



ConvConf: Einstieg mit Beispielen

Autor: Sepp Dorfschmid

Vorläufige Fassung vom 11. Februar 2026

Kontakt:

Sepp Dorfschmid Dipl. Kult.-Ing. ETH

Telefon 079 644 56 33

do@adasys.ch

Inhalt

1	Vorbemerkungen	3
1.1	ConvConf in Kürze	3
1.2	Ziel dieses Dokumentes	4
2	Konzeptionelles Modell und Test-Daten	5
3	Beispiele logischer Modelle	8
3.1	Vorbemerkung	8
3.2	LM gemäss INTERLIS 2	8
3.3	LM's mit vollständigem Inhalt aber anderer Form	9
3.3.1	GEMÄSS RELATIONALEM MODELL MIT TABELLE PRO SUBKLASSE	9
3.3.2	GEMÄSS „FLACHEM“ RELATIONALEN MODELL	10
3.3.3	KEINE REFERENZATTRIBUTE, NUR BEZIEHUNGEN	11
3.3.4	OID IN EINER SEPARATEN TABELLE ABGELEGT	12
3.3.5	STRUKTURATTRIBUTE NICHT MIT NESTING SONDERN EIGENEM DATA	14
3.4	Umgang mit Aufzählungen	15
3.5	Potenzielle Abfragen	16
3.5.1	PERSONEN MIT ZUGEORDNETER ORGANISATION (LMVEREIN_PERSORG)	16
3.5.2	ORGANISATION MIT ZUGEORDNETEN PERSONEN (LMVEREIN_ORGPERS)	17



1 Vorbemerkungen

1.1 ConvConf in Kürze

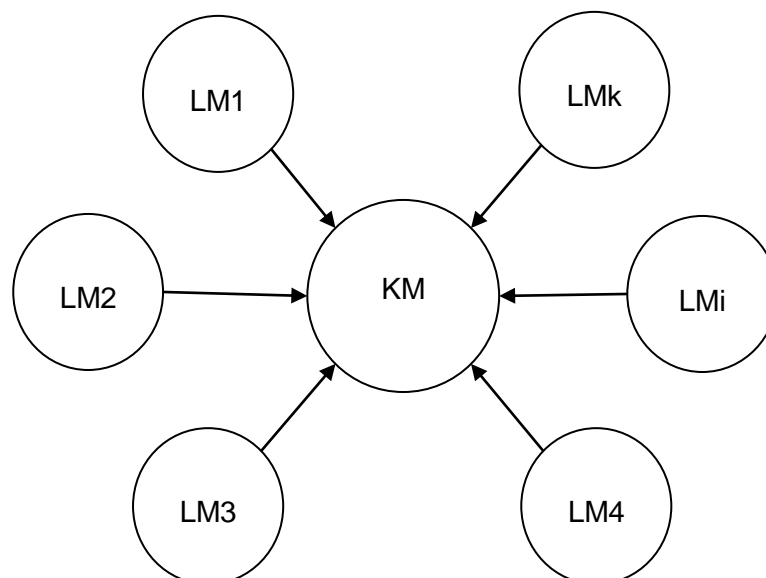
ConvConf eröffnet die Möglichkeit, dass der Aufwand für den Transfer von Daten zwischen verschiedenen Systemen (bei gemeinsamem konzeptionellen Datenmodell) minimal ist und nur für die einzelnen Systeme nicht aber für jede Sender-/Empfänger-Kombination geleistet werden muss.

Alle, die an Daten eines bestimmten Themas interessiert sind, orientieren sich am gemeinsamen konzeptionellen Datenmodell (KM) – vorzugsweise in INTERLIS2 formuliert.

Mittels der ConvConf-Sprache wird für ein bestimmtes System das individuelle logische Modell (LM) beschrieben:

- Beschreibung der Struktur der System-Daten
Arten von Datensätzen mit Attributen (Wert-Attribute und Verweise auf andere Datensätze) und eingeschachtelten Datensätzen.
- Beschreibung des Zusammenhangs mit dem KM
Mittels Views, welche auf den Klassen des KM aufsetzen, wird beschrieben, wie sich die Datensätze ergeben.

Es ergibt sich folgendes Bild:



Für einen Datentransfer von Sender-Daten gemäss LMs ins Empfänger-System, das Daten gemäss LMe erwartet, müssen sich Sender und Empfänger nicht kennen. Der Brückenschlag erfolgt mittels ConvConf.

1.2 Ziel dieses Dokumentes

Dieses Dokument zeigt an Hand einiger Beispiele, wie logische Modelle mittels der ConvConf-LM-Sprache einfach formuliert werden können. Dabei wird hier nicht detailliert auf die Möglichkeiten und Bedeutungen eingegangen. Es wird angedeutet, wie einfach verschiedene Datenstrukturierungen zum selben Dateninhalt gemäss KM beschrieben werden können.

Zur Illustration wird ein einfaches (etwas künstliches) konzeptionelles Modell verwendet.

Logische Modelle werden insbesondere für folgende Daten-Situationen formuliert:

- LM's, die drei Strukturierungsmöglichkeiten einer relationalen DB entsprechen
- Keine Referenzattribute, nur Beziehungen
- Attribute/Attributgruppen werden in einer eigenen Tabelle geführt.
- Strukturelemente in eigenen Tabellen statt eingeschachtelt.
- Vereinfachte Sichten auf das KM (typischerweise durch einen Nutzer gebraucht)

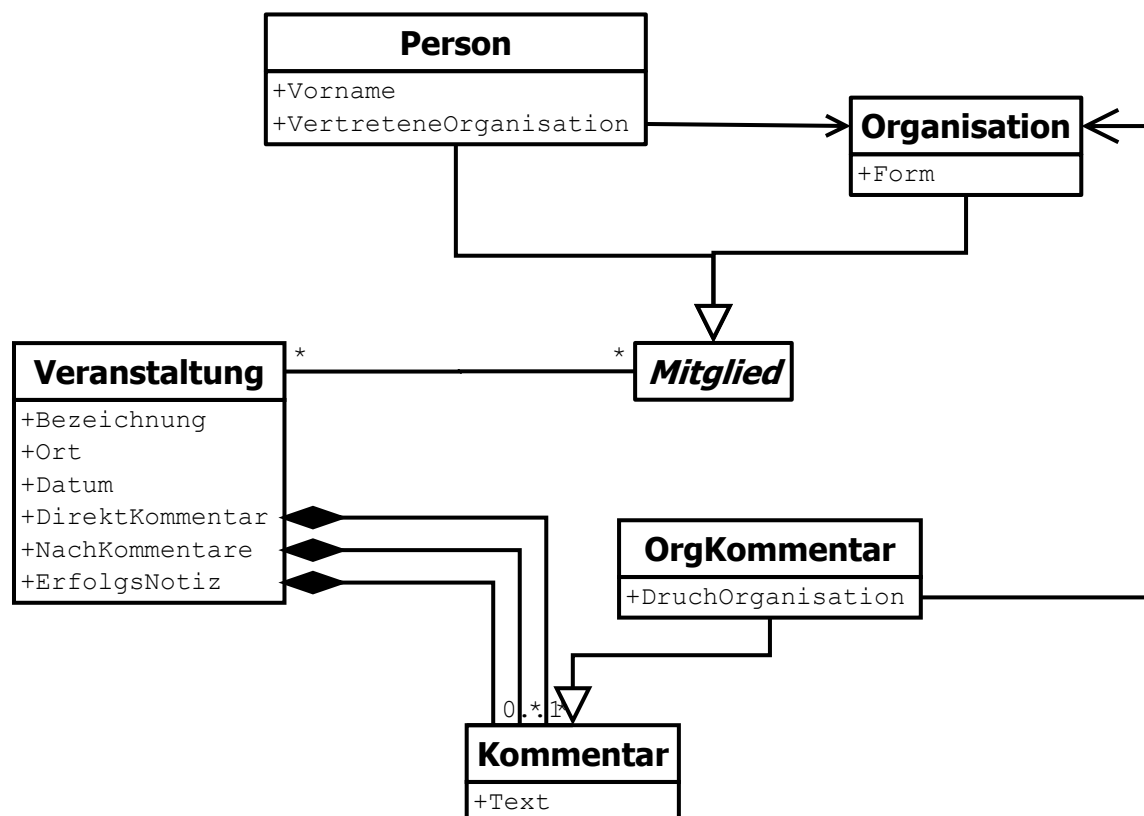
Die aufgeführten LM's (ausser den vereinfachten Sichten), enthalten alle die vollständigen Objekt-Daten (in den jeweiligen LM-Fragmenten sind nur die jeweils relevanten Teile aufgeführt).

Alle diese Konversionen wurden mit dem Prototyp-Konverter von ConfConv mit konkreten (künstlichen Daten) auch erfolgreich konvertiert. Als Beleg werden sie in diesem Dokument auch aufgeführt. Im Interesse einfacher Verständlichkeit und Verarbeitung (Simpler-Konverter) wird ein einfaches Zeilenformat (Stil INTERLIS 1) verwendet, das +/- selbst erklärend sein sollte.

2 Konzeptionelles Modell und Test-Daten

Wir stellen uns einen Verein vor:

- Er hat Mitglieder, die entweder Personen oder Organisationen sind. Personen können auch in Vertretung einer Organisation Mitglied sein.
- Er organisiert Veranstaltungen. Es wird festgehalten, welche Mitglieder daran teilgenommen haben. (Für die Tests war es interessant, wenn sowohl Personen wie Organisationen Teilnehmer sein können, obwohl es natürlich real nur Personen sein können).
- Zu Veranstaltungen können Kommentare festgehalten werden. Damit verschiedene Situationen getestet werden konnten, wurden insbesondere folgende Modellelemente definiert:
 - Die Klasse „Kommentar“ ist zu „OrgKommentar“ erweitert, wenn der Kommentar einer bestimmten Organisation zugeordnet ist.
 - In der Klasse „Veranstaltung“ werden verschiedene Arten von Kommentare unterschieden: direkt während der Veranstaltung, nach der Veranstaltung, zusammenfassende Erfolgsnotiz.





Es ergibt sich folgendes KM:

```
CM Verein:
  ENUM JurPersonArt =
    Einzel,
    EinfacheGesellschaft,
    Koerperschaft (
      HandelsGesellschaft (
        Kollektiv, Kommandit,
        AG, GmbH, KommanditAG
      ),
      Genossenschaft,
      Verein,
      Stiftung
    ),
    Amt,
    Hochschule
  END;

  STRUCTURE Kommentar =
    Text: MANDATORY TEXT;
  END;

  STRUCTURE OrgKommentar EXTENDS Kommentar =
    DurchOrganisation: REFERENCE TO Organisation;
  END;

  CLASS Veranstaltung =
    Nummer: INTEGER/OID;
    Bezeichnung: TEXT;
    Ort: MANDATORY TEXT;
    Datum : TEXT;
    DirektKommentare: BAG OF Kommentar;
    NachKommentare: BAG OF Kommentar;
    Hinweise: BAG OF TEXT;
    ErfolgsNotiz: Kommentar;
  END;
  CLASS Mitglied (ABSTRACT) =
    Nummer: INTEGER/OID;
    Name: MANDATORY TEXT;
  END;
  CLASS Organisation EXTENDS Mitglied =
    Form : JurPersonArt;
  END;
  CLASS Person EXTENDS Mitglied =
    Vorname: MANDATORY TEXT;
    Vertretene_Organisation: REFERENCE TO Organisation;
  END;
  ASSOCIATION Teilnahme =
    Veranstaltung -- Veranstaltung;
    Teilnehmer -- Mitglied;
  END;

END.
```



Dazu wurden konkrete Daten zusammengestellt (als „Verein“ wurde die GGMM-Arbeitsgruppe mit stark reduziertem Dateninhalt gewählt).

```
DATA Person $HRG Gnägi Hans_Ruedi @
DATA Person $do Dorfschmid Sepp $A
DATA Person $Jens Ingensand Jens $heig
DATA Person $Ertz Ertz Oliver $heig
DATA Organisation $A Adasys_AG Koerperschaft.HandelsGesellschaft.AG
DATA Organisation $heig HEIG-VD Hochschule
DATA Event GGMM#52 Sitzung52 online 13.10.2025
NEST DirectComment
DATA Comment c1
DATA Comment c2
DATA OrgComment c3 $heig
ENDN
NEST PostComment
DATA Comment c4
DATA OrgComment c5 $heig
ENDN
NEST Success
DATA Comment Super
ENDN
DATA Participation $HRG GGMM#52
DATA Participation $Jens GGMM#52
```

3 Beispiele logischer Modelle

3.1 Vorbemerkung

Die logischen Modelle (LM's) bestehen jeweils aus einer Beschreibung der Daten (inkl. Reihenfolge und einer Zuordnung der Daten zum KM. Externe Daten werden dabei KM-Klassen oder Sichten (Views) auf KM-Klassen zugeordnet. Die Sichten können entsprechend INTERLIS 2 formuliert werden.

Damit insbesondere sich bei dieser Zuordnung nicht oft gleiche Namen für Daten- und KM-Attribute ergeben, werden bei allen LM's englische Name verwendet. (dummerweise häufig gleich – werde ich bei Gelegenheit noch ändern)

In den Daten werden als OID lesbare Kurzbezeichnungen mit einem \$ am Anfang verwendet.

3.2 LM gemäss INTERLIS 2

Es gibt Datensätze (vgl. oben) für Veranstaltung, Person, Organisation sowie für die Beziehung zwischen Veranstaltung und Mitglied (Person oder Organisation). Zur Veranstaltung sind einige Kommentare eingeschachtelt (NEST).

Damit die Daten der verschiedenen LM's einfach vergleichbar sind, wurden auch beim LM gemäss INTERLIS 2 englische Namen verwendet (gemäss INTERLIS-Modell müssten sie deutsch sein).

Bei den beiden Datensätzen zu LM's gemäss INTERLIS 2 stimmt also die Namensgebung nicht mit den CM-Namen überein.

Diese Daten sind die Ausgangsdaten für dieses Beispiel. Alle weiteren Beispiel-Daten sind direkt und/oder indirekt aus diesen Daten mittels ConvConf-Konversion erzeugt.

3.3 LM's mit vollständigem Inhalt aber anderer Form

3.3.1 Gemäss relationalem Modell mit Tabelle pro Subklasse

Die Datensätze für Personen und Organisation enthalten nur die spezifischen Attribute. Die gemeinsamen befinden sich in Datensätzen für Mitglied. (Auf die Kommentare wurde hier verzichtet, weil sie keine zusätzlichen Hinweise abgeben).

Im LM: Beschreibung der Datensätze	und der Zuordnung zum KM
<pre> DATA Member = Number: TransferID TO \$OWNER; Name: TEXT; END; // Member DATA Organisation = ID: TransferID; OrgForm: INTEGER; END; // Organisation DATA Person = ID: TransferID; FirstName: TEXT; myOrg: TransferID TO Organisation; END; // Person </pre>	<pre> DATA Member: <-> ANNEXE (Mitglied); = Number <-> Nummer; Name <-> Name; END; DATA Person: <-> Person; = ID<-> Nummer; FirstName <-> Vorname; myOrg <-> Vertretene_Organisation; ANNEXED Member; END; DATA Organisation: CONVERSION UFormMap(OrgForm -- UForm); <-> Organisation; = ID<-> Nummer; UForm <-> Form; ANNEXED Member; END; </pre>

Hinweise zur Zuordnung zum KM:

- Person und Organisation des LM sind der entsprechenden KM-Klasse zugeordnet
- Member (des LM) wird als Zusatz (ANNEXE) zur KM-Klasse Mitglied definiert.
- Bei den Zuordnungen von Person und Organisation wird definiert, dass zusätzlich die Member-Daten genutzt werden, um das KM-Objekt vollständig zu beschreiben.
- Der Umgang mit Aufzähltypen ist in 3.4 näher beschrieben.

Daten:

```

DATA Event GGMM#52 Sitzung52 online 13.10.2025
DATA Member $A Adasys_AG
DATA Member $heig HEIG-VD
DATA Member $HRG Gnägi
DATA Member $do Dorfschmid
DATA Member $Jens Ingensand
DATA Member $Ertz Ertz

```



```
DATA Organisation $A 5
DATA Organisation $heig 200
DATA Person $HRG Hans_Ruedi @
DATA Person $do Sepp $A
DATA Person $Jens Jens $heig
DATA Person $Ertz Oliver $heig
DATA Participation GGMM#52 $HRG
DATA Participation GGMM#52 $Jens
```

3.3.2 Gemäss „flachem“ relationalen Modell

Statt Datensätzen für Person und Organisation, gibt es nur solche für Mitglied. Diese enthalten einen Hinweis, ob es sich um eine Person oder eine Organisation handelt und alle Attribute von Person und Organisation. (Auch hier wiederum ohne Kommentare)

Im LM: Beschreibung der Datensätze	und der Zuordnung zum KM
<pre>DATA FMember = Typ: INTEGER; Number: TransferID; Name: TEXT; FirstName: TEXT; myOrg: TransferID TO FMember.Number; OrgForm: INTEGER; END;</pre>	<pre>VALUEMAP MitgliedTyp = INTEGER <-> CLASS; MAPS 1 -- Person; 2 -- Organisation; END; DATA FMember: CONVERSION UFormMap(OrgForm -- UForm); CONVERSION MitgliedTyp(Typ -- myClass); <-> Mitglied ALIAS Person ~ Mitglied#Person, Organisation ~ Mitglied#Organisation; = myClass<-> \$CLASS; Number<-> Nummer; Name<-> Name; WITH Person: FirstName<-> Vorname; myOrg<-> Vertretene_Organisation; END; WITH Organisation: UForm<-> Form; END; END;</pre>

Hinweise zur Zuordnung zum KM:

- Die FMember (F für flat) sind der Klasse Mitglied zugeordnet.
- Die Zuordnung der Attribute, die nur für Person bzw. Organisation gelten, erfolgt im Rahmen von WITH, die oberhalb definierte ALIAS-Namen verwenden (Person als Mitglied der Klasse Person).
- Die konkrete Klasse (Person oder Organisation) wird extern als Zahl (1 oder 2) festgehalten.



- Die Umsetzung erfolgt wie bei Aufzählungen (vgl. 3.4.) mit einer Wert-Abbildung (VALUEMAP), welche dann bei FMember-Zuordnung verwendet wird (CONVERSOION).

Daten:

```
DATA FEvent GGMM#52 Sitzung52 online 13.10.2025
DATA FMember 1 $HRG Gnägi Hans_Ruedi @ @
DATA FMember 1 $do Dorfschmid Sepp $A @
DATA FMember 1 $Jens Ingensand Jens $heig @
DATA FMember 1 $Ertz Ertz Oliver $heig @
DATA FMember 2 $A Adasys AG @ @ 5
DATA FMember 2 $heig HEIG-VD @ @ 200
DATA FParticipation GGMM#52 $HRG
DATA FParticipation GGMM#52 $Jens
```

3.3.3 Keine Referenzattribute, nur Beziehungen

Statt einer Referenz von der Person zur Organisation wird dieser Zusammenhang mit Beziehungsdaten beschrieben.

Im LM: Beschreibung der Datensätze	und der Zuordnung zum KM
<pre>DATA Organisation = Number: TransferID; Name: TEXT; OrgForm: INTEGER; END; // Organisation DATA Person = Number: TransferID; Name: TEXT; FirstName: TEXT; END; // Person DATA PersOrg = Pers: TransferID TO Person; Org: TransferID TO Organisation; END;</pre>	<pre>DATA PersOrg: <-> PVO ~ Person/Vertretene_Organisation = Pers <-> PVO.Person; Org <-> PVO.Organisation; END;</pre>

Hinweise zur Zuordnung zum KM:

- Die PersOrg-Daten werden dem Referenzattribut Person/Vertretene_Organisation und damit der impliziten Assoziation mit den Rollen Person und Organisation zugeordnet.
- PVO wird als lokaler Name eingeführt, damit anschliessend kurze Schreibweise.

Daten:

```
DATA Event GGMM#52 Sitzung52 online 13.10.2025
DATA Organisation $A Adasys_AG 5
DATA Organisation $heig HEIG-VD 200
DATA Person $HRG Gnägi Hans_Ruedi
DATA Person $do Dorfschmid Sepp
DATA Person $Jens Ingensand Jens
DATA Person $Ertz Ertz Oliver
DATA PersOrg $do $A
DATA PersOrg $Jens $heig
DATA PersOrg $Ertz $heig
DATA Participation $HRG GGMM#52
DATA Participation $Jens GGMM#52
```

3.3.4 OID in einer separaten Tabelle abgelegt

In externen Daten werden bestimmte Attribute oder Attributgruppen häufig nicht im Rahmen der Objekte sondern in eigenständigen Daten (mit Bezug zum jeweiligen Objekt [ID TO \$OWNER]) geführt. Hier am Beispiel der OID.

Im LM: Beschreibung der Datensätze	und der Zuordnung zum KM
<pre>{ DATA Oid = ID:TransferID TO \$OWNER; Number: ObjID; END; } { (DATA Event = ID:TransferID; EventTitle: TEXT; ...</pre>	<pre>DATA Oid: <-> ANNEXE (Veranstaltung); ALIAS Object ~ Veranstaltung; <-> ANNEXE (Mitglied); ALIAS Object ~ Mitglied; = Number <-> Object.Nummer; END; DATA Event: <->Veranstaltung!; = ID<-> \$IDENT; EventTitle<-> Bezeichnung; Location<-> Ort; Date<-> Datum; ANNEXED Oid(Veranstaltung); END; DATA Member: <-> ANNEXE (Mitglied); = Name <-> Name; ANNEXED Oid; END; ...</pre>

**Hinweise zur Zuordnung zum KM:**

- Zu den Oid-Daten wird ausgesagt, dass sie sich aus Mitgliedern und Veranstaltungen ergeben. Da beiden dasselbe Attribut angesprochen wird, wird ein gemeinsamer Alias „Object“ definiert und bei der Attributzuordnung verwendet.

Daten:

```
DATA Oid 1 GGMM#52
DATA Oid 2 $A
DATA Oid 3 $heig
DATA Oid 4 $HRG
DATA Oid 5 $do
DATA Oid 6 $Jens
DATA Oid 7 $Ertz
DATA Event 1 Sitzung52 online 13.10.2025
DATA Member 2 Adasys_AG
DATA Member 3 HEIG-VD
DATA Member 4 Gnägi
DATA Member 5 Dorfschmid
DATA Member 6 Ingensand
DATA Member 7 Ertz
DATA Organisation 2 5
DATA Organisation 3 200
DATA Person 4 Hans_Ruedi @
DATA Person 5 Sepp 2
DATA Person 6 Jens 3
DATA Person 7 Oliver 3
DATA Participation 1 4
DATA Participation 1 6
```

3.3.5 Strukturattribute nicht mit Nesting sondern eigenem Data

In externen Daten werden Struktur-Daten (von Strukturattributen) häufig (insbesondere in relationalen Systemen) nicht eingeschachtelt, sondern in eigenständigen Daten geführt. Diese enthalten insbesondere den Bezug zum Objekt. Gibt es dort mehrere Strukturattribute muss (als Bezeichnung im externen System) auch festgehalten werden, zu welchem Strukturattribut die Strukturdaten gehören. Zudem wird auch hier (analog zu 3.3.2) Kommentar bzw. OrgKommentar gemeinsam („flach“) geführt.

Im LM: Beschreibung der Datensätze	und der Zuordnung zum KM
<pre> DATA Comment = StructAttr: INTEGER; Owner: DataIdent TO Event; Typ: INTEGER; Line: MANDATORY TEXT; ByOrg: DataIdent TO Organisation; END; ... </pre>	<pre> VALUEMAP CommentStructAttr = INTEGER <-> STRUCTATTR; MAPS 1 -- Veranstaltung.DirektKommentare; 2 -- Veranstaltung.NachKommentare; 3 -- Veranstaltung.ErfolgsNotiz; END; DATA Comment: CONVERSION KommentarTyp(Typ -- myClass); CONVERSION CommentStructAttr (StructAttr -- myStructAttr); <-> INSPECTION Kommentar ALL; ALIAS OrgKommentar ~ Kommentar#OrgKommentar!; = myStructAttr <-> \$STRUCTATTR; Owner <-> \$PARENT; myClass <-> \$CLASS; Line<->Text; WITH OrgKommentar: ByOrg<->DurchOrganisation; END; END; </pre>

Hinweise zur Zuordnung zum KM:

- Die Zuordnung Kennung des Strukturattributs erfolgt wiederum mit einer ValueMap.
- Die Comment-Daten ergeben sich aus einer INSPECTION (im Sinne von INTERLIS 2) zu allen Strukturattributen, welche die Struktur „Kommentar“ verwenden. (Es ist auch möglich, nur bestimmte Strukturattribute explizit zu verlangen.)
- Der Verweis auf das Oberobjekt (Datenfeld „Owner“) wird \$PARENT (und damit dem Objekt, welches das Strukturattribut enthält) zugeordnet.

Daten:

```

DATA Event GGMM#52 Sitzung52 online 13.10.2025
DATA Comment 1 GGMM#52 2 c3 $heig
DATA Comment 2 GGMM#52 2 c5 $heig
DATA Comment 1 GGMM#52 1 c1 @

```

```

DATA Comment 1 GGMM#52 1 c2 @
DATA Comment 2 GGMM#52 1 c4 @
DATA Comment 3 GGMM#52 1 Super @
DATA Organisation $A Adasys_AG Koerperschaft.HandelsGesellschaft.AG
DATA Organisation $heig HEIG-VD Hochschule
DATA Person $HRG Gnägi Hans_Ruedi @
DATA Person $do Dorfschmid Sepp $A
DATA Person $Jens Ingensand Jens $heig
DATA Person $Ertz Ertz Oliver $heig
DATA Participation $HRG GGMM#52
DATA Participation $Jens GGMM#52

```

3.4 Umgang mit Aufzählungen

Aufzählwerte sind in Systemdaten häufig mit Zahlen repräsentiert. Sie müssen also den Aufzählwerten gemäss INTERLIS zugeordnet werden.

Im LM: Beschreibung der Datensätze	und der Zuordnung zum KM
<pre> DATA Organisation = Id:TransferID; Number: ObjID; Name: TEXT; OrgForm: INTEGER; END; // Organisation </pre>	<pre> VALUEMAP UFormMap = INTEGER <-> JurPersonArt; MAPS 1 -- Einzel; 2 -- EinfacheGesellschaft; 3 -- Koerperschaft.HandelsGesellschaft.Kollektiv; 4 -- Koerperschaft.HandelsGesellschaft.Kommandit; 5 -- Koerperschaft.HandelsGesellschaft.AG; 6 -- Koerperschaft.HandelsGesellschaft.GmbH; 7 -- Koerperschaft.HandelsGesellschaft.KommanditAG; 8 -- Koerperschaft.Genossenschaft; 9 -- Koerperschaft.Verein; 10-- Koerperschaft.Stiftung; 100-- Amt; 200-- Hochschule; END; </pre>

3.5 Potenzielle Abfragen

3.5.1 Personen mit zugeordneter Organisation (LMVerein_PersOrg)

Man möchte alle Personen je mit zugeordneter Organisation (sofern vorhanden).

Im LM: Beschreibung der Datensätze	und der Zuordnung zum KM
<pre>DATA PersonWithOrg = FirstName: TEXT; Name: TEXT; OrgName: TEXT; END;</pre>	<pre>DATA PersonWithOrg : <- JOIN P ~ Person, ? O ~ Organisation WHERE P.Vertretene_Organisation==O; = FirstName<- P.Vorname; Name<- P.Name; OrgName<- O.Name; END;</pre>

Hinweise zur Zuordnung zum KM:

- LM ist nur für Output brauchbar. Darum bei Zuordnung „<-“ statt „<->“.
- Die PersOrg-Daten ergeben sich als JOIN von Person mit Organisation gemäss dem Referenzattribut „Vertretene_Organisation“.
- Es sollen auch Personen, die keiner Organisation zugeordnet sind ausgewiesen werden. Mittels dem „?“ wird darum eine Outer-Join verlangt.

Daten:

```
DATA PersonWithOrg Hans_Ruedi Gnägi @
DATA PersonWithOrg Sepp Dorfschmid Adasys_AG
DATA PersonWithOrg Jens Ingensand HEIG-VD
DATA PersonWithOrg Oliver Ertz HEIG-VD
```


3.5.2 Organisation mit zugeordneten Personen (LMVerein_OrgPers)

Man möchte alle Organisationen und darin eingeschachtelt die Personen, welche diese Organisation vertreten.

Im LM: Beschreibung der Datensätze	und der Zuordnung zum KM
<pre>DATA Org = OrgName: TEXT; Representatives: NESTING { DATA Representative = FirstName: TEXT; Name: TEXT; END; }; END;;</pre>	<pre>NESTED DATA Representative: <- P ~ Person = FirstName<- P.Vorname; Name<- P.Name; END; DATA Org : <- O ~ Organisation = OrgName<- O.Name; Representatives<- O.Person/Vertretene_Organisation; END;</pre>

Hinweise zum LM:

- In den Org-Daten werden die Repräsentanten im Rahmen eines NESTINGs definiert.
- Dazu braucht es zwei Zuordnungen:
 - Die eingeschachtelten „Representative“-Daten sind der Klasse Person zugeordnet.
 - Das Nesting „Representatives“ im Rahmen der Org-Daten ist dem Referenzattribut „Vertretene_Organisation“ der Person zugeordnet.

Daten:

```
DATA Org Adasys_AG
NEST Representatives
DATA Representative Sepp Dorfschmid
ENDN
DATA Org HEIG-VD
NEST Representatives
DATA Representative Jens Ingensand
DATA Representative Oliver Ertz
ENDN
```