Metadata Management for Data Integration in Medical Sciences
- Experiences from the LIFE Study -

Toralf Kirsten, Alexander Kiel,
Mathias Rühle, Jonas Wagner

LIFE Reserach Center for Civilization Diseases
University of Leipzig

BTW, Stuttgart, 08.03.2017

LIFE
Leipziger Forschungszentrum für Zivilisationserkrankungen

UNIVERSITÄT LEIPZIG

Data in Medical Sciences

- Clinical Care
  - Patients with dedicated problems in health
  - Many unstructured data, e.g., anamneses, findings, discharge reports, images
  - Structured data captured or derived from unstructured data: diagnoses, procedures etc. → goal: mostly billing
- Medical reserach projects
  - Recruited patients/probands
  - Determining a specific scientific goal
  - Mostly structured data + complex types (genetic data, images, …)

LIFE Research Center

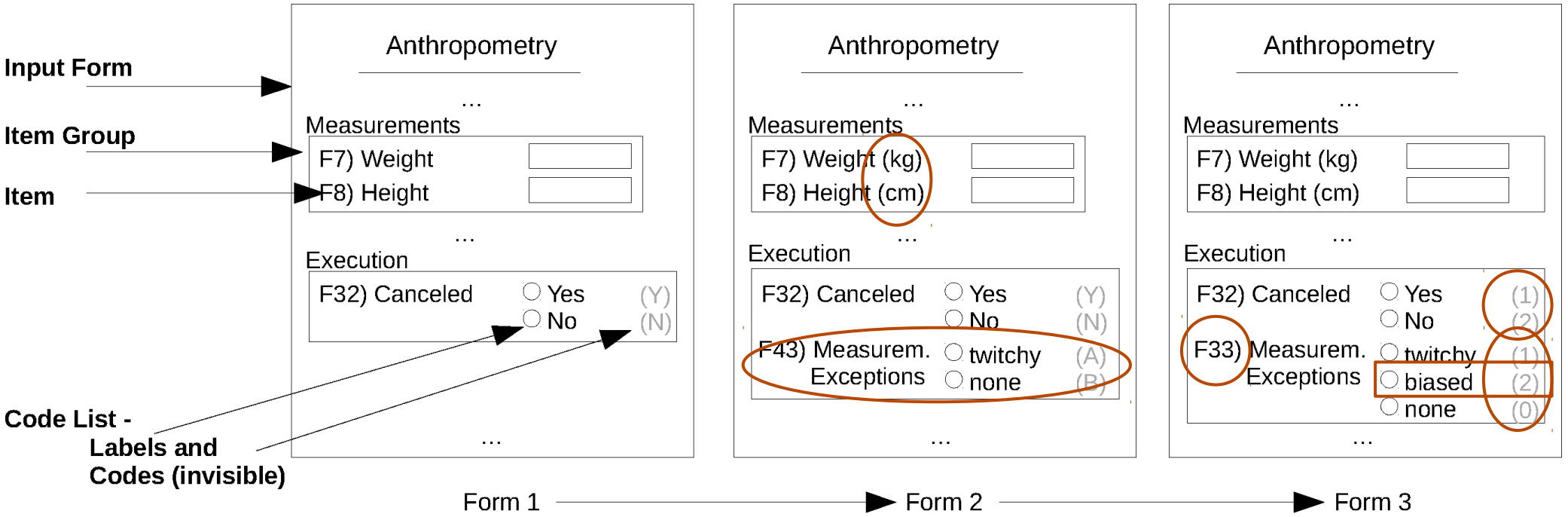
- Center at the Medical Faculty, Univ. of Leipzig
- Goal: Prevalences, risk factors and development of common civilization diseases
- Different epidemiological studies
  - Two population based cohorts (inhabitants of Leipzig)
  - Three disease specific cohorts
- Complex data capturing processes by multiple hospitals and ambulances
  - Mostly structured data capturing
  - Complex data, e.g., omics data

Multiple Input Forms (10/'16)

Assessment Type	# Assess-ments	avg(|Input Forms| / Assessment|)	|Items|	Avg(|Items| / |Assessment|)
Interview	317	3	18,980	59.9
Questionnaire	217	2	16,740	77.1
Physical Examination	78	2.5	10,606	136
Laboratory	114	1.5	2,110	18.5
. . .	. . .	...	. . .	
Total	> 850	2.4 > 1,700	> 51,000	66.7 (8 - 844)

- Evolution of input forms within a single input system
- Multiple input systems: Online ~, paper based data capturing, spreadsheets, desktop databases, ...

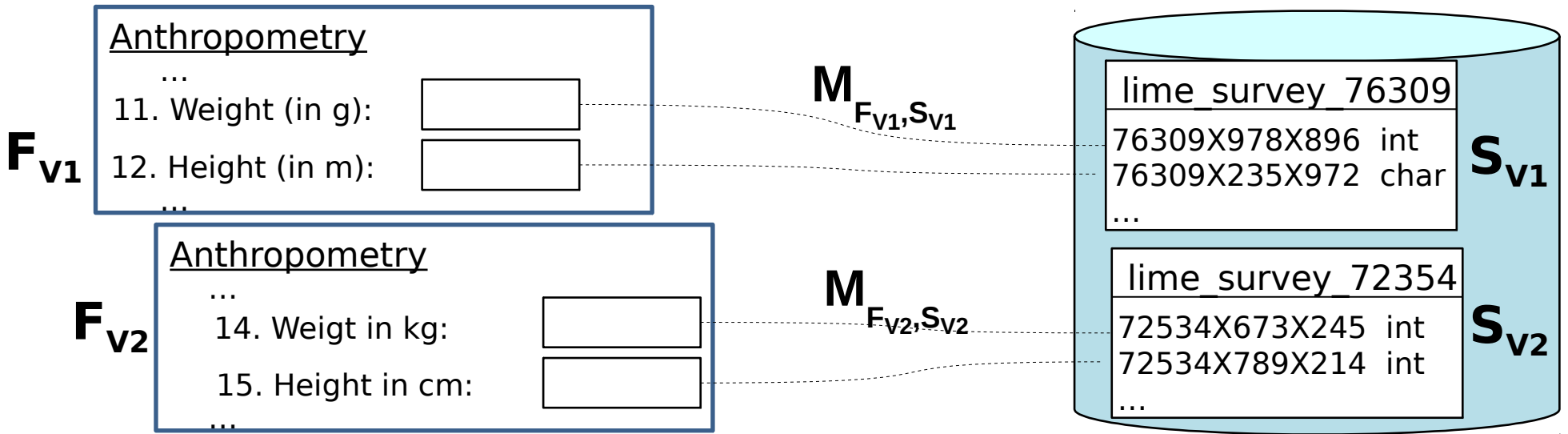
Evolution of Input Forms: Example



Änderungen

Evolution of Input Forms

- Problem: How form modifications can be managed with implications on data integration and later data analyses ?
- Two alternatives
  - Single evolving input form (per input system)
  - Multiple input forms: New form whenever a relevant modification need to be implemented

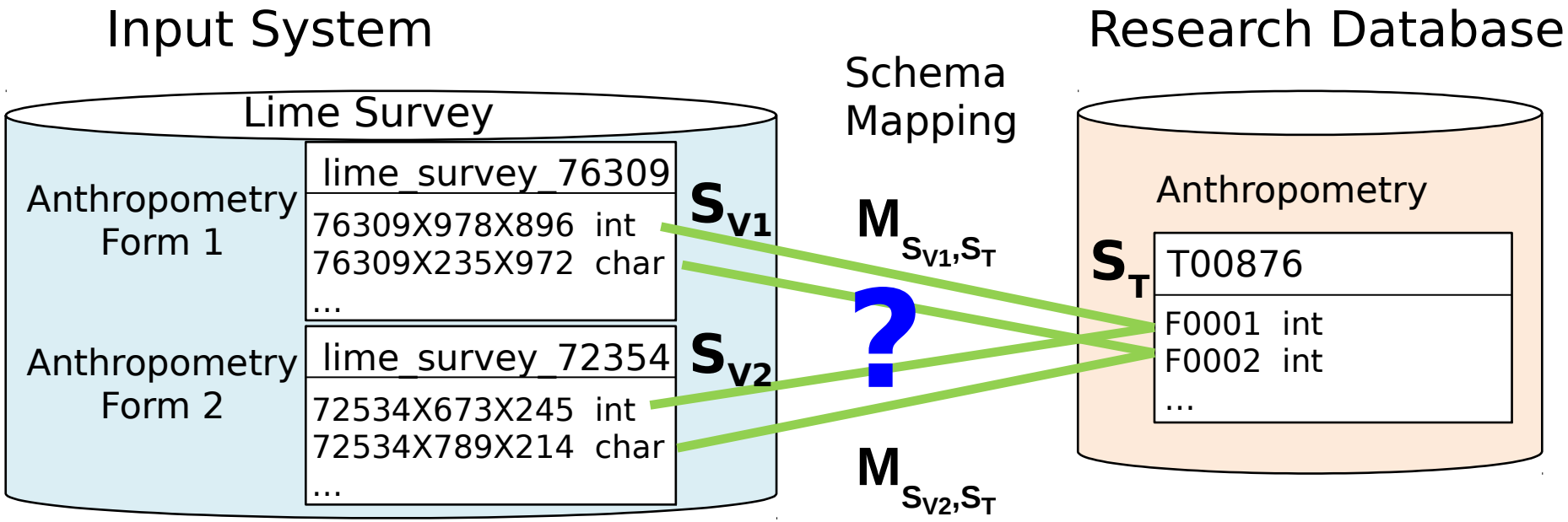


Requirements

- Data capturing and analysis in parallel
- Large set of analysis projects (> 350, Jan. 2017)
- Consider data provenance
- Harmonization of schemas according to evolution of input forms and multiple input systems
  - Study Items (questions, parameters)
  - Code lists (coding of answers)
- Efficiency
  - Automatic data transfer & transformations
  - Dynamic extension of target schema (research database)
- Further requirements: „Data descriptions“ used in analysis, Metadata for query generation, reporting, curation ...

Problem: Integration of Input Forms

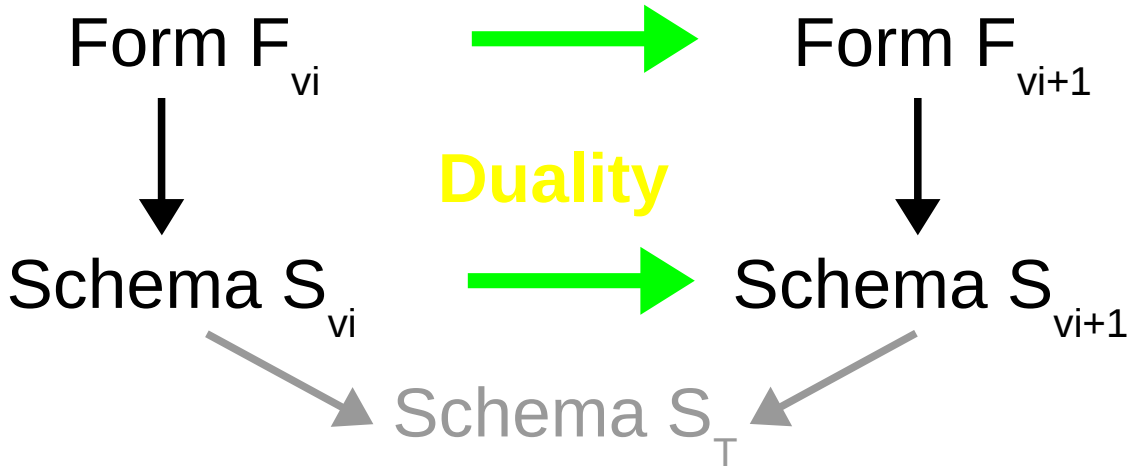
- Harmonization of study items
- Schema examples



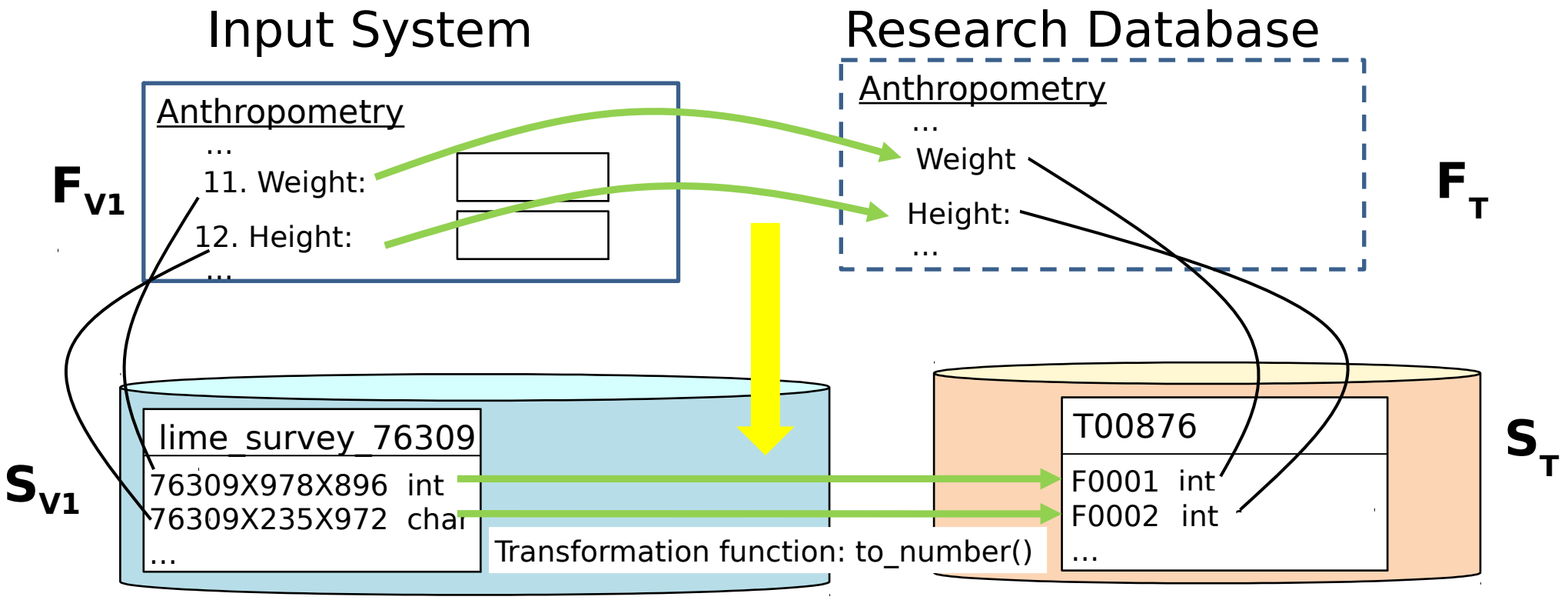
No application of matching techniques on schema level
– mostly names of schema elements are technically induced

Mapping based Approach

- Two step realization
  1) Extension of target schema T for each new assessment – first version (first input form)
  2) Mapping all further forms (vi > 1) to the succeeding form and reuse existing schema mappings M
- Central Idea: Transforming schema mapping problem into form mapping problem

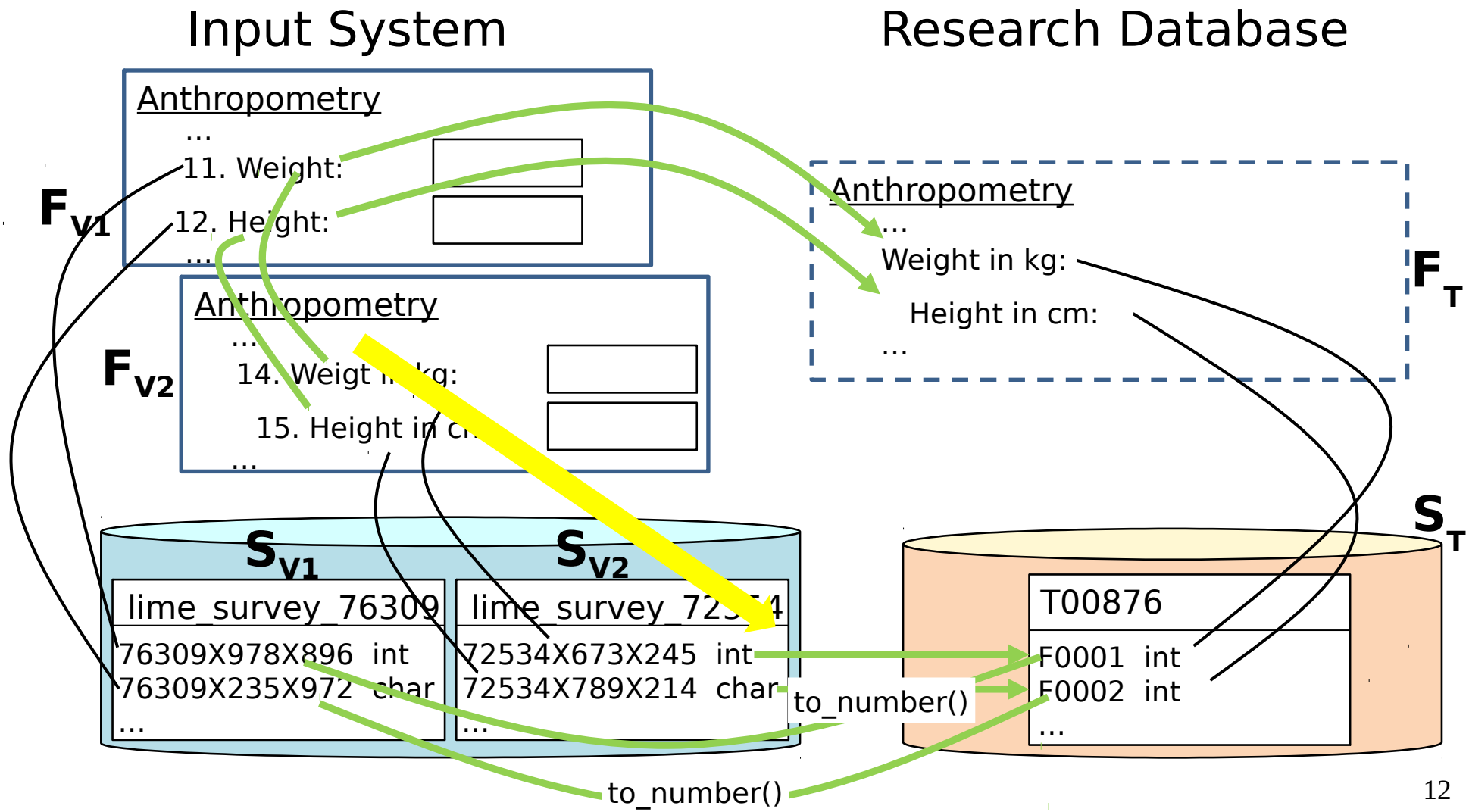


Step 1: Mapping of first Form Version



Derive schema mapping M_{S_{V1},S_T} by mapping composition

Step 2: Mapping of Form Version > 1



Form Matching

- Match process taking item description into account: Question, parameter name
- Different matcher calculating similarity between two items, e.g.,
  - String based similarity: n-gram, Levenshtein, …
  - Set based similarity: Jaccard, ...

Blocking

- Basic Idea: Reducing the number of item – item comparisons without loosing quality
- Different blocking strategies
- **In LIFE**
  - Recurring item groups, e.g., questions according to each drug (medication)
  - Item groups typically unmodified in succeeding forms
- Block → item group (block key → group name)
  - Comparing items of two dedicated blocks belonging to succeeding input forms having the same block key

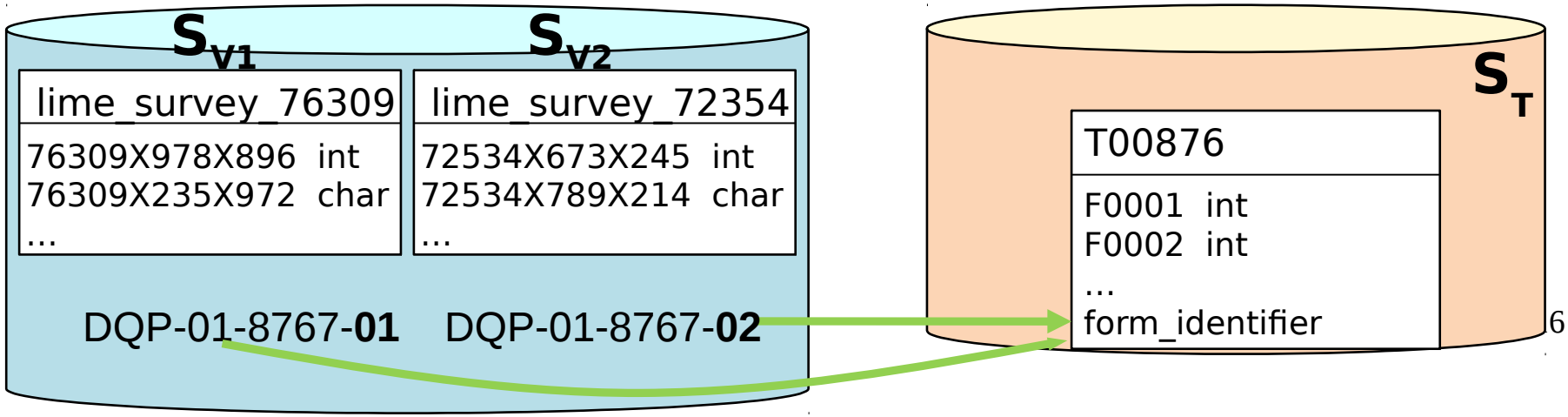
Data Type Mappings

- Mapping data types when extracting data from source system and store them into a target DB
  - Different DBMS specific data types, e.g., TEXT (MySQL), VARCHAR2, LONG (ORACLE)
  - Implementation: **type [length|precision[, scale]]** e.g., VARCHAR2 (20), INT(1), DECIMAL(5, 3)
- Building data type patterns
- Map data type patterns of sources to target DB

Source Data Type Pattern (MySQL)	Target Data Type Pattern (ORACLE)
VARCHAR(<LENGTH>)	**VARCHAR2(<LENGTH>)**
TEXT	**CLOB**

Data Provenance

- Multiple input forms per assessment
- Key question in LIFE: What data have been produced by which input system – by which input form F_x?
- Idea:
  - Associate an identifier for each form in MD
  - Represent form identifier in target table as instance



Evaluation

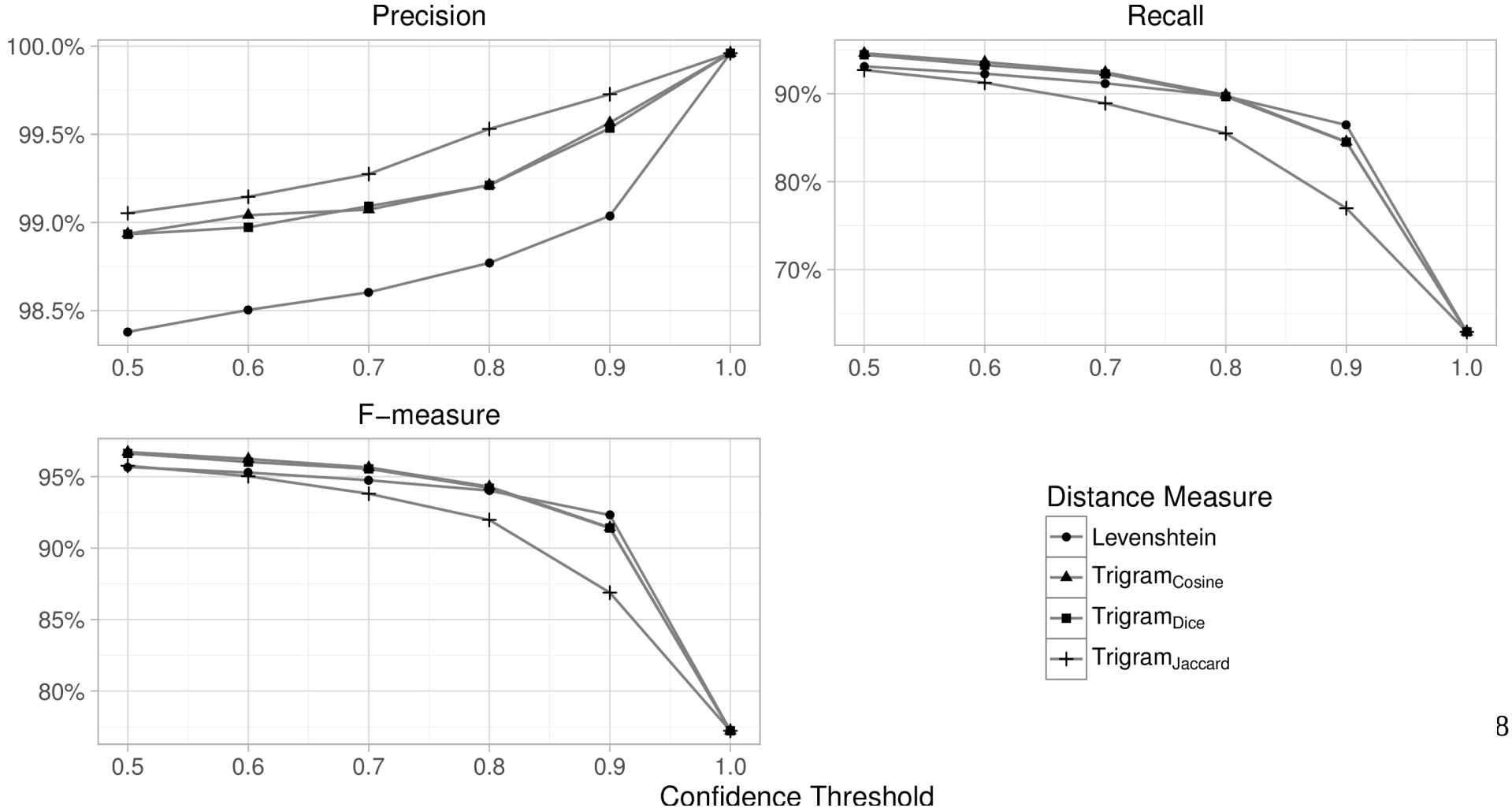
- Set up
  - Use all checked mappings as gold standard
  - Map all input forms per assessment in chronologic order
  - Evaluate match quality – no user adaptations of descriptions, aliasing etc.

|Mappings|	|Assessments|	avg (|Items per Assessment|)	min - max(|Items per Assessment|)
1	528	47.5	6 - 789
2	138	78.9	6 - 844
3	62	59.7	9 - 279
4	37	101.9	19 - 637
5	31	73.6	16 - 463
6	22	74.6	19 - 411
7	22	58.9	17 - 425
8	12	83.6	20 - 467
9	2	65.6	27 - 197
10	1	102.9	83 - 119

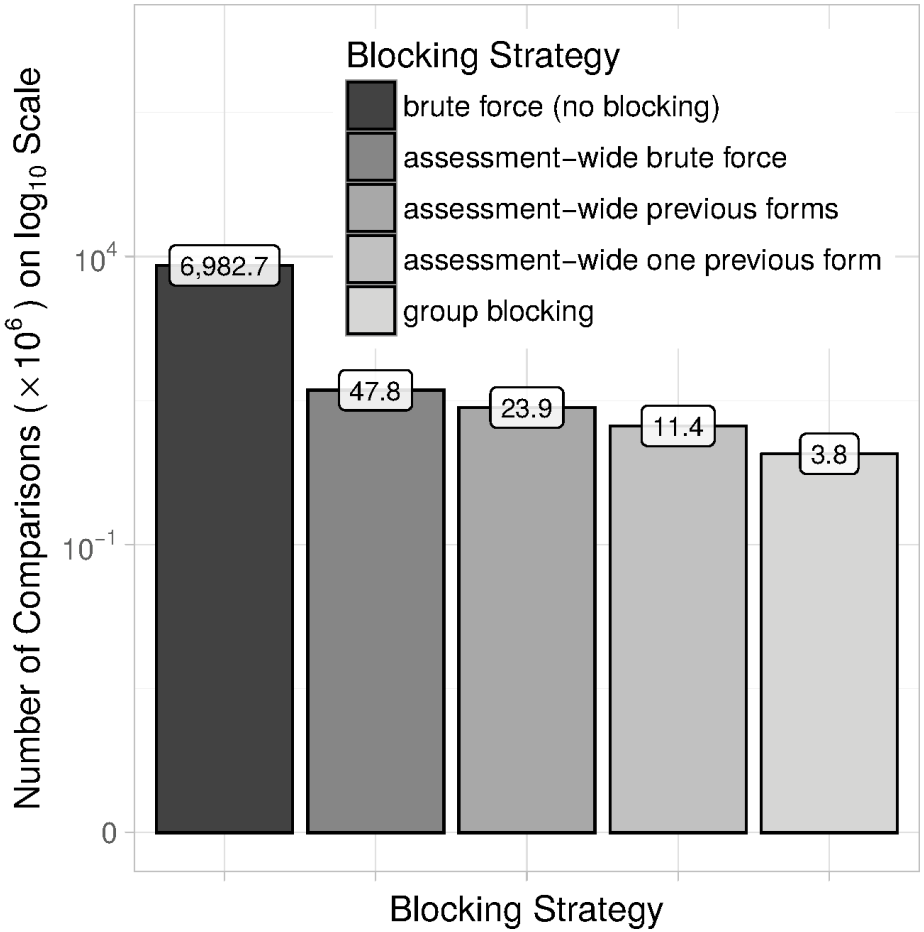
1,166 forms
327 assessments

Evaluation Results: Quality

- Trigram-Jaccard (string) with best precision but worsest recall
- Trigram-Dice with best F-Measure for nearly every threshold



Evaluation Results: Blocking



- Different blocking strategies
- Brute force = vector of all items
- Most reduction when blocking based on item groups
- Reduction factor 1,838

- No significant loss of quality when blocking mode is used

Metadata Repository

- Sometimes called data dictionary
- Central collection of
  - Sources MD
  - Assessments and input forms, code lists, data types
  - Mappings on different levels
- Used for
  - Extraction, transformation & loading
  - Query generation
  - Reporting
  - Curation (in close connection with R)

Conclusions

- LIFE: Epidemiological study with large set assessments
  - Evolving input forms (multiple forms per assessment)
  - Different input systems
- Need for harmonization
- Matching input forms → derive schema mappings
  - Automatic generation
  - Manual check & adaptation (if necessary)
- Scientific evaluation
- Running in production mode for 5y

Thank You