conductor needs to be taken definitively into account.

This material information can only be retrieved from the specification via an Regex. (This was basically stated in the discussion with Leoni / Dräxlmaier)

Nevertheless, the same fallback like for the diameters should be employed. A new XML similar to the Diametersettings.xml should be implemented. It seems to be a better idea to separate these information completely and not to mix them in the diameters.xml.

Example:



The retrieval process could be like: try to match the part number (1.a), if not try to match the wire type (1.b), if this fails try to apply the given regular expressions to the specified field of the wire definition (2.) and if a match is detected use the specific weight information supplied with this regex to calculate the specific weight in g/m. In this case the generic insulation weight needs to be added for the given CSA.

In case no information is found (3.), we may either set this wire to zero weight and notify this, or we use a generic formula (based on copper then) for the total and the conductor weight. We can use the insulation generics and calculate the conductor weight from CSA and the specific weight for copper. (we should add the specific weight for copper and aluminum in the general settings independently from the material specs in the XML).

The third fallback seems to be necessary as the wire type does not always give information about the material of the conductor.

- Total weight in g

- Conductor weight in g

Abbildung : Weight calculation process

* **Lookup Calculation**

1. Wire/Core

Total weight (Tw) in g = [Total] g/m \* wire/core length in m

Conductor weight (Cw) in g = [Conductor] g/m \* wire/core length in m

1. Cable

Tw in g = [Total] g/m \* cable length in m

Cw in g = [Conductor] g/m \* cable length in m

* **Material spec calculation**

1. Wire

Cw in g = specific weight g/cm³ \* CSA cm² \* wire length cm

Tw in g = Cw in g + (Insulationweight g/m \* wire length in m)

1. Cable

Cw in g = core-conductor-weight g \* cable.cores.count

Tw in g = = core-total-weight g \* cable.cores.count

1. Insulation weight calculation

The formula to calculate the specific insulation weight (Iw) in g/m will be hardcoded but uses the coefficient from the XML.

Iw = [GIW\_Slope] g/(m\*mm²) \* [CSA] mm² + [GIW\_Offset] g/m

The source of the weight information should be kept on the wires of the selection locally, so that we can see what was retrieved for the individual wires.

The length comes from the KBL, but we need to check which of the two length values are to be taken: This has to be derived from the defined length information setting (production, DMU or supplement length) in the general settings.

These are examples from Leoni for material specifications, which in turn are kept in the abbreviation field of the general wire definition in the kbl.

Fp           ->          Kupfer-Magnesium Legierung

Fq           ->          Kupfer - Zinn Legierung

Ie            ->          Aluminium

Jz            ->         Aluminium

Om        ->           Kunststoff (LWL) - might need special treatment as this in not a conductor

These fragments need to be kept as regex along with the specific weight of the material. The field where to match the regex can take three different values: “Partnumber”, “Description”, “Abbreviation”.

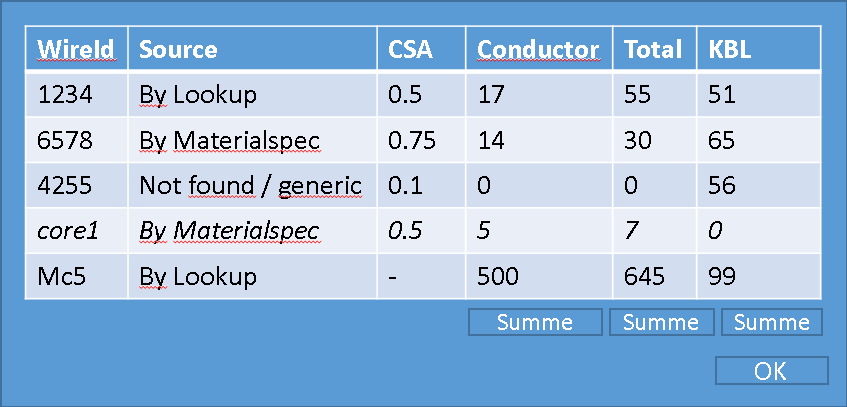
The weight result should separate between **mere conductor weight** and **total weight**.

Additionally the summed up values from the KBL mass information of the selection should be supplied for easy comparison-this needs some calculation as it is length specific information (gr/m) .

There might be a small error regarding LWL and the material spec as if this is retrieved we would still add an insulation to it.

* LengthInfos and CSA from Core.
* KBL for core is the KBL from Cable divided cores-count ()

Regarding multicores, there are two cases to be considered. All cores of a multicore are in the selection or only a part is selected. If the entire cable is selected we need to retrieve the data for the cable. If some cores are selected, we calculate these individually (kbl weight is calculated from cable as average weight). It would be helpful to visualize in the grid of the calculator if we have all cores or not. Either a hierarchical cable/cores display is used, or a flat structure of cores and wires is used and individual cores are marked in italics. We could add the cable instead of the cores if all are found.



To calculate the KBL value, the length of the wire and the length specific value from the part definition is needed. We should try to find out the dimension by analyzing the unit entries given. After having determined the result, a validation should be done (by using the original kbl-weight in g/m), and in case the value is not reasonable, no kbl weight is supplied (n/a) in the grid.

This validation will be done by two hardcoded formulas taking the CSA and handing out the max and min value for checking.

MinValue= 9,0\*CSA -1,0 in gr/m, CSA mm^2

MaxValue= 11,0\*CSA+15,0 in gr/m, CSA mm^2

A similar functionality might be implemented for a selection of segments. Here we need to clearly state that the result is only taking the mere wire content regarding the weight into account - no protection or clips are included. Basically the mass information from these parts are available in the KBL, but the quality of the data seems to be questionable. It needs to be considered that we can take only the summed up segment length for the calculation. So a calculation over all segments and a calculation of all the wires in these segments might yield slightly different results.(wire length versus segment length)

Further issue:

* What about the water tubes and LWL content? It is sometimes difficult to detect what is modelled as a wire, but actually is a real tube. As we select segments, there is no chance to remove these “tube or LWL wires”
* We may provide the similar functionality for user wires and removal of wires? What about the cut and paste things, we have to implement for the bundle calculation. A module content based calculation seems to be overkill here, but adding or removal might be useful? Removal would be relatively easy (just remove from the grid and recalculate) Adding would need a meter weight entry or a specification.
* There is still a problem with the units in the kbl. Specific meter weight in wires is not modelled properly. This must be a length related value for the general wire definition with a unit gram/meter or the like. This is can only be done with a UnitName information in the kbl, but we have no real modelling. Most of the values in the kbl use here gram or kilogram only! For calculation now, this gets more difficult. (see proposal above)

# “What if” analysis on bundles (Rappel)

The bundle diameter determination allows to add or remove wires on a selected segment for a given configuration. This can only be done manually. A typical use case is the following: a user wants to move a selection of wires from a bundle to another bundle temporarily to get the diameter changes.

A selection of wires should be movable by means of the clipboard or possibly some other local storage. The clipboard would be more flexible, as we could handle also a selection from some Excel? The add / remove information could be in there, too.

A selection of wires are to be removed from one segment and added to another:

* Multi-selection of the wires on the source segment
* Remove the whole selection and keep the information also in the clipboard
* Close form on that segment and open the dialog on the target segment
* Select Insert user wires from clipboard information (only if the clip board contains reasonable content)

The existing context entries need to be checked, there is “copy wire(s)” but this can only copy all added user wires from one configuration of a given segment to another configuration of the same segment. Maybe a solution to retrieve all modifications (removed and added user wires) can be copied from one configuration and applied to another configuration, so that it is easier to propagate these changes? In this case we would need to find wires to remove and add user wires if these are not already there.

Possibly this needs to be streamlined a bit more. Basically there are adds and removes in the configurations of a given segment as well as moves between different segments and configurations.

Another way could be to separate the copy to clipboard from the add and remove, so a selection from the wires (added or removed) are copied to some clip board and this can be applied to any configuration. Wire numbers which are to be removed and which are found in the target configuration are removed, user wires are added for each wire to be added in the clipboard.

* Added wires from clipboard has to be user wires; we need to use the given wire number with “UW\_” prefix to identify these pasted objects later
* Is some kind of preview of this clipboard reasonable, before the user applies the temporary changes?

The final dialog on close of the modal dialog after modification should be reworked as it is a bit misleading- changes are discarded, not removed.

Possible typical use-case is following: User opens bundle dialog for first segment, deactivate (temp. remove) two wires, add two user wires and opens a second bundle dialog of another segment. The two deactivated wires and both user wires should be “moved” to the second segment. While pasting the content, it has to be checked if the given wire(s) are already existent:

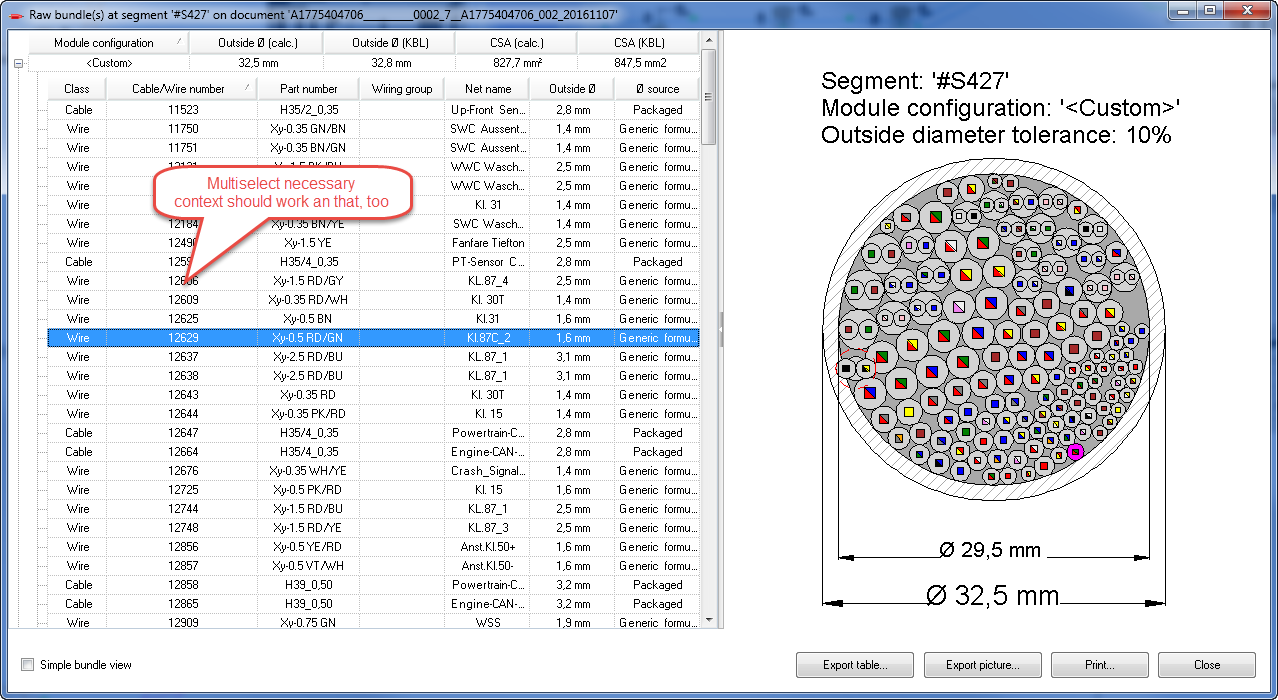
YES 🡪 Wire has been deactivated in source segment, the same wire will be

deactivated in destination segment, too.

Wire is active in source segment, nothing happens

NO 🡪 Wire will be created as user wire with number “UW\_” and original wire

number.



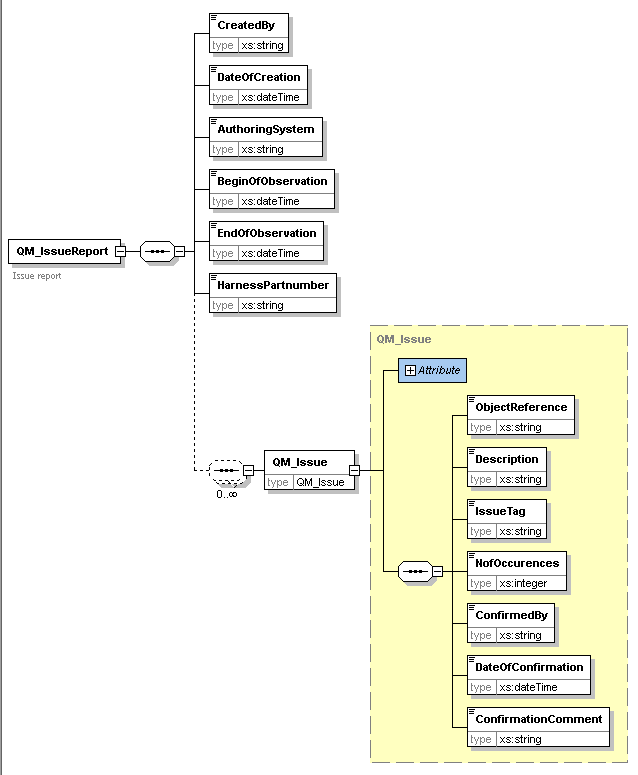
# Visualization of QM issue reporting (Hahn)

This is a new feature regarding quality management. In production, a test report can be generated where an object reference (user ID in KBL) can be bound to different problem messages found by the test system. The number of occurrences of a given problem can be retrieved for a given time period.

This will be an XML structure which should be read in by HarnessAnalyzer and visualized somehow. Additionally a comment for each entry should be editable by the HarnessAnalyzer, so that this feedback can be documented for QM.

After opening of the issue file , the harness part numbers should be compared and a warning issued, if these mismatch. If the part number is not given in the issue file, no warning should be shown.

Schema of the QM\_issue report:



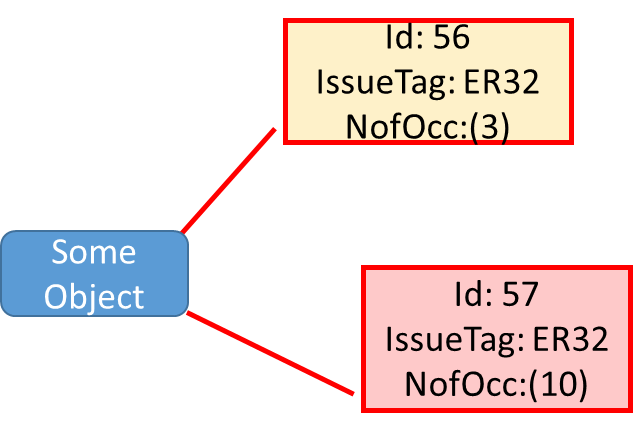
The visualization should be able to classify the number of occurrences by color: “the more often a problem on a given references comes, the more red it gets….”

It is to be discussed, if the scaling / grouping is a fixed value (from 1 to 10 is blue…) or if the scaling is done automatically by the highest number of occurrences. We may need to defined coloring and number of available groups in the general settings.

For editing / viewing of this XML structure, a dedicated dialog respectively pane similar to the redlining might be best. On top of the form we should display the meta data from the XML like date of creation and time frame.

This would allow for a streamlined cross probing, but it could also be done in some dialog which should not be modal then.

Basically, the usage of a real analysis view would be more streamlined. On opening of the view, the file might be requested and the classification is done. As long as the view is open, the grid representation of the XML needs to be accessible with cross highlight and jumping. The coloring could be the global pen style switching, possibly.



There could be some more nice eye catching statistics shown in bar or whatever graphics for QM department.

There came an additional idea/enhancement from Leoni: there can be more than one error code for a given object, so a display of some kind of maker with the error code (which should be colored) should be placed in the graphics on the object. So there can be more than one marker (possibly in different colors) on the object.

These marker should be clickable, so that jumps to the object or to the report entries are possible.

# Windows 10 support (All)

We must check the application regarding functionality and look regarding windows 10. There are already some “look” issues reported, so we have to check how this can be solved.

# Tablet application (Rappel)

The tablet functionality is already implemented for some demonstration, but needs to be fully implemented, so that we can deploy this feature. This functionality might require some settings to enable / disable several features or control the behavior.

# Analysis view protection (Hahn)

A new analysis view is to be implemented, showing a selected protection in the drawing. The selection should be done by a drop down, filled with all available protection specifications from the kbl. There are changes discussed, that the protection specification might get also a part number (Daimler) instead of the specifier “G10”, so we need to take this into account on implementation. Both information should be accessible. From the kbl there may be four fields on the wire protection interesting here: part number, abbreviation, description, and protection type.

# Bundle calculation according “Daimler” formula (Rappel)

Today, HarnessAnalyzer uses a circle packaging algorithm to generate the bundle diameter values. At Daimler, there is a request to all suppliers that they have to provide the diameters calculated by a predefined formula.

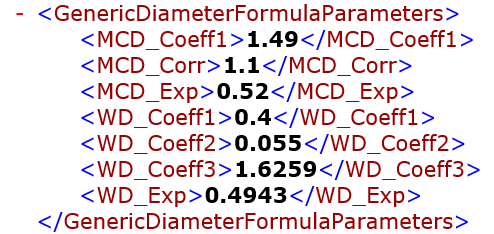
Therefore we need to implement an additional calculation using this formula and add this information to all viewpoints. The fallback regarding single wire diameter and the like will be the same as for the existing packaging.

The basic formula could be hardcoded, but the individual parameters need to be defined in the diameters.xml. There are already parameters implemented which are used to calculate multicore outer diameters. So we need to add a second set of these parameters to be used for the entire bundle calculation. (beware that in the initial formula definition, the tolerance is integrated).

So either this formula uses a different tolerance independent of our tolerance, or we remove the tolerance from this formula calculation and apply the tolerance given from our settings? The second proposal would be clearer, but then the users need to be aware of setting the right tolerances! Using individual tolerances will cause confusion, so we apply our tolerances and set the BDL\_Corr to 1 !

Return CSng(Math.Round((\_diameterSettings.GenericDiameterFormulaParameters.BDL\_Coeff1 / (diameterCount ^ \_diameterSettings.GenericDiameterFormulaParameters.BDL\_Exp)) \* accDiameters \* \_diameterSettings.GenericDiameterFormulaParameters.BDL\_Corr, 1))

New Set of parameters BDL\_xxx to be added



A generic term has to be found, so that the user understands the differences between all the different diameter values - circle packaged, from kbl, from formula. (Calculated versus packaged or packetized; calculated versus approximated, geometrical versus calculated)

Proposal would be geometrical versus calculated. We have to adapt the screen shots anyway and localization is also affected.

The calculated value should be displayed also in the bundle picture by some circle and a legend.

# Center curves as splines (Rüsseler)

Today, the center curve information from the KBL is transformed to mere polylines. In reality, these are spline definitions and we should try to display splines here, too. The transformation of the degree definition in the KBL needs to be defined, as Vdraw uses only a given set of splines.

* Investigation of a reasonable transformation is necessary – we will not be able to exactly reproduce the original splines in all circumstances.

In Vdraw there are basically four possibilities:

vdStandard= polyline (degree 1)

vdQuadratic= spline (degree 2, vertices are control points, tangent in Vdraw not applicable)

vdControlPoints= spline( degree 3 vertices are control points, tangent in Vdraw not applicable)

vdFitting= spline( degree 3, vertices are on track, tangent necessary)

If the degree in the kbl is 1, we just generate a polyline and forget about any start/end vector. (vdStandard)

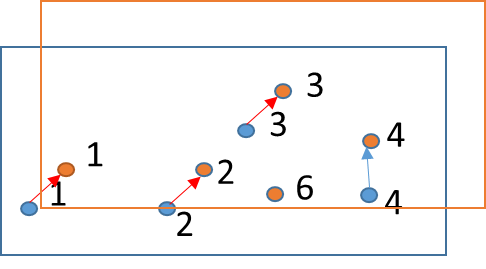
If the degree is 2, we can use a quadratic spline (vdQuadratic + vectors)

If the degree is **more** than 2, we could use the cubic spline (vdControlPoints + vectors)

# Offset control in graphical compare (Rappel)

A basic general offset for the graphical compare was discussed. In case a move of the whole graphical elements is done, this could be corrected by this offset and at least these differences will not show up. We should discuss this in detail, as it is not clear if the efforts to be taken give substantial benefits.

The adjustment of the vector needs to be easy, as it does not help to do this by trial and error. Having a big drawing this will take too much time. If we had a small kind of preview displaying certain main points like the vertex or connector face center points, this might help if we overlay these from both drawings and try to find the move vector? (if i.e. 80% of the points – with some representation id – have the same move vector, this might be an indication).



# Enhancements of graphical compare to reduce move differences (Rappel)

There are still many move differences in practical work, which in turn causes a lot of work to check them. There are requests to enhance this detection somehow. In LDorado there are functionalities to align bundles somehow automatically to some borders and this results often in smaller moves of objects.

The main problem here is, that not all moves are implemented by real transformations, especially segments are created and hence moved by absolute value modifications.

* Tables in general are marked different, if there is a change somewhere in the graphics. Is there a possibility to handle this more efficient? A single change of a text in i.e. a cavity causes the whole table to be marked. Is there a chance to separate texts changes from line modification? This would ease the understanding of the changes from the users prospective greatly.
* Small segment moves or rotations cause many differences
* We need to extract typical changes from the SVG and try to analyze the possibilities here in conjunction with business logical know how. Is there some pattern detection helpful?

# Additional small issues (All)

## Hourglass or progress in KBL compare

The KBL data compare is quite time consuming and there is no feedback to the user, the application is hanging somehow.

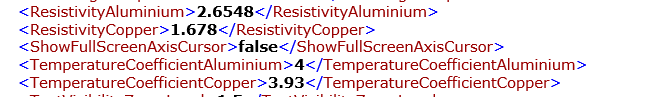
# Wire resistance calculator and Material Spec

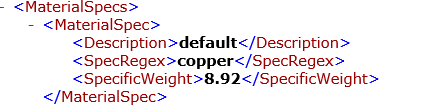
Today the wire resistance calculator uses predefined materials as copper and aluminum from the general settings xml. Copper is always proposed, the user can change to aluminum or a custom material and enter a resistivity value.

The weight calculator was added and this uses now a material spec expression to handle the specific weight and now the user would expect that the material is also proposed properly in the resistance calculator.

The problem is, that the specific resistance and the temperature coefficient needs to be moved from the general settings to the weight settings xml. This will affect also the display of the material in the drop down of the calculator.

We still need to handle the case, that there are no material specs used or set and therefore the values in the general settings must remain for the fallback.





Proposal:

On a selected wire, the calculator can be called.

The values in the material drop down list should be filled with the defaults copper, aluminum and <custom> from the general settings, and all material spec. entry descriptions from the weight settings (best would be to fill only conductors, so we could detect this by the resistivity value). Beware, the default entries are localized. In case the description is used multiple times or collides with the predefined entries, we just add them and get two entries with the same display name. If the material spec is found, the corresponding entry needs to be selected and the appropriate resistivity value is to be displayed in the box. (what happens if there are two entries with the same name? take first matching then)

If the material spec is not found, the default copper is used - first entry in the list.