**Representation of Wires and Cores**

#### History

|  |  |  |
| --- | --- | --- |
| Name | Changes | Date |
| Kliemannel | Initial creation | 22.08.2013 |
|  |  |  |
|  |  |  |

#### Affected Tools

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EE-Browser | FUNCoSAR | HARcad | TOPcad | VESADO |
|  |  |  |  |  |

#### Table of contents

[1. Use Case 1](#_Toc364933734)

[1.1. Description 1](#_Toc364933735)

[1.2. Objects 2](#_Toc364933736)

[1.3. Object Attributes 2](#_Toc364933737)

[2. Detailed Description 2](#_Toc364933738)

[3. Graphical User Interface (GUI) suggestion 2](#_Toc364933739)

# Use Case

Discussions with costumers have shown that the representation of cables and wires in separate tabs is not sufficient.

The separation confuses the user when he wants to:

1. Identify connections of a connector
2. Identify connections in a segment
3. Identify connections of a module
4. Show all connections of harness

In every case he needs to know if the connected objects are wires or cables. If it is about a mix of cables and wires, he is not able to get both in one view.

The “show wires/cores dialog” was also discussed with customer.

The dialog doesn’t fit in the common EE Browser workflow with data tables, drawing, cross selection, context menu functionalities.

We discussed the three following alternatives to provide a sufficient solution for this problem.

## Show cores in wires tab:

This will be like it is in EE Browser 5.5 with an additional cables tab.

Advantages:

* Best integration in common EE Browser workflow
* Most user friendly, best to achieve the mentioned use cases
* Most accepted by costumer

Disadvantages:

* Wires and cores are different classes in KBL and so they have different attributes. The different attributes must be shown in the same tab/table. I think they are separate classes in our data model too. This means we need a proxy class to redirect some attributes. See 1.1.1 for details.
* Triggering any functionality on a combined tab needs differ between the objects that are selected, e.g. context menu will under circumstance look different for core and for wire. For example cores have a “show corresponding cable function”.   
  Do we need an additional context menu for mixed selection of wires and cores with overlapping functionalities? That may confuse the user.
* Redundant representation of cores. The representation on wires tab and on cables tab will even look different, because of the different attributes shown. This may have some influence on cross selection between data table tabs (show cores? on which tab). Cross selection between drawing and data table is not affected.  
  To avoid this Problem we may remove the cores from the wires tab. This will match with our common Workflow. If user wants to see core details he uses “show corresponding cores” on cable and vice versa.

### Comparison of wire and core attributes

|  |  |
| --- | --- |
| Attributes of wires tab | Analog core attribute classes |
| Wire number | From core\_occurrence |
| Cross section area | From core |
| Color | From core |
| Length information [dmu] | From core\_occurrence |
| Start connector | from connection |
| Start cavity | from connection |
| End connector | from connection |
| End cavity | from connection |
| Net name | from connection |
| Part number | from cable (General\_wire) |
| Description | from cable (General\_wire) |
| Abbreviation | from cable (General\_wire) |
| Company name | from cable (General\_wire) |
| Alias id | from cable (General\_wire) |
| Version | from cable (General\_wire) |
| Predecessor part number | from cable (General\_wire) |
| Degree of maturity | from cable (General\_wire) |
| Copyright note | from cable (General\_wire) |
| Mass information | from cable (General\_wire) |
| External references | from cable (General\_wire) |
| Change | from cable (General\_wire) |
| Material information | from cable (General\_wire) |
| Processing information | from cable (General\_wire) |
| Installation information | from cable occurrence (Special\_wire\_occurrence) |
| Cable designator | from core |
| Wire type | from core |
| Bend radius | from core |
| Outside diameter | from core |
| Assigned modules | from cable occurrence (Special\_wire\_occurrence),  e.g. module pointing on cable occurrence |

## Show cables in wires tab

Show the cables in wires tab exactly like they are shown in cables tab, e.g. with the tree functionality for grouping cores.

Advantages:

* Same attributes for cables and wire, except Wire\_number/Special\_wire\_number.

Disadvantages:

* Again there are different functionalities for wires and cables (example: show start end connectors), so that we have to differ between selected objects. See 1.1 for details.
* To fulfill the use cases we need a mixed multi selection of cores and wires. Which means we need to select objects of different tree levels at the same time, I think that’s not possible at the moment.
* Circuitous for the user.

## Show an additional Connection tab

Derived from the KBL classes we could show a new tab called connections. Connections are the same for wires and for cores and they fulfill the mentioned use cases

Advantages:

* Doesn’t change current behavior
* Fits into common EE Browser workflow
* Accepted by customer
* No problems with different classes/proxy class and different functionalities in the same tab

Disadvantages:

* Only a little information on this tab, due to the attributes of connection class: Id, description, Signal\_name, Installation\_information, Start\_connector & cavity, End\_connector & cavity.  
  Important information like cross section area, wire color and length won’t be part of this tab.
* Very circuitous for the user. He needs to go through two tabs to get most of the information.
* Doesn’t really solve the problem. User needs also to know if connection is for cable or for wire to get additional information.