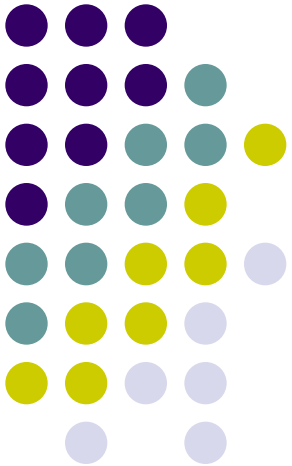
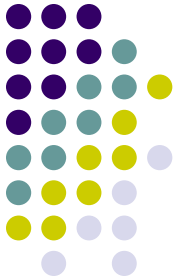
Eine statistische Modellierungsmethode zur Datenintegration

Wilfried Grossmann,
Beatrice Gurell
Fakultät für Informatik
Universität Wien

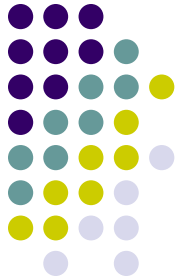


Inhalt



- Problemstellung
- Lösungsmodell
- Integrationskomponente
- Zusammenfassung

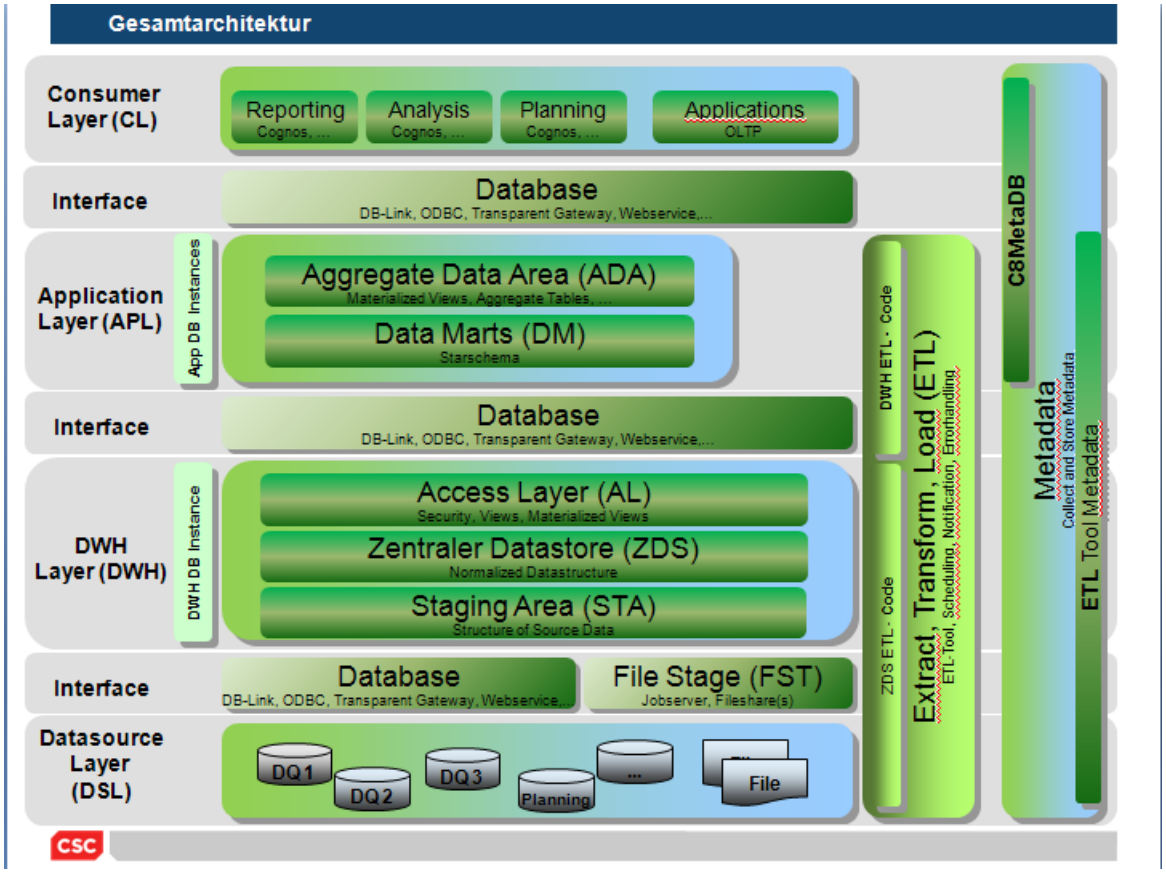
Problemstellung



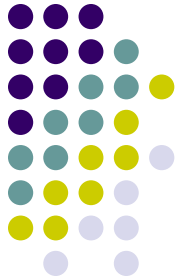
- Business Analytics setzt meist eine konsolidierte Datenbasis voraus, die alle wesentlichen Informationen enthält
- Daten werden aus verschiedenen Quellen integriert und dann als Datamarts zur Verfügung gestellt

Problemstellung

- Beispiel: ETL Architektur für Transport und Verkehr bei CSC



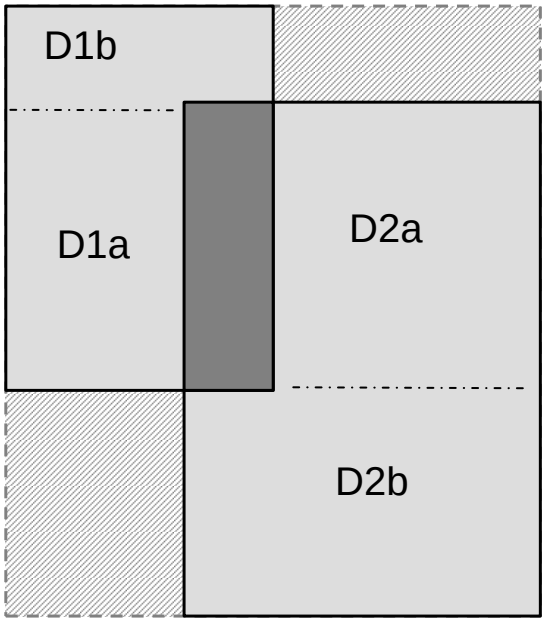
Problemstellung



- Technisch sind die Schritte für Integration zu einer normalisierten Datenstruktur meist relativ einfach
  - Abfragen, Joins, Unions,…
- Warum ist ETL dann so zeitintensiv und erfordert viel spezifisches Know How?

Problemstellung

- Probleme am Beispiel des Joins von zwei Datensätzen

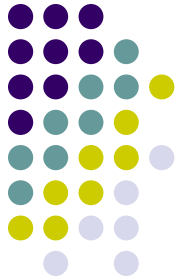


Horizontal Integration

Problemstellung

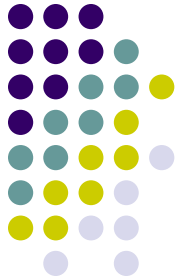
- Gibt es ein konzeptionelles Modell das
  - Fragen nicht ad-hoc löst?
  - Methodische Vielfalt zur Lösung bietet?
  - Qualitätsbeurteilung erlaubt?
- Lösungsmodell: Kombination von
  - Konzepten des (scientific) Workflow
  - Statistischer Metadaten Modellierung
- Integration in Open Models Plattform

Lösungsmodell



- Metamodellierung beruht auf vier Core-Objects:
  - (Semantics: Domainlogik)
  - Syntax: Übersetzung des Problems in eine Informationslogik
  - Mechanics&Algorithms: Prozesslogik
  - (Notation: wird nach Bedarf gewählt)

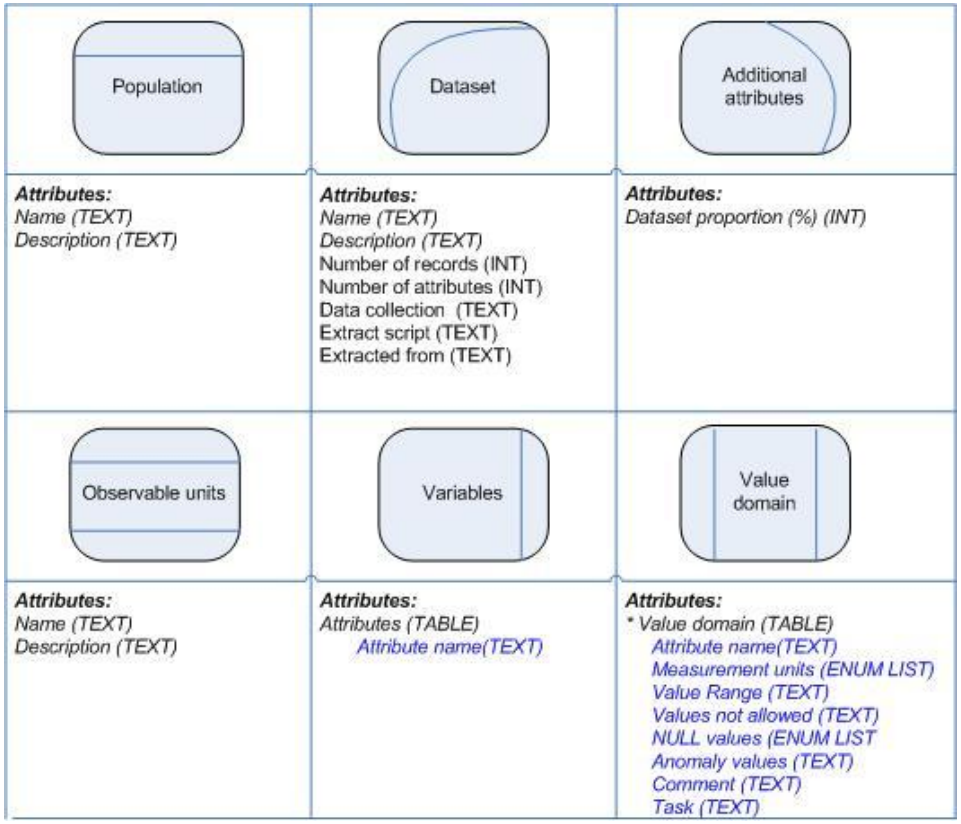
Lösungsmodell-Informationslogik



- Informationslogik für Business Analytics orientiert sich an der Informationslogik der Statistik
  - Beachte: Viele Verfahren des Data Mining sind im Ursprung statistische Methoden
- Beschreibung der Daten in einer Form, die für Fragen wie in der Problemstellung adäquat ist
  - Meta-Model für Daten

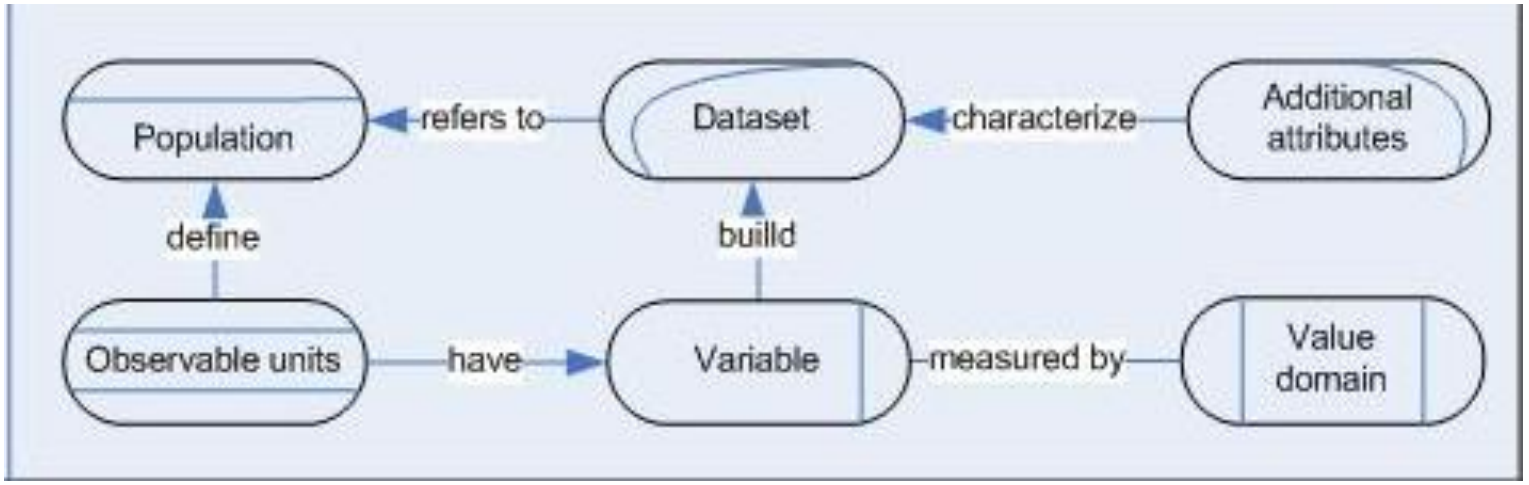
Lösungsmodell – Informationslogik

- Kategorien der Informationslogik

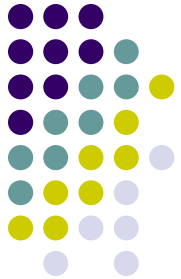


Lösungsmodell – Informationslogik

- Struktur der Informationsobjekte

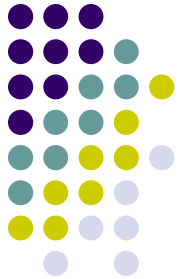


Lösungsmodell – Prozesslogik



- Prozesslogik bearbeitet die vorhandenen Informationsobjekte auf zwei Ebenen:
  - Data-Level (Algorithmen)
  - Meta-Level (Beschreibung der Daten in der Informationslogik)
- Daten und Metadaten werden also gemeinsam durch die Prozesse verändert

Lösungsmodell – Prozesslogik



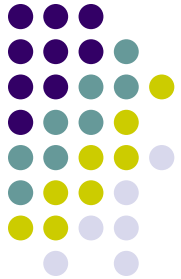
- Basisbausteine sind Transformationen der Kategorien

$$(C_1, C_2, \ldots, C_p) \rightarrow T(C_1, C_2, \ldots, C_p) = (T_1(C_1, C_2, \ldots, C_p), \ldots, T_k(C_1, C_2, \ldots, C_p))$$

mit

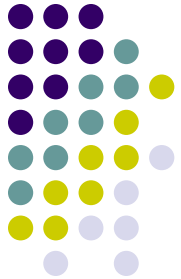
$$T_i(C_1, C_2, \ldots, C_p) = [T_i^{(D)}(C_1^{(D)}, C_2^{(D)}, \ldots, C_p^{(D)}); T_i^{(M)}(C_1^{(M)}, C_2^{(M)}, \ldots, C_p^{(M)})]$$

Lösungsmodell – Prozesslogik



- Eine Prozesskomponente besteht aus
  - einer Folge von Transformationen
  - einer Evaluationsfunktion, die das Ergebnis der Transformationen aus Sicht des Gesamtprozesses evaluiert (Datenqualität)

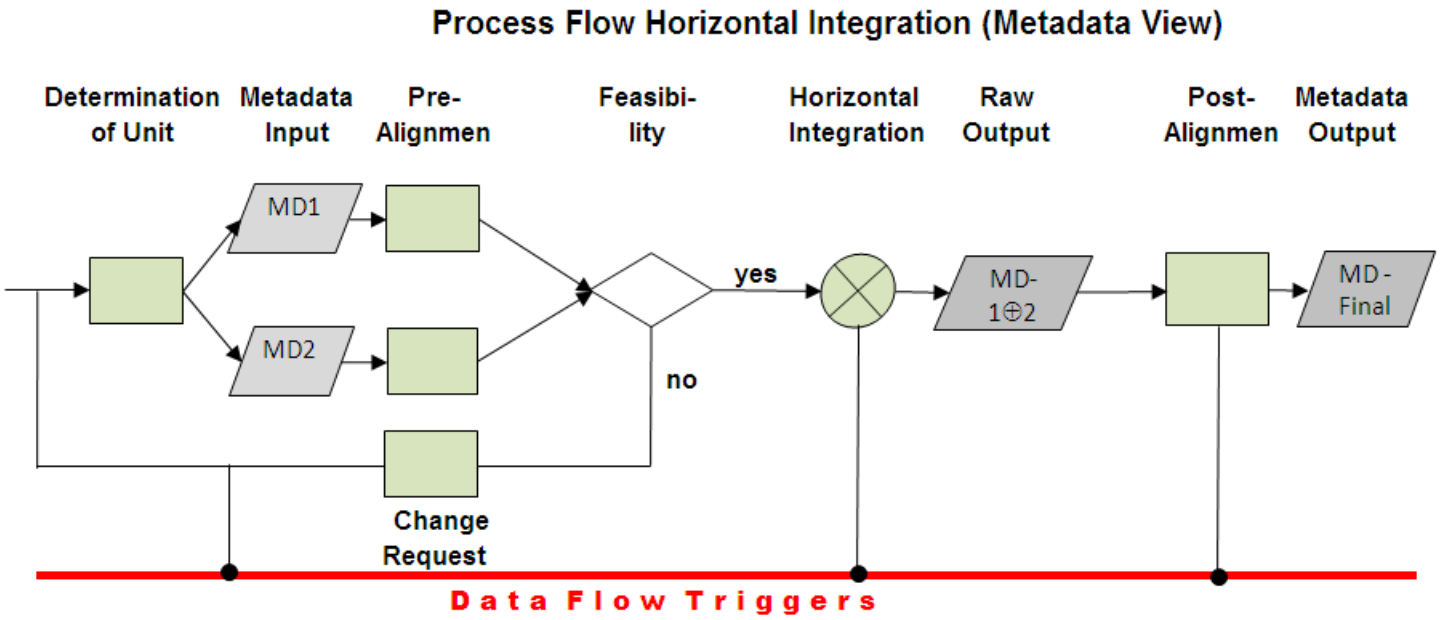
Lösungsmodell – Prozesslogik



- Generisches Format einer Prozesskomponente:
  - Task definition
  - Pre-alignment
  - Feasibility check
  - Main Transformation
  - Post-alignment
  - Evaluation

Integrationskomponente

- Allgemeiner Ablauf für Datenintegration



Integrationskomponente

- Struktur Algorithmus für Main Task

Algorithm Horizontal Integration

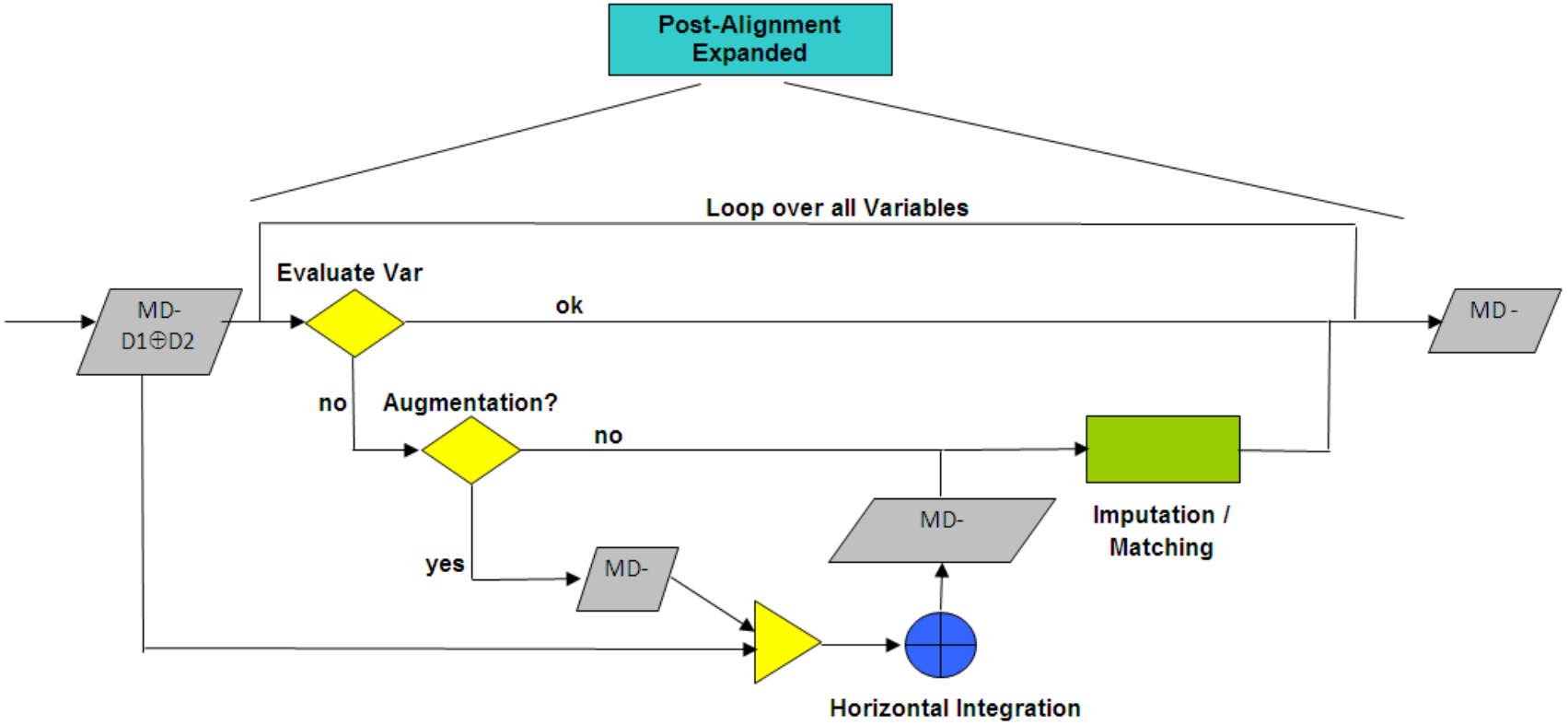
Input $D1, D2, MD1, MD2, ID$
% (Data, Meta-Information, join on observation unit ID)%

Output $D1 \oplus D2$ $MD1 \oplus 2$

begin
% Data-Level algorithm %
$D1 \oplus D2 = D1$ INNER JOIN $D2$ ON $D1.Id = D2.Id$
% Meta-Level algorithm %
$P1 = MD1\_Pop \cap MD2\_Pop$ %(joined population)%
$P2 = MD1\_Pop \setminus MD2\_Pop$ %(no join in D1)%
$P3 = MD2\_Pop \setminus MD1\_Pop$ %(no join in D2)%
$MD1 \oplus 2\_Pop = P1 \cup P2 \cup P3$
$MD1 \oplus 2\_Unit = MD1\_Unit$
$MD1 \oplus 2\_Var = MD1\_Var \cup MD2\_Var$
$MD1 \oplus 2\_ValDom = MD1\_ValDom \cup MD2\_ValDom$
$n1 = size(P1)$
$n2 = size(P2)$
$n3 = size(P3)$
end

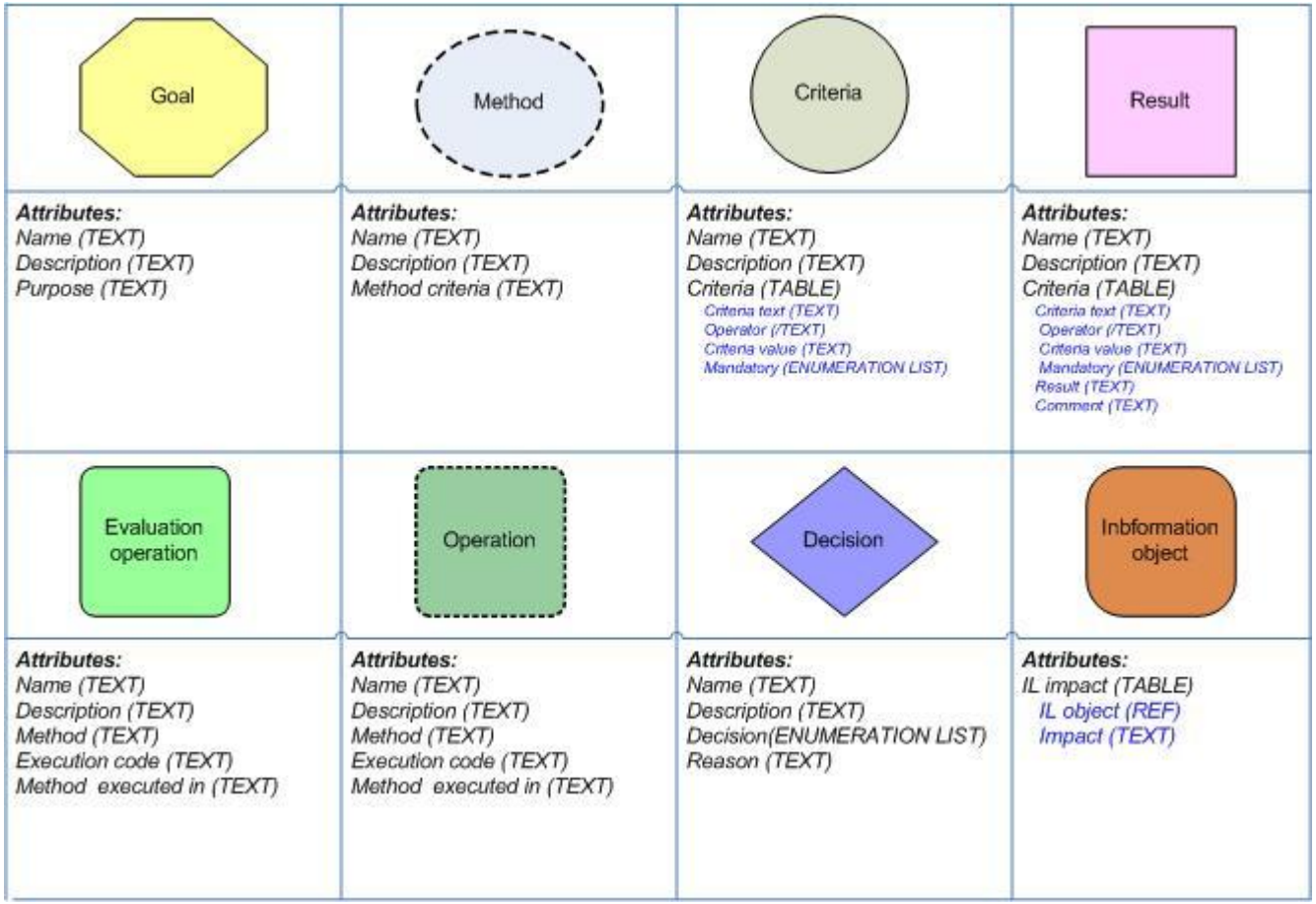
Integrationskomponente

- Zoom in Post-alignment



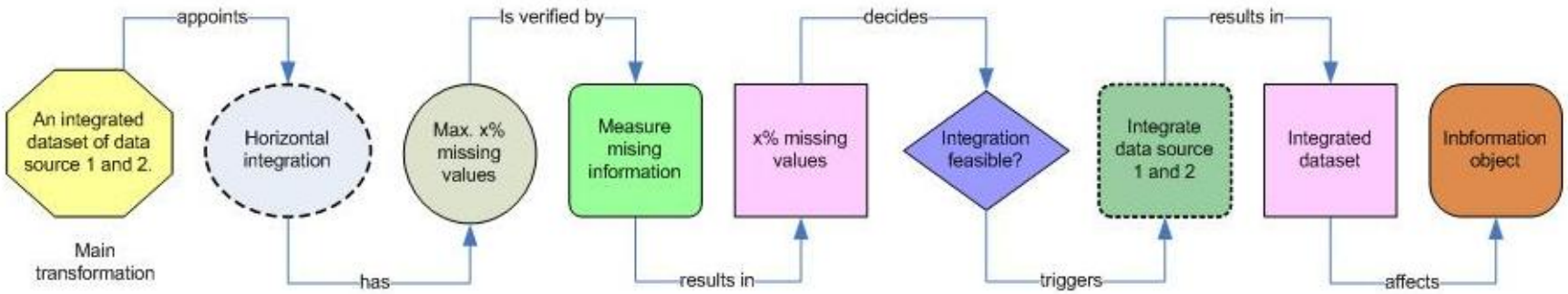
Integrationskomponente

Prozess-elemente in Open Models Architektur



Integrationskomponente

Prozessablauf Main Task



Integrationskomponente

- Datenstruktur nach Main Task

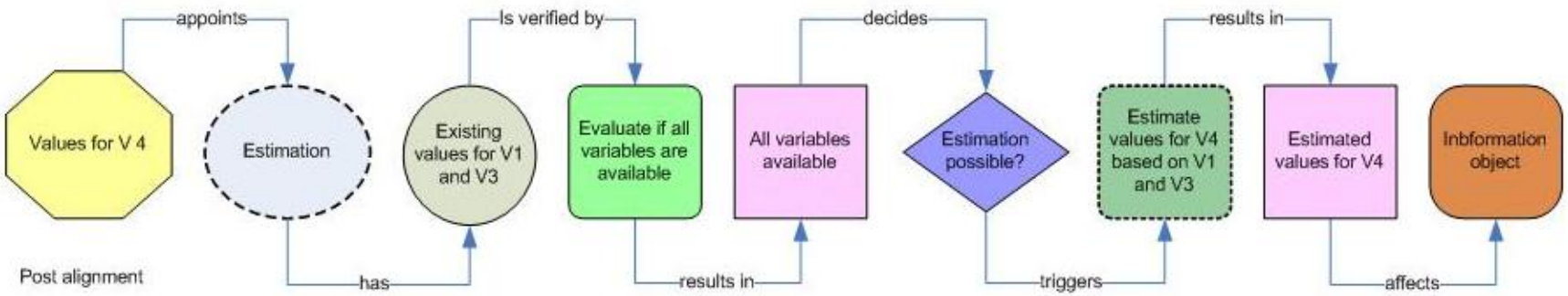
Step1: Horizontal Integration D1, D2

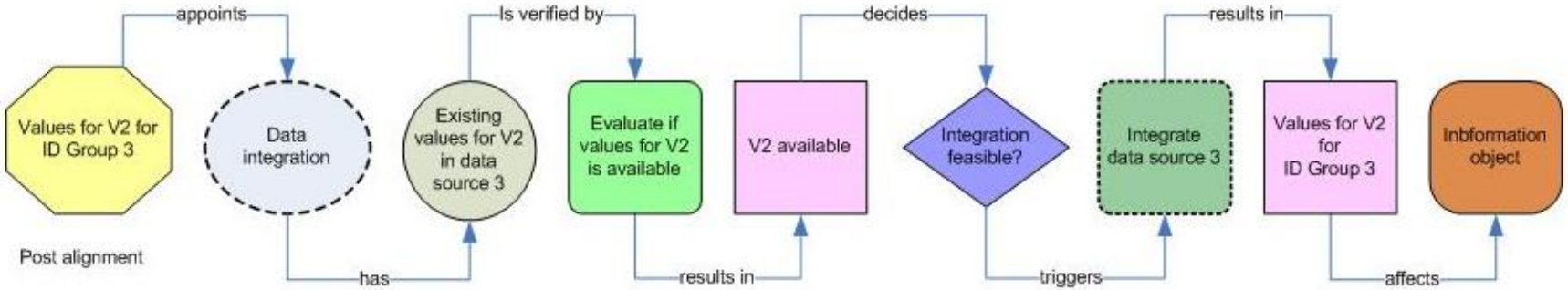
D1⊕D2

Id	V1	V2	V3	V4
G1				
G2				
G3				

Integrationskomponente

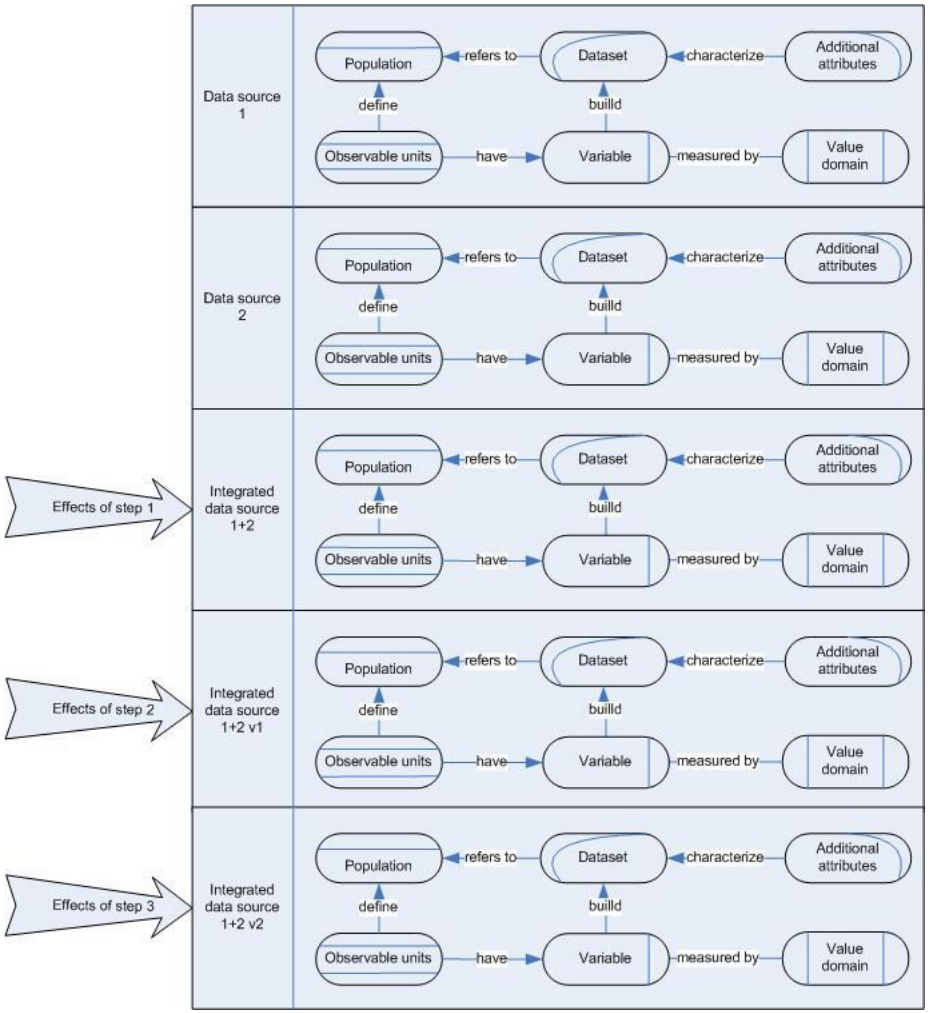
Post-alignment für V2 und V4



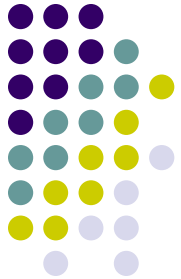


Integrationskomponente

Entwicklung auf dem Meta-Level



Zusammenfassung



- Integration erfordert Betrachtung von verschiedenen Gesichtspunkten:
  - Databanken / Data Warehouse, Statistics, (Scientific) Workflow
- Open Models als Plattform für gemeinsame Betrachtung
- Ziel: Von der grafischen Repräsentation zum exekutierbaren Code