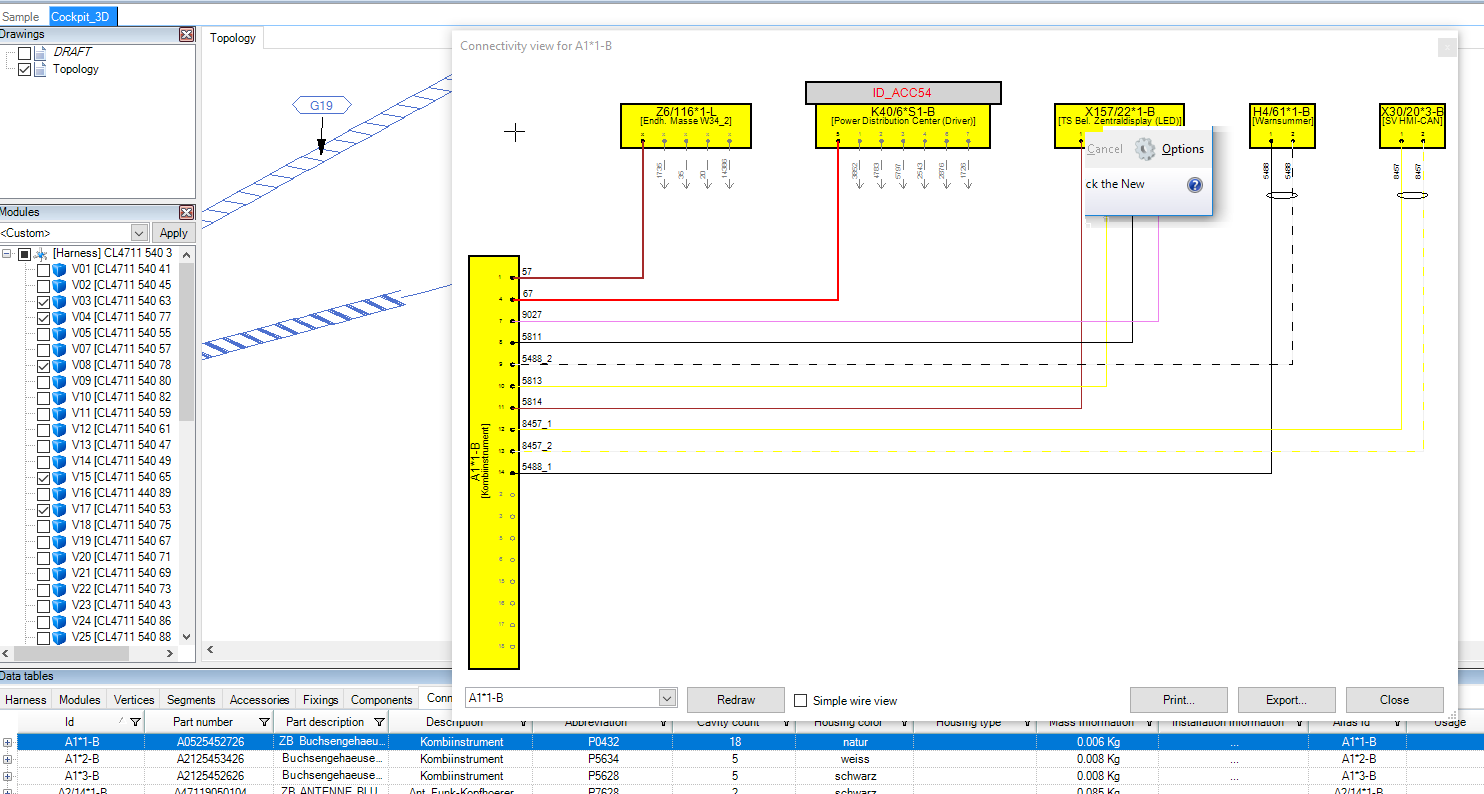
# Einleitung

Als Zusatzmodul zu dem existierenden Produkt E3.HarnessAnalyzer ist die Integration eines Connectivity-Viewers auf Basis von NL-View geplant.

Dieses Zusatzmodul soll als kostenpflichtige Option über Lizenz-Code aktiviert werden können und sowohl für existierende HA-Kunden interessant sein (Nachrüst-Geschäft) als auch das Neukundengeschäft bestärken durch eine Funktionserweiterung in einem Bereich, der bisher nicht bzw. nicht in diesem Umfang abgedeckt ist.

Ausgehend von einem ausgewählten Stecker gibt es heute die Möglichkeit, sich die Verbindungen dieses Steckers durch eine Art Schaltplanskizze zu visualisieren:



Dies ist eine Standardfunktion des HA seit Jahren und bleibt auch mit Verfügbarkeit des neuen Moduls unverändert erhalten. Dieser „Connectivity-Viewer - Light“ ist also eine Funktionalität, die der Kunde „umsonst“ erhält, während er für den „Connectivity-Viewer - Pro“ eine Zusatz-Lizenz kaufen muss. Dieser Umstand erklärt auch den Grund, dass wir mit dem „Connectivity-Viewer - Pro“ substantielle Mehrwerte bieten müssen.

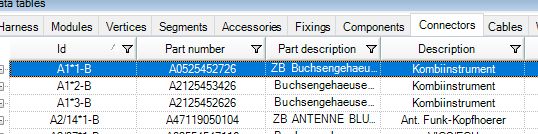
# Funktionalität der Views

Die Funktionalität ist auf absehbare Zeit durch den Daten-Content limitiert, der mit KBL maximal transportiert werden kann (wobei zu beachten ist, dass selbst dieser Umfang in der Praxis häufig nicht wirklich ausgelebt wird und vorhandene Felder gar nicht genutzt, falsch genutzt oder unzureichend gepflegt werden.

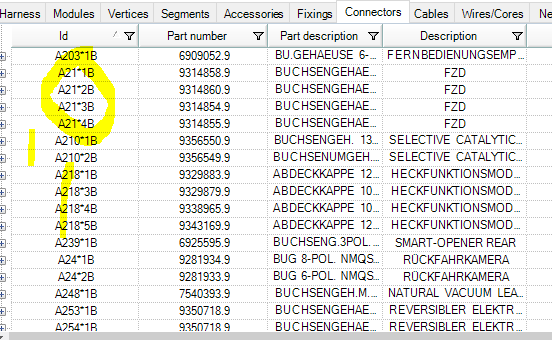
Basis ist auf alle Fälle die Konnektivität, die durch ein oder mehrere geladene KBL´s transportiert wird. Eine Interpretation von Daten sollte auf möglichst geringen Level gehalten werden, da dies viel kundenspezifische Konfiguration mit sich bringt.

Nicht umhin kommen wir allerdings bei der Generierung von virtuellen Komponenten: In der Regel werden in den KBL´s nur die Stecker übertragen, jedoch nicht die ECU Komponenten als solches. Um aber vernünftige Views zu erzeugen muss im Vorprozess (vor dem Laden der EDB) über Namenskonventionen aus den Steckern passende Komponenten erzeugt werden.

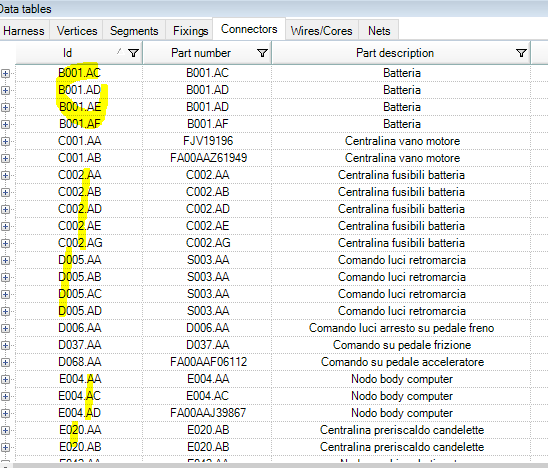
Im Umfeld Daimler muss z.B. eine Komponente A1 – Kombiinstrument erzeugt werden, welche die drei Stecker A1\*1B, A1\*2B und A1\*3B trägt.



Im Umfeld von BMW ist die Nomenklatur zufällig gleich:



Im Umfeld von Fiat hingegen nutzt man einen anderen „Trenner“

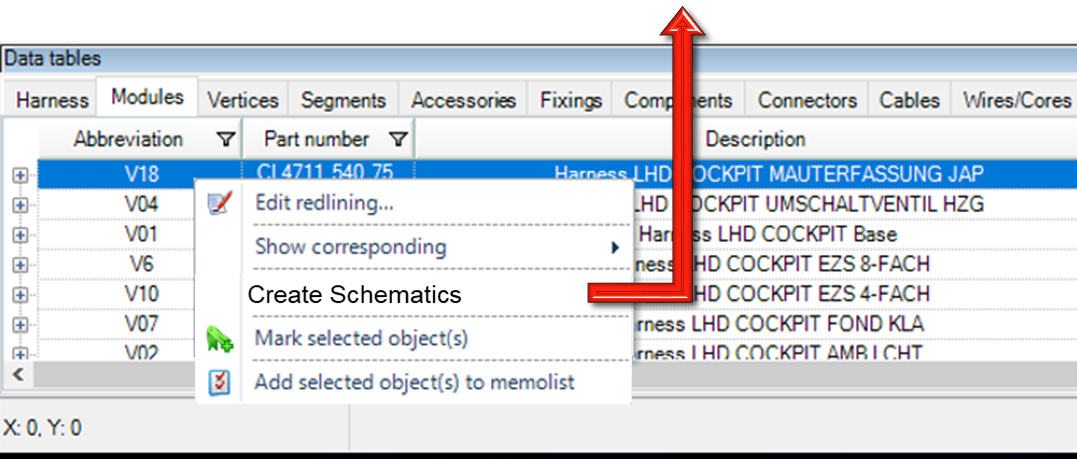


Eine Identifizierung bzw. Erkennung von Splices, Eyelets und Inliner über kundenspezifische Nomenklatur wird zusätzlich erforderlich sein, um entsprechende Darstellung in den Schaltplänen zu erreichen.

## Arten von Sichten

### Modul-Schaltpläne

Ausgehend vom Modul-Grid eines Kabelsatzes wird am selektierten Modul dafür mit rechter Maus das Kontext Menü aufgerufen.



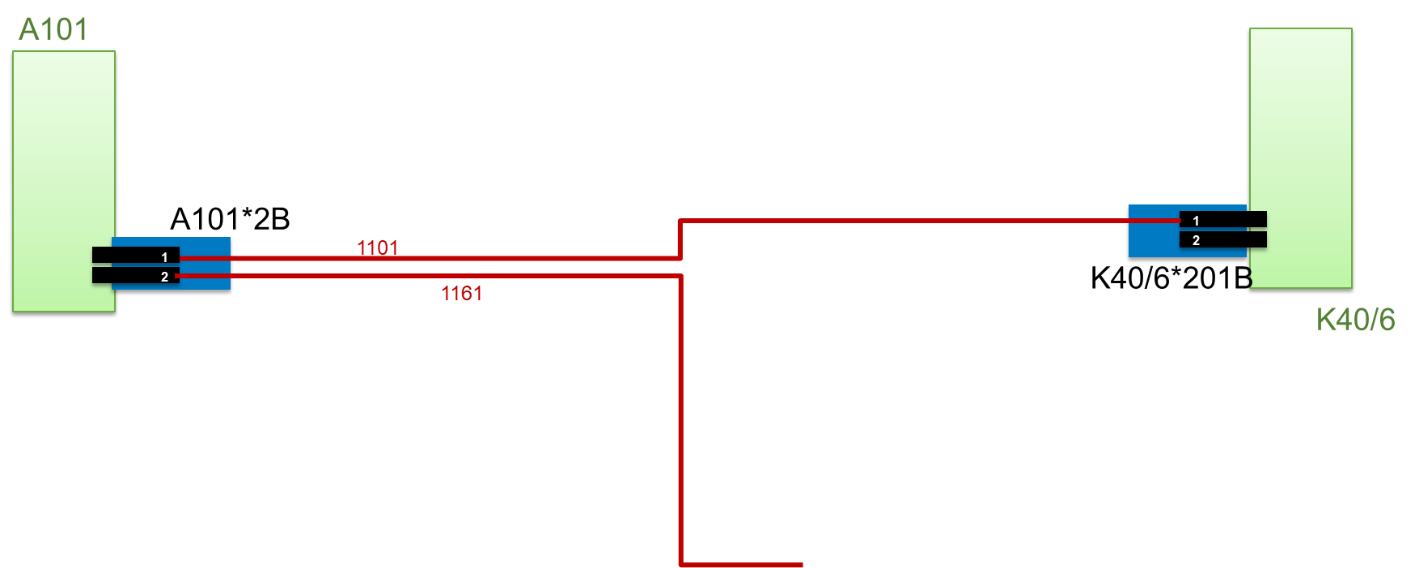
Als Ergebnis soll nun für den Umfang dieses ausgewählten Moduls ein Schaltplan erstellt werden – und zwar als fertiger „Komplett-Plan“, kein iterativer.

Anmerkung: Wir die Funktion für ein Grundmodul eines großen Kabelsatzes aufgerufen, so muss zwangsläufig ein schwerst lesbarer Plan herauskommen. Da ist sowie OK, dafür ist die Funktion auch nicht gedacht, sondern für die Visualisierung von optionalen Zulegemodulen. Falls es bei Basis-Modulen zu Performance-Problemen kommen sollte, könnte der User ab einer definierten Schwelle auch vorher darauf hingewiesen werden.

Module ergeben im Kabelsatz oft erst einen Sinn in der Kombination mit anderen Modulen. Daher scheint es nicht sinnvoll, nur den nackten Modul-Umfang darzustellen, sondern auch die unmittelbare Nachbarschaft als ausgegrauten Neben-Objekte.

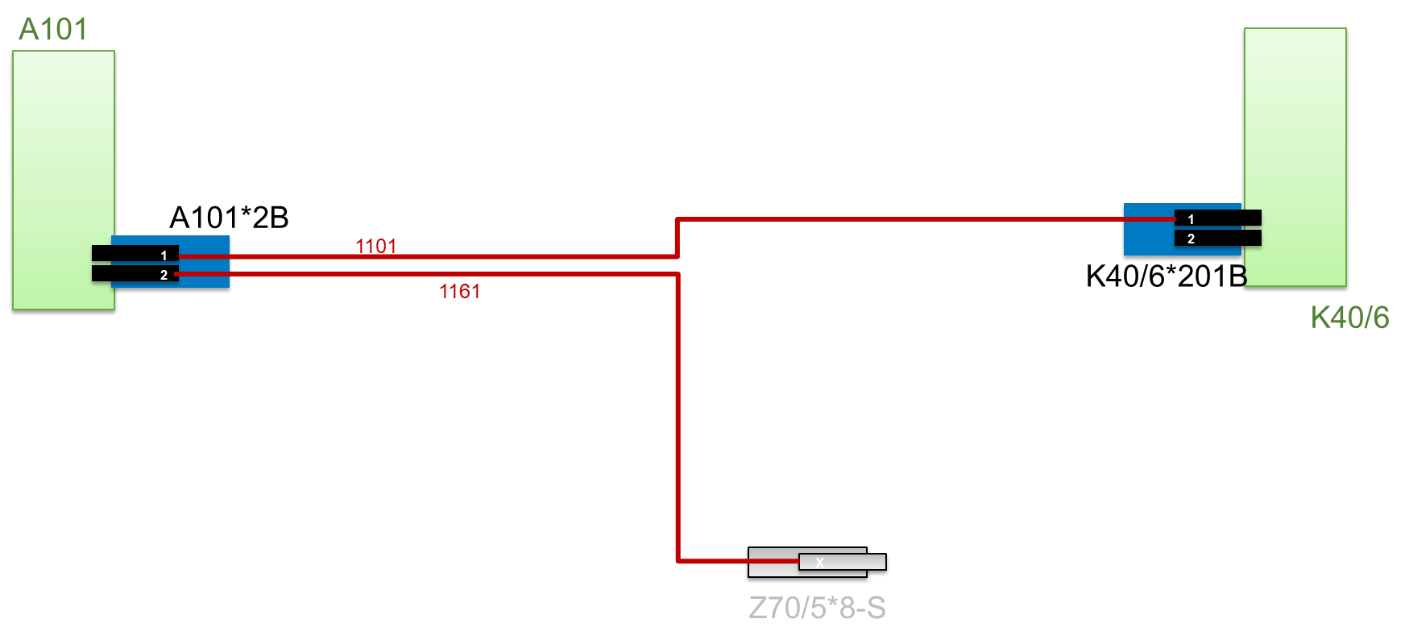
Angenommen ein Modul besteht aus dem Stecker A101\*2B und K40/6\*201B sowie den Leitungen 1101 und 1161.

Stellen wir nur exakt diesen Umfang dar, so würde folgende unvollständige Schaltung erscheinen:



d.h. Draht 1161 würde in der Luft hängen.

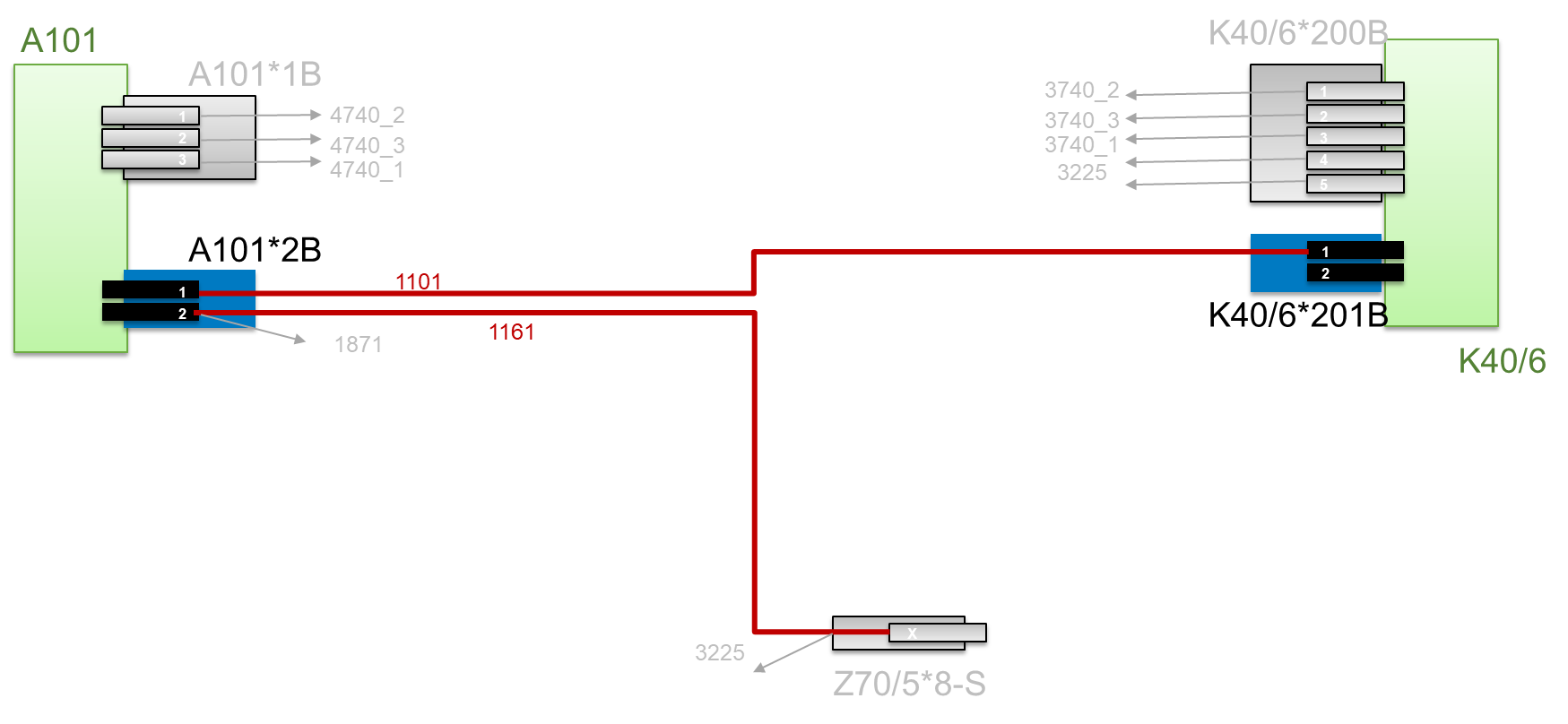
Somit gebietet es sich zumindest, die angeschlossenen Stecker / Komponenten mit zu visualisieren, selbst wenn diese zu einem anderen Modul gehören.



Die nicht zum Modul gehörenden Objekte werden gegraut um eine klare Differenzierung zu den aktiven zu erhalten.

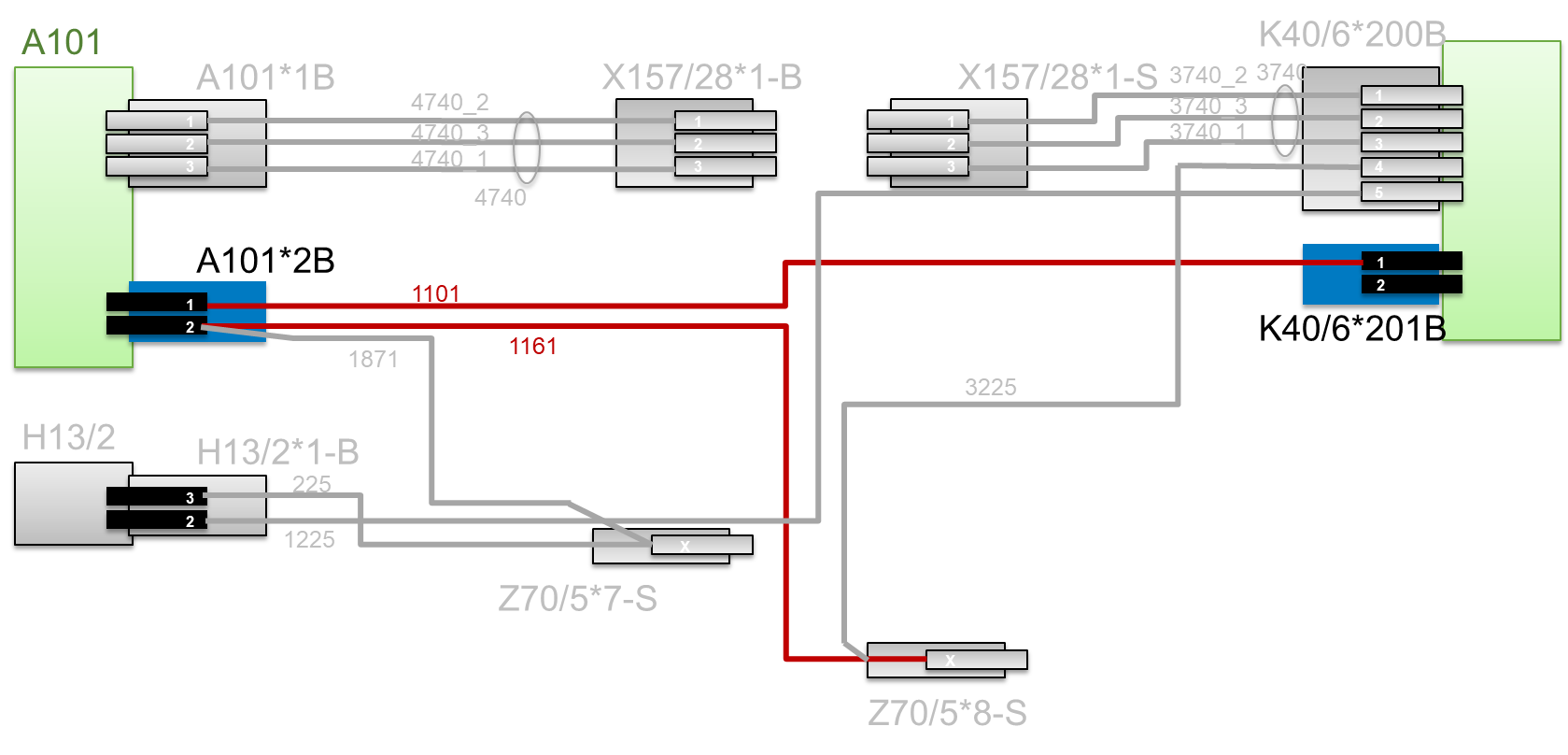
Die Frage ist natürlich noch, wie weit man diese Darstellung extrapolieren soll um nicht aktive Elemente um die Verständlichkeit zu erhöhen.

Eine erweiterte Stufe wäre

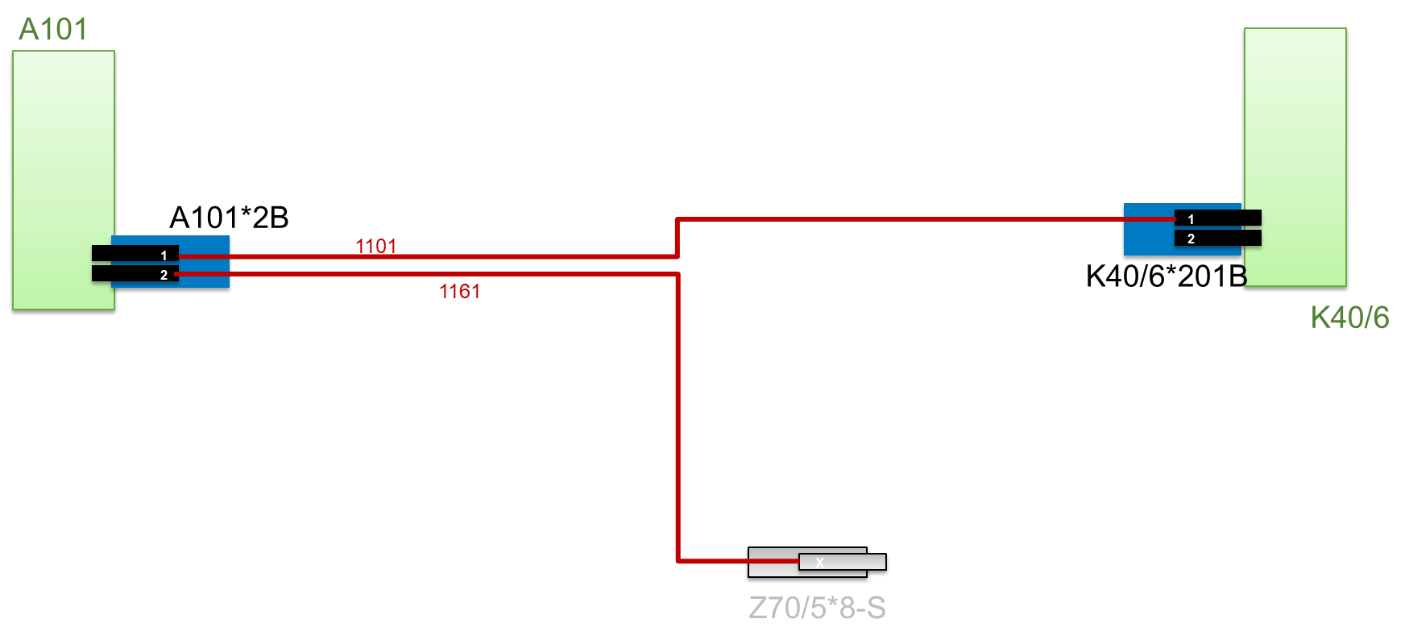


Hier werden alle Stecker einer Komponente dargestellt und die daran angeschlossenen Leitungen als „Verweise“.

Stellt man diese aktive Verbindungen komplett dar, so würde man landen bei



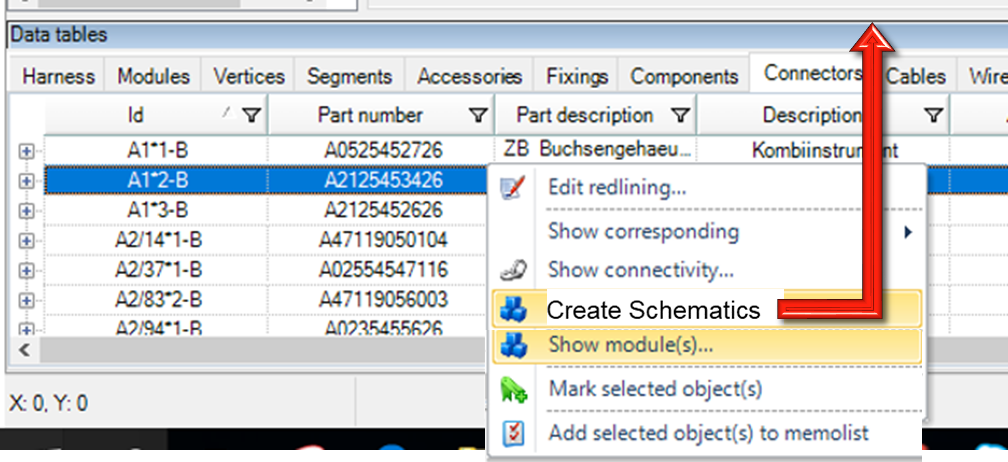
**Vielleicht liegt der Königsweg darin, mit einer Darstellung wie**



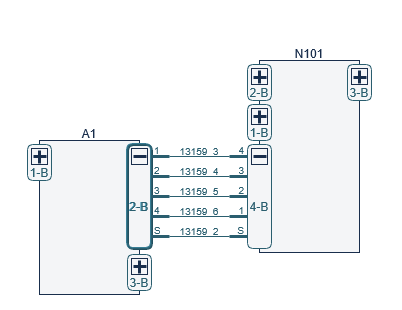
**zu starten und dann doch eine Interaktion anzubieten mit der der User den Umfang erweitern kann. Dabei werden alle nicht dem Modul „gehörenden“ Elemente jedoch immer gegraut.**

### Komponenten-Schaltpläne (Kabelsatz-übergreifend)

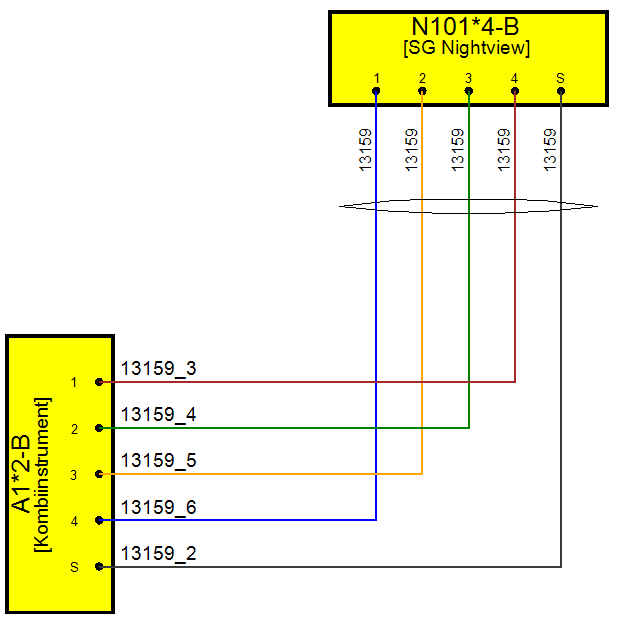
Einstiegspunkt für diese View ist wieder das Grid im HarnessAnalyzer, diesmal die Connector-View



Darauf eröffnet sich eine Schematics-View die diesem ausgewählten Stecker im Mittelpunkt und geöffnet:



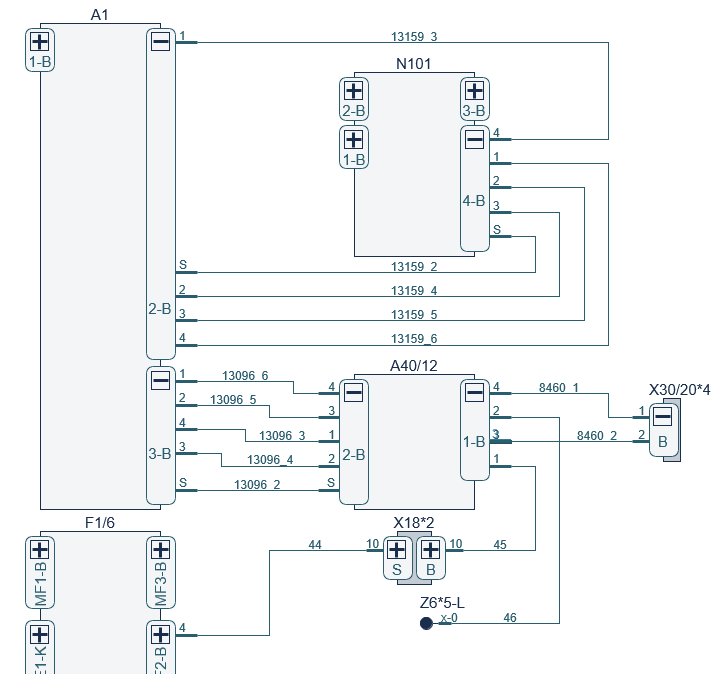
Dies entspricht erst einmal der heute im „Connectivity-Viewer - Light“ verfügbaren Funktionalität.



Die klare Differenzierung liegt nun aber in folgenden gewichtigen Nutzen-Aspekten:

* Der „Connectivity-Viewer - Pro“ kann interaktiv die Schaltung erweitern und erlaubt den User damit durch die Schaltung zu navigieren und zu „browsen“
* Der „Connectivity-Viewer - Pro“ hat die Gesamt-Connectivity aller geladenen KBL´s im Zugriff und kann dadurch über Kabelsatzgrenzen hinweg visualisieren

Ausgehend von dem ursprünglichen Stecker lande ich dann in einer Connectivtät, die sich interaktiv erschließen läßt.

* 

Konsequenz / ist hier natürlich, dass dem Viewer mit dem Aufruf eines Steckers aus einem Kabelsatz heraus eine Gesamt-Connectivity zur Verfügung gestellt werden muss.

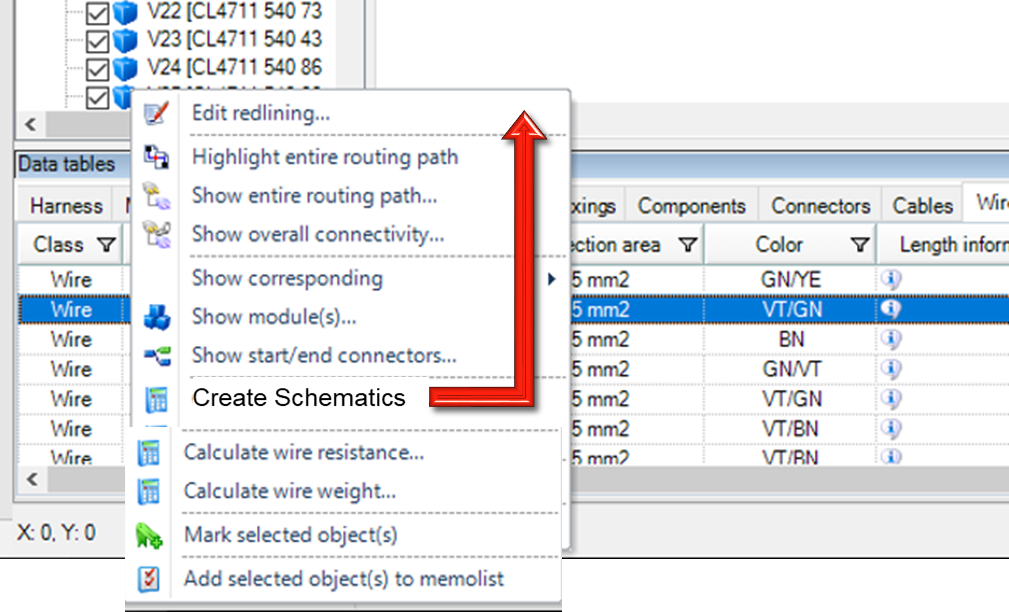
In diesem View haben wir die Konstellation dass es Drähte gibt, die aufgrund der aktuell aktivierten Modulkonfiguration nicht aktiv sind. Allerdings handelt es sich um eine etwas anders gelagerte „Nicht-Aktivierung“ als beim Modul-View. Beim Modul-View gibt es einen Umfang der zu dem Modul fest gehört und den (ausgegrauten) Rest.

Beim Komponenten-View gibt es Leitungen und Komponenten, die in der aktuellen Konfiguration nicht „scharf“ sind, aber in den 150% durchaus vorhanden sind. Als Lösung bietet es sich an, hier nur den Umfang anzuzeigen, der in der jeweiligen Konfiguration „scharf“. Wird die Konfiguration während eines offenen Schema-Views umgeschaltet, so wird der Content des angezeigten Views angepasst. Alternativ könnte man auch hier eine Darstellung des 150% Contents anbieten und auch hier die geblockten Objekte farblich downlighten – allerdings anders als im Block-View

### Funktions-Schaltpläne (Kabelsatz-übergreifend)

Leider bildet die Struktur der KBL keine saubere Funktions-Gruppierung. Da eine solche Sicht aber überaus interessant wäre, sollten wir zumindest einen WorkAround hierzu anbieten:

Einstiegspunkt für diese View ist wieder das Grid im HarnessAnalyzer, diesmal der Wire-View

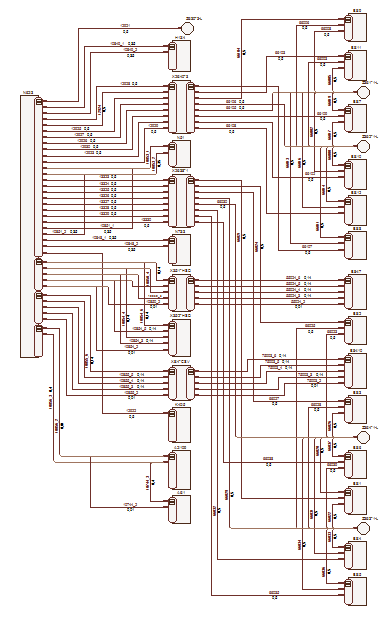


Abhängig vom selektierten Draht gibt es jetzt folgende use-Cases:

* Es ist nur ein Draht selektiert und es gibt keinen anderen Draht, der in der Spalte „Description“ den gleichen Eintrag hat => Öffnen eines Views mit diesem Draht im Mittelpunkt
* Es ist nur ein Draht selektiert, aber es gibt weitere Drähte, die in der Spalte „Description“ den gleichen Eintrag haben => den User fragen ob er nur den selektierten Draht haben will oder alle Drähte aus der Gruppe „Description“ (das Feld kann als Funktionskenner verwendet werden. Wenn die Antwort „alle“ 🡺 Öffnen eines Views mit allen Drähten dieser Description-Gruppe als Mittelpunkt
* Es sind bereits mehrere Drähte selektiert, 🡺 Öffnen eines Views mit allen selektierten Drähten als Mittelpunkt

Wenn sich der View öffnet, so werden alle Elemente dargestellt, die durch die selektierten Drähte verbunden sind.

Wurde ein Description-Kenner sauber durchgezogen, so würde ein kompletter System View entstehen wie



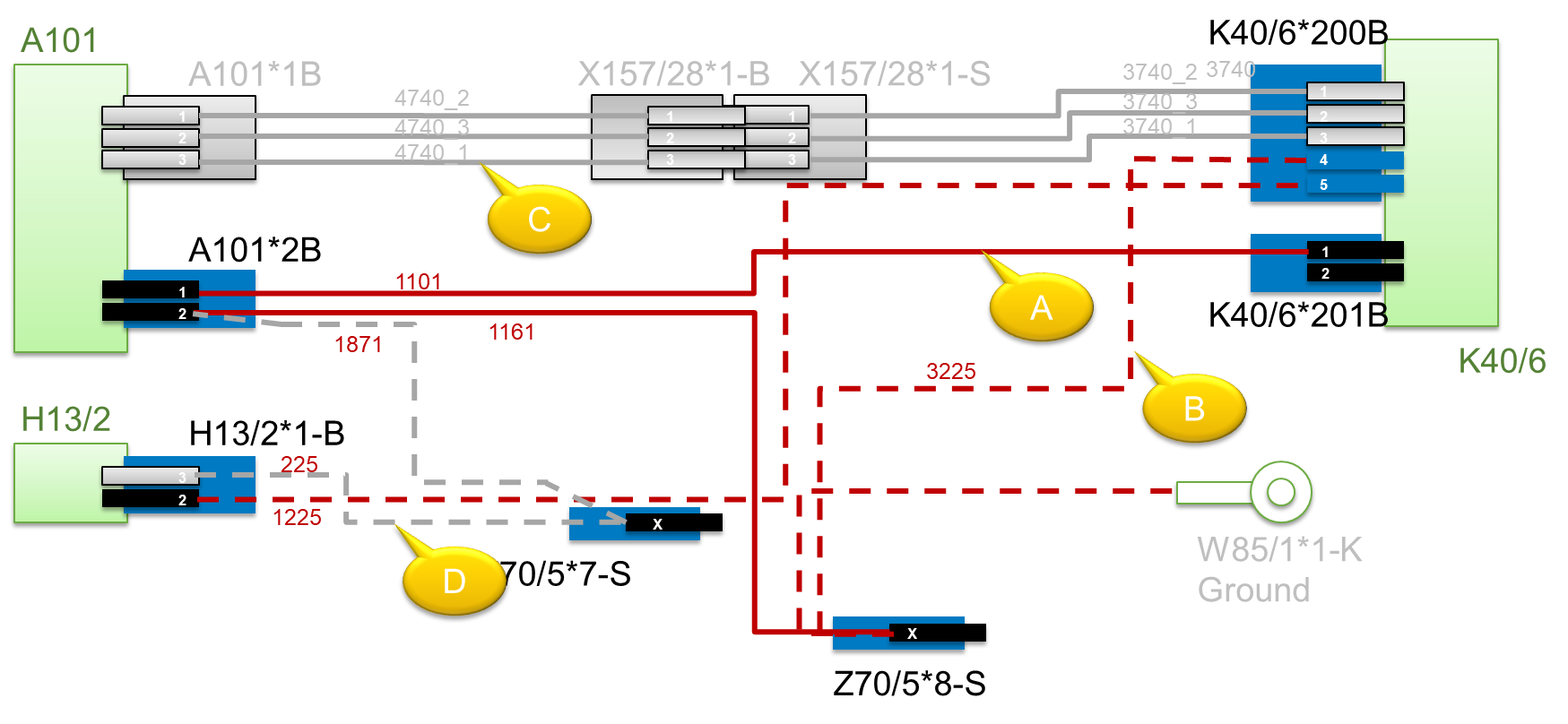
Auch hier ergibt es die Konstellation, dass es Drähte in dieser Gruppe geben wird, die in der derzeit aktivierten Konfiguration deaktiviert sind. Um die Funktion aber umfänglich darzustellen, sollten wir auch diese Drähte mit anzeigen – farblich so gekennzeichnet wie unter Component-View angeregt.

Und auch hier stellt sich die Frage, ggf. über den Umfang dieser Funktion interaktiv hinaus gehen zu wollen.

Dies führt zu dem Ansatz, letztlich einen View zu generieren, der alle 3 Funktionsmerkmale abdecken kann:

* Der View startet immer in einem definerten Grund-Umfang
  + Bei Modul: Modul-Content
  + Bei Component: Selectierter Connector mit direkt angeschlossen Verbindungen
  + Bei Function: Alle Verbindungen analog einer Wire-Liste
* Im View wird das Blocking entspr. Konfiguration berücksichtigt und geblockte Elemente werden nur gegraut dargestellt. Zusätzlich sollte es die Möglichkeit geben, die 150% auch ganz verschwinden zu lassen und auf eine echt aktivierte Konfiguration zu reduzieren
  + Bei Modul: kommt das nicht vor
  + Bei Component: kommt vor
  + Bei Function: kommt vor
* Im View hat der User immer die Möglichkeit, durch Interaktion sich weiter über die ursprüngliche Darstellung / Content zu bewegen. Es kommen dabei Elemente hinzu, die nicht zu dem ursprünglich selektierten Modul oder Funktion gehören und diese sollten ebenfalls graphisch unterschiedlich dargestellt werden
  + Bei Modul: kommt vor
  + Bei Component: kommt nicht vor
  + Bei Function: kommt vor

Das würde bei einem Functional-View im Extrem-Szenario zu einer Situation mit vier Stati eines Objektes führen mit entspr. graphischer Repräsentation



A: Objekt ist aktiv in der gewählten Konfiguration und gehört zu der ursprünglich ausgewählten Funktion / Modul

B: Objekt ist aktiv in der gewählten Konfiguration und gehört aber nicht zu der ursprünglich ausgewählten Funktion / Modul - d.h. erst später interaktiv dazu gekommen

C: Objekt ist inaktiv in der gewählten Konfiguration und gehört aber zu der ursprünglich ausgewählten Funktion / Modul

D: Objekt ist inaktiv in der gewählten Konfiguration und gehört auch nicht zu der ursprünglich ausgewählten Funktion / Modul - d.h. erst später interaktiv dazu gekommen

Wäre das für den User verständlich ?

Oder wäre es besser, in der Darstellung nur zwei Modi zu fahren:

* Objekt gehört zu der ursprünglich ausgewählten Funktion / Modul
* Objekt gehört nicht zu der ursprünglich ausgewählten Funktion / Modul

Und die Schaltung der „150%“ <> „aktiv in der gewählten Konfiguration“ durch einen getrennten Button alternativ zu steuern

# Spezielle Aspekte

### Modal / Non-Modal

In den o.G. Sichten wird ausgehend vom Grid im HarnessAnalyzer ein Fenster mit der Schaltplan-Sicht geöffnet.

Im Gegensatz zum „Connectivity-Viewer - Light“ soll dieses Fenster im „Connectivity-Viewer - Pro“ **nicht-modal** sein. D.h. das Fenster soll „stehen bleiben können und der User soll sich parallel in anderen Fenstern (2D, 3D oder Grid) bewegen können. Dabei soll es die gleichen Gross-Highlights geben wie zwischen den existierenden Fenstern.

### Aktualisierung

Wird z.B. ein Konnektivitäts-View für das Modul M3 gestartet und ggf. interaktiv erweitert, so bleibt dieser Content unverändert, selbst wenn der User in der Zwischenzeit im Grid das Modul M5 selektiert um sich z.B. über dieses Modul zu informieren. Der Konnektivitäts-View wird nur dann neu aufgebaut, wenn der User dies durch das entsprechend RM-Menü neu aufruft.

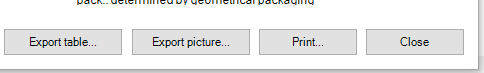
### Bedienlogik / GUI

Für den neuen Konnektivitäts-View sollten wir nach Möglichkeit die gleiche Bedienlogik zur Verfügung stellen wie in den bestehenden Sichten (2D und 3D). Dies bedeutet u.a.

* Zoom über Mausrad
* PAN über gedrückte Mausrad-Taste
* Keine Strokes (da diese sonst ggf. auch für 2D/3D gefordert werden)
* Nicht ganz wichtig, aber „perfekt“ wäre auch Kompatibilität für Tablet-Bedienung
  + Zoom
  + PAN
  + Last View
* Multi-Language Unterstützung – falls wir in diesem Fester Text-Menüs anbieten

Drucken / Export

Die existierenden Views im HarnessAnalyzer zeichnen sich dadurch aus, dass sie ein Weiterprozessen der Daten unterstützen.

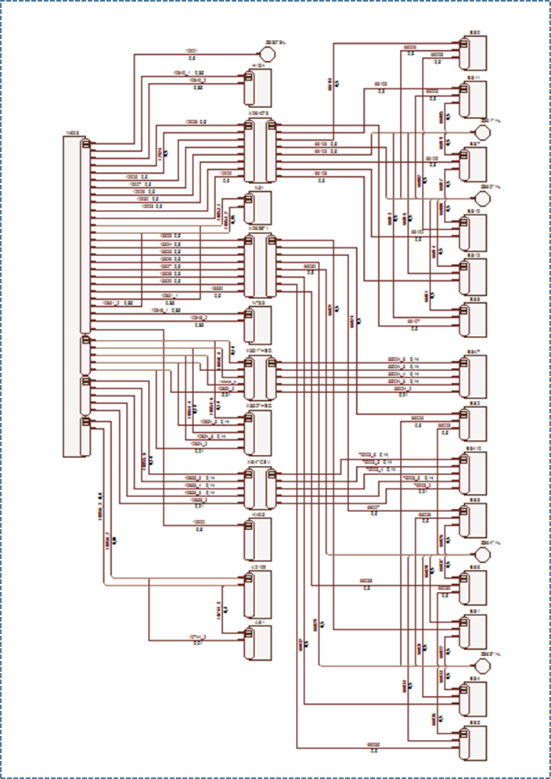


Für den „Connectivity-Viewer - Pro“ sollten wir nach Möglichkeit diese Tradition fortsetzen und ebenfalls einen Export und eine Druckfunktion anbieten.

Für den Export wäre wohl PDF das perfekte Mittel der Wahl, ggf. auch SVG.

Das Drucken könnte entweder eine direkte Druckfunktion sein oder als Workaround auch der Umweg über PDF-Generierung.

Sowohl bei PDF als auch Drucken stellt sich dann natürlich die Formatierungsfrage. Denn für beide gibt es letztlich einen Layout-Constraint in Form eines DIN-Format in Quer oder Hoch



Virtuelle Format-Grenze die das Verhältnis 1:Sqr(2) abdeckt.

Beim Drucken oder PDF kann dieses virtuelle Format dann einfach auf das gewählte DIN Format (A4 oder A3) skalliert werden

Die Frage ist, ob NL-View als Standard-Funktion dieses Verhältnis optional schon als Lay-Out – Constraint verwenden kann, um z.B. lange vertikale oder horizontale Ketten zu vermeiden.

Falls wir diese Komfortfunktion anbieten wollen, so wäre natürlich auch noch zu beachten, dass wir z.B. in USA leicht abweichende Formatverhältnisse haben (Legal, Letter)