

# AP212-KBL

## Ein Standard durchleuchtet

Dipl.-Ing.  
**Siegfried Eineder**  
Softwareingenieur



## Was ich Ihnen zum Thema KBL vermitteln will

- Entwicklungsziele
- Einsatz und Anwendung
- Genereller Aufbau
- Wesentliche Elemente
- Leistung und Grenzen



## Historisches 1 Problemstellung

- Proprietäre Systeme
- Datenaustausch OEM/Zulieferer
- Automatisierung der Prozessketten
- Weg vom Papier
- Langzeitarchivierung



## Historisches 2 VDA-Arbeitskreis

- Wunsch nach Standardisierung
- Hersteller und Zulieferer an einen Tisch
- Bildung KBL-Arbeitsgruppe bei ProSTEP
- Implementor-Forum ECAD-IF



## Einsatzziele KBL 1 OEM und Zulieferer

- Datenaustauschformat
- Bestellformat
- Vollständig und verbindlich
- Entlastung beim Zulieferer durch Wegfall proprietärer Systeme



## Einsatzziele KBL 2

### Komplette Leitungsstrangdaten

- Elektrische und mechanische Eigenschaften von Leitungssträngen
- Teile (Stückliste u. Stammdaten)
- Verbindungen (Konnektivität)
- Geometrie (Topologie)
- Varianten (Module)



## Einsatzziele KBL 3 Übermenge Kabelbaum

- Eine KBL-Datei kann die Datensätze 1..n konkreter "Kabelbäume" enthalten
  - Einzelmodule
  - "100%-Kabelbaum": kompletter Leitungsstrang
  - "120%-Kabelbaum": überkompletter Ltg.-Str.
  - "150%-Kabelbaum": Modulsammlung



## KBL und AP212 1 STEP, AP212 und ProSTEP

- AP212: ISO-Norm auf der Basis von STEP
- "Application protocol: Electrotechnical design and installation"
- Sehr umfassend und komplex (ca. 2700 Seiten)
- Auftrag an ProSTEP: Ableitung eines spezifischen Datenmodells aus AP212 für den Bereich physische Bordnetze



## KBL und AP212 2

### Positionierung KBL • AP212

- **KBL:** Untermenge aus AP212 mit spezifischer Begriffswelt
- **AP212:** Umfassend, generell, flexibel komplex, "kompliziert"
- **KBL:** Spezifisch, begrifflich konkret  
Leichter verständlich, unflexibler
- **Merke:** KBL ist NICHT gleich AP212!

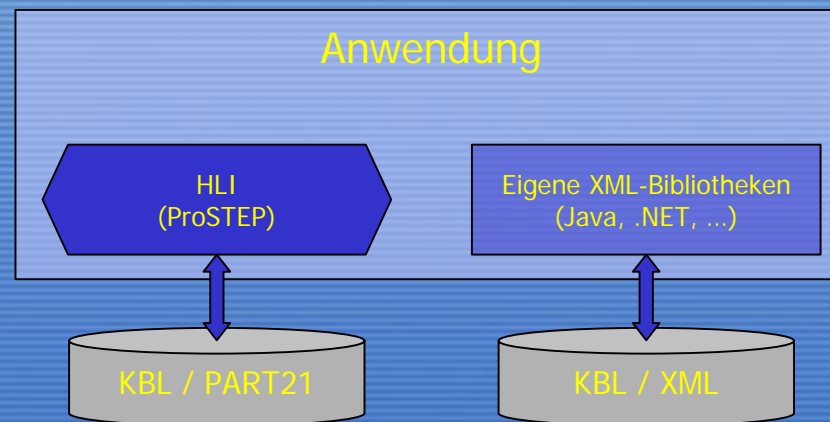


## Datenformate 1 PART21 und XML

- Ursprünglich nur STEP-Datenformat PART21 (ASCII)
- Bald Wunsch nach XML-Datenformat
- XML-Schemadefinition
- Aktuelle Version: 2.2
- Im Prinzip: "Eine KBL ist eine Datei"



## Datenformate 2 Schreib- und Lesemodule



- HLI: High Level Interface
- Programmierschnittstelle für versch. Sprachen
- ProSTEP

- XML: "menschlesbar"
- einfache Übertragung (SOAP)
- Einfache Verarbeitung (SAX oder DOM)



## Elemente 11 Zusammenbauteile - Schlussfolgerungen

- Struktur durch Verweise, nicht durch Schachtelung
- Alle Komponenten eines ZBs auf oberster Ebene
- Liste der Komponenten muss ermittelt werden
- Nur einstufige Zusammenbauten
- Mögliche Konsistenzprobleme bei Modullisten und anderen Verweisen



## Text-Attribute 1 Allgemeines

- Viele Elemente besitzen textuelle Attribute
- Werte nicht genormt (höchstens Empfehlungen)
- Kategorien:
  - Typ-Bezeichnungen
  - Schlüsselwerte
  - Farben
  - Freie Werte



## Text-Attribute 3 Hinweise und Schlussfolgerungen

- Typ-Bezeichnungen, Schlüsselwerte und Farben müssen zwischen Sender und Empfänger vereinbart werden!
- Konfigurationsausdrücke (Logistic\_control\_information) sind "freie" Texte!
- Wire\_number nicht notwendigerweise numerisch!



## Praktische Hinweise 2 Problematische Elemente

- **Doppelrolle von Connector\_occurrence:**
  - Mechanisches Teil (Steckergehäuse)
  - Elektrisches Anschlusselement (Träger der Cavities)
- **Doppelrolle von Cavity:**
  - Container für mech. Teile (Kontaktteile, Dichtungen, Blindstopfen)
  - Schnittstelle zwischen el. Verbindung (Connection) und Anschlusselement (Connector\_occurrence – s.o.)



## Praktische Hinweise 3 Konfiguration

- KBL enthält KEINE Konfigurationslogik
  - Verbindungen nicht konfigurierbar
  - Verwendete Teile nicht konfigurierbar
  - Topologie nicht konfigurierbar
- Konfigurationsausdrücke nur auf Ebene
  - Module
  - Harness\_configuration
- Logistic\_control\_information
  - Nur freier Text
  - Keine Syntax, keine Semantik



## Offene Punkte und Wünsche 1 Elemente und Eigenschaften

- **Einheiten für Längen und Gewichte**
  - Keine vorgegebenen festen Units
  - Bezeichner müssen selbst definiert werden
  - Unit\_name, Si\_unit\_name, Si\_prefix
- **(Id-)Description für Occurrences fehlen**
- **Keine maximalen Feldlängen definiert**
  - Abbildung der KBL in Datenbank problematisch
  - Eventuelle Verluste bei Datenaustausch



## Offene Punkte und Wünsche 2 Modulübergreifende Teile

- Kein (offizieller) Mechanismus für modulübergreifende Teile:
  - Teile, die in keinem Modul enthalten sind, sondern für einen Umgang extra zugesteuert werden müssen
  - Teile, die pro Modulzusammenstellung nur ein Mal verbaut werden dürfen



## Offene Punkte und Wünsche 3 Topologie

- Diskussion zum benötigten Umfang der Topologie noch nicht abgeschlossen
  - Topologiemodell für Transport "echter" 3D-Modelle (z.B. Catia) zu "mager"
  - Topologie für 2D-Layoutzeichnungen (Harness Lab Design) nicht ausreichend
- Evtl. mehrere Topologien erforderlich
- Topologie nicht konfigurierbar



## Verwendung KBL 1 Inhalt variabel

- Optionale Teilstrukturen (Topologie)
- Optional Elemente (Module)
- Optionale Attribute (Stammdaten)
- => Spektrum der KBL-Komplexität von "sehr mager" bis "sehr reichhaltig"



## Verwendung KBL 2 Transport Komplettsätze / Ergänzungen

- **Beispiel:**
  - Erste Lieferung: Topologie und Teile
  - Zweite Lieferung: Basismodule
  - Nächste Lieferung: Weitere Verbindungen
  - Nächste Lieferungen: Zusatzmodule und Kabelbaumkonfigurationen



## Verwendung KBL 3 Datenmodell für Anwendungen

- Vom reinen Datentransport zur Datenhaltung
- AP212-KBL als Basis für Datenmodelle zum Produktdatenmanagement
- Ergänzt und erweitert die Funktionalität von KBL



## Zusammenfassung

- KBL-Kenntnisse wichtig für Prozessverständnis
- KBL hat sich im praktischen Einsatz bereits bewährt
- Initialzündung für SW-Aktivitäten im KBL-Umfeld



## Ausblicke

- KBL als Basis für weitere Datenmodelle
- Evtl. noch Erweiterungen durch neue Technologien
- Vom Austauschmedium zum Datenmodell für Anwendungen
- Bidirektionaler Austausch (OEM <- Zulieferer)
- Homogenisierung der Prozesse



**Dank und Ende**

**Vielen Dank für Ihre Aufmerksamkeit!**

