



## Konzept Variantenkodierung

Franz Lohberger Visu-IT!, Feb 2010

- **Variantenkodierung:** Der Begriff Variantencodierung bedeutet die Anpassung von Steuergeräte Eigenschaften zur Laufzeit an Fahrzeug oder Motoreneigenschaften
- **Selektor:** Ein Variantenselektor ist eine Onlinegröße (Variable) im Steuergerät, deren Zustand die aktuell gültige Ausprägung einer Variante widerspiegelt
- **Ausprägung:** Als Variantenausprägung werden die unterschiedlichen Zustände eines (Varianten-)Kriteriums bezeichnet. Sichtbar werden die Ausprägungen durch die möglichen Werte des Variantenselektors

Beispiel: In einem Fahrzeug werden sowohl Handschalter (HS), als auch Automatikgetriebe (AT) verbaut

- Variantenkriterium: Getriebetyp
- Variantenausprägungen: HS, AT
- Variantenselektor: VarCod\_stGbxTyp  
(kann verschiedene Werte entsprechend den Ausprägungen annehmen)

## Ziele

- Modell und Funktionscode dürfen **nicht verändert** werden. Damit sind auch nachträgliche Variantenkodierungen an existierenden Funktionen möglich
- Die variantencodierten Größen sollen im A2L File/der Dokumentation sichtbar und als solche **erkennbar** sein
- Alle Ausprägungen der Größen sollen gleichzeitig applizierbar sein
- Während der Applikationsarbeit muss im Klartext in INCA bzw CRETA erkennbar sein, welche Variante gerade bearbeitet wird (z.B. HS/AT)
- Mehrfache Varianten einer Größe müssen unterstützt werden (z.B. HS Combi, AT Cabrio)
- Das Kopieren von variantenkodierten Größen gleicher Größe aber unterschiedlicher Bedeutung zwischen verschiedenen Projekten soll nicht möglich sein.

### Beispiel:

PARAM\_1 ist in den Projekten A bzw. B mit verschiedenen Varianten codiert

- Projekt A: PARAM\_1 ist über Getriebe HS und AT codiert
- Projekt B: PARAM\_1 ist über Antrieb 4WD und FRNT codiert  
=> PARAM\_1 soll/darf nicht von Projekt A nach Projekt B kopiert werden, da die Bedeutung der applizierten Daten in PARAM\_1 nicht identisch ist.

# Visu-IT! – Konzept Variantenkodierung

## Variantenkodierung in ADD

In ADD erfolgt die Variantenkodierung in 3 Schritten:

1. Schritt 1: Definition der **Variantenselektoren** mit den jeweiligen Varianten (und mit allen möglichen Ausprägungen)
2. Schritt 2: Definition der im Projekt **genutzten Varianten** inklusive der im Projekt verwendeten Ausprägungen
3. Schritt 3: Definition der **variantenkodierten Größen** im Projekt (Auswahl der vorhandenen Größen und Zuordnung der gewünschten Variante(n) aus Schritt 2)

# Visu-IT! – Konzept Variantenkodierung

## Schritt 1 – Definition der Variantenselektoren

- Bearbeiten des „VarCod“ Container nur durch einen eingeschränkten Personenkreis (Rolle Variantenmanager)
- Im Container „VarCod“ werden nur Outputs (Variantenselektoren) produziert. Jede dieser Outputgrößen steht für eine bestimmte Kodierung und dient bei der Initialisierung als Auswahlkriterium für das entsprechende Element der kodierten Größe

The screenshot displays the software interface for defining a variant selector. On the left, a 'Container' window shows 'VarCod (V1.1.0 - draft)' with an 'OUTPUT' table listing several variant selectors. The main window shows the 'Data Object' 'VarCod\_stGearTyp (1.0, draft)' with a table of 'Data Object Versions'. Below this, the 'Variant Criterion' tab is active, showing fields for 'Variant Name' (Gear), 'Variant Desc.' (Type of gear box), and 'Coding Number' (13). A table of 'Values' lists 'AT' (Automatic) and 'HS' (Manual). The 'Variant Criterion' tab is highlighted with a yellow circle.

T...	Long Name	SIM Data Type	Phys. Unit	Base Type	S
ONL	1.0	Variant Selector for ty...	RATIONALFUNCTION_...	-	FLOAT32... 0

Variant Name:	Value	Description
Gear	AT	Automatic
Type of gear box	HS	Manual



# Visu-IT! – Konzept Variantenkodierung

## Schritt 2 – Definition der genutzten Varianten

Auf **Projekt**-Ebene:

- Auswahl der benötigten Variantenselektoren
- Setzen der im Projekt verwendeten Ausprägungen zu den Werten aus dem Wertebereich entsprechend der Zuordnungsliste

VIT\_PRO (V1.2.0 - draft)

Project Definition | Project References | System Constants | **Variant Selector Configuration** | Data Variants

Name	Version	State	Value	Variant Criterion Va...
ONL VarCod_stCaseTyp	1.0	draft	0	Limo
			1	Kombi
			2	Cabrio
ONL VarCod_stExTyp	1.0	draft	0	EU4
			1	EU5
ONL VarCod_stFuTyp	1.0	draft	0	Diesel
			1	Gas
ONL VarCod_stGearTyp	1.0	draft	0	AT
			1	H5

Varianten-  
selektor

Verfügbare  
Varianten

Daten  
Varianten

# Visu-IT! – Konzept Variantenkodierung

## Schritt 3 – Definition der Daten Varianten

Auf **Projekt**-Ebene:

- Auswahl der Parameter/Kennfelder, die variantenkodiert werden sollen
- Zuordnung der Variantenselektoren über die kodiert werden soll

VIT\_PRO (V1.2.0 - draft)

Project Definition | Project References | System Constants | Variant Selector Configuration | **Data Variants**

Name	Version	State	Variant Selector
IP_AMP_8	1.0	fixed	VarCod_stCaseTyp (1.0) VarCod_stExTyp (1.0)
Veh_bAdd_cpAdh_M	1.2	fixed	VarCod_stFuTyp (1.0)
VehPen_arAdh_cpAdp_T	1.0	fixed	VarCod_stFuTyp (1.0) VarCod_stGearTyp (1.0)

Varianten-  
selektor

Verfügbare  
Varianten

Daten  
Varianten

# Visu-IT! – Konzept Variantenkodierung

## Variantenkodierung - Toolkette

Die in ADD **zentral** definierte Variantenkodierung kann **automatisiert** und **transparent** in den weiteren Prozessschritten verwendet werden:

- Simulation / Modellebene
- SW Entwicklung
  - Handcode (e.g. via DDS)
  - Autocode via Ascet
  - Autocode via TargetLink
- A2L File Generierung
  - ELF/I3E, CVX/DCM
- Datenaustausch e.g. MDX
- Funktionsdokumentation