



Reutlingen
University

27.09. – 1.10.2020, DBKDA 2020, Lisbon

Live Data Integration




Fritz Laux
Prof. emeritus
Reutlingen University
Dept. of Informatics
Reutlingen, Germany

fritz.laux@reutlingen-university.de

© F. Laux


This talk is about virtual data integration, called “Live Data Integration”. It discusses the aspects of mediated data sharing, live (real-time) integration and data preparation for quality control.

My name is Fritz Laux. I’m a retired professor from Reutlingen University where I was responsible for the database teaching and research since the start of our department in 1984.



Reutlingen
University

My short CV



Once upon a time ...
~40 years ago

- *Education: MSc (Diplom) and PhD (Dr. rer. nat.) in Mathematics*
- *Working as SW-analyst, designer and architect of commercial information systems for ZF, Porsche, Bosch/ Junkers, Telekurs, and Swiss PTT*
- *Full Professor for Database and Information Systems at Reutlingen University. Dean of Studies. Supervised >200 Bachelor and Master students, and 3 Ph.D. students*
- *Cofounded DBTechNet (www.dbtechnet.org)*
- *Research activities in Database Modelling, Transaction Processing, Data Warehousing, and Data Mining.*
- *Research Award, IARIA fellow.*

2 /26
© F. Laux

I have studied Mathematics at the University of Tübingen and worked as SW-SW-analyst, designer and architect of commercial information systems for major companies in Germany and Switzerland.

From 1986 to 2015 I was full Professor for Database and Information Systems at Reutlingen University. With nearly 30 years in this position I supervised hundreds of students, among them many who became successful as managers, researcher and academics.

I am a cofounder of DBTechNet, a European initiative of academics and industry to improve and promote database education.

Over the years I was part of three EU-funded projects and numerous research and development projects in cooperation with industry.

My research activities are Database Modelling, Transaction Processing, Data Warehousing, and Data Analysis. The latter usually require careful data preparation and integration for meaningful analytical applications.

This is the motivation for my talk.

Outline	
Reutlingen University	↳ <i>Motivation for Live Data Integration</i>
	↳ <i>Requirements & Challenges</i>
Outline	↳ <i>Framework for Integration</i>
Motivation	
Framework	
Live Data	
Preparation	↳ <i>Live Data using Views and REST-Service</i>
Integration	
TGM	
Example	↳ <i>Data Preparation Precondition for Data Quality</i>
Transactions	
Conclusion	↳ <i>Data Integration using the Typed Graph Model illustrated by Example</i>
References	
	↳ <i>ReadCheck for Transaction support</i>


In this presentation we will motivate the need for Live Data Integration and discuss a list of requirements.

We start with a framework for Integration as a guideline for the talk.

A crucial part of integration is the selection of trustable sources and up-to-date, so called „Live“, data. This will be technically achieved with Views and REST services.

The data preparation is a necessary precondition to make data ready for integration and processing. The mapping of source data structures to a target structure is called schema mapping. It constitutes the main data integration part.

Finally we show that the ReadCheck mechanism can help with efficient virtual data integration also serves for distributed transaction support.



Reutlingen
University

Outline

Motivation

Framework

Live Data

Preparation

Integration

TGM

Example

Transactions

Conclusion


References

4 / 26
© F. Laux

Motivation for Live Data Integration

↳ *There is a need to integrate heterogeneous data sources to...*

- ↳ gain added value (knowledge, insights) for decision support, predictive analysis, performance management, etc.
- ↳ coordinate complex processes in (near) real-time with transaction support (e.g. traffic control, industry 4.0, fight epidemic)




It is rather obvious to motivate the integration of different data sources.

We expect to gain new knowledge and insights to our data that help us to make predictive analysis, coordinate complex processes and so on.

Many of the control decisions need up-to-date or near real-time information, not only in industry production but also in emergency situations like fighting epidemic, earthquake or other natural disasters.

All of them need up-to-date high quality information from different sources.

Out of current interest let us look a little closer to the analysis and fight against epidemic diseases.



Reutlingen University


- Outline
- Motivation**
- Framework
- Live Data
- Preparation
- Integration
- TGM
- Example
- Transactions
- Conclusion
- References

5 / 26
© F. Laux

Need for Live Data Integration (Example)

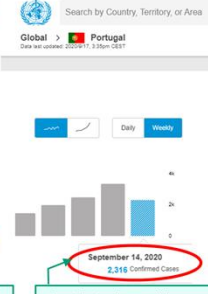
↳ Covid-19 Pandemic Analysis and Control

- ☞ Collected from 196 states under the International Health Regulations (IHR 2005)
 - ⇒ Case numbers differ due to collection methods and actuality of sources
- ☞ National authorities like RKI (Germany), CDC (USA), DGS (Portugal), ... collect the case data on county or city level, others collect on district level (e.g. SPF (France))
 - ⇒ Data need to be adjusted to the same granularity for transnational analysis
- ☞ ETL is not sufficient because of periodic updates as can be seen from the Covid-19 data below for Portugal (all from the same day 18.09.20)

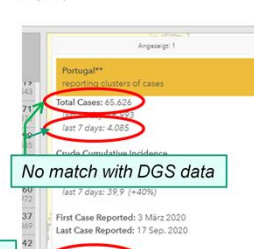


No match with WHO data

Sum of last 7 days = 4270



Outdated weekly cases



No match with DGS data

↳ Live (permanent) data integration is necessary for up-to-date information and epidemic control.


With the now present Corona Pandemic we have an actual example that needs near real-time data that is collected by different organizations and prepared by national health organization and the WHO in order to give actual information on the development and provide a fact based grounding for measures to control the disease.

If we compare the various presentations on the screenshot samples from the Web we notice that the reported data does not match despite being taken at the same time. This is due to the different collection methods and actuality of sources. In order to make the data comparable it need to be adjusted to the same granularity.

It is notable that the WHO explains that data on its dashboard “is refreshed every fifteen (15) minutes and data is accurate as at time of refreshing”, but most national authorities refresh their data only daily. In addition, Germany does not update their Covid-19 data at the weekend which results in a periodicity of the case data that is not present in reality.

Another remarkable statement from the WHO is: “We manually aggregate data from official government websites and latest new reports”. This manual process is certainly prone to errors. The situation in Germany is not better, here, the public health office (Gesundheitsamt) in some case receives data by telephone or fax and sends it the same way to the RKI.

As the problem with this method is obvious, the collection is in transition to a fully digitalized process. **Instant data integration is necessary for up-to-date information and timely epidemic control measures.**



Reutlingen University

Outline
Motivation
Framework
Live Data
Preparation
Integration
TGM
Example
Transactions
Conclusion
References

6 / 26
© F. Laux

Requirements and Challenges

- ↪ *Combine data from heterogeneous sources*
 - ☞ Challenge: transform data to be compatible for integration
- ↪ **Integrate latest data (even real-time data)**
 - ☞ Challenge: get data on the fly, increased network traffic
- ↪ *Ensure high data quality*
 - ☞ Challenge: prepare and improve data quality
- ↪ *Transaction support*
 - ☞ Challenge: distributed transactions for data management

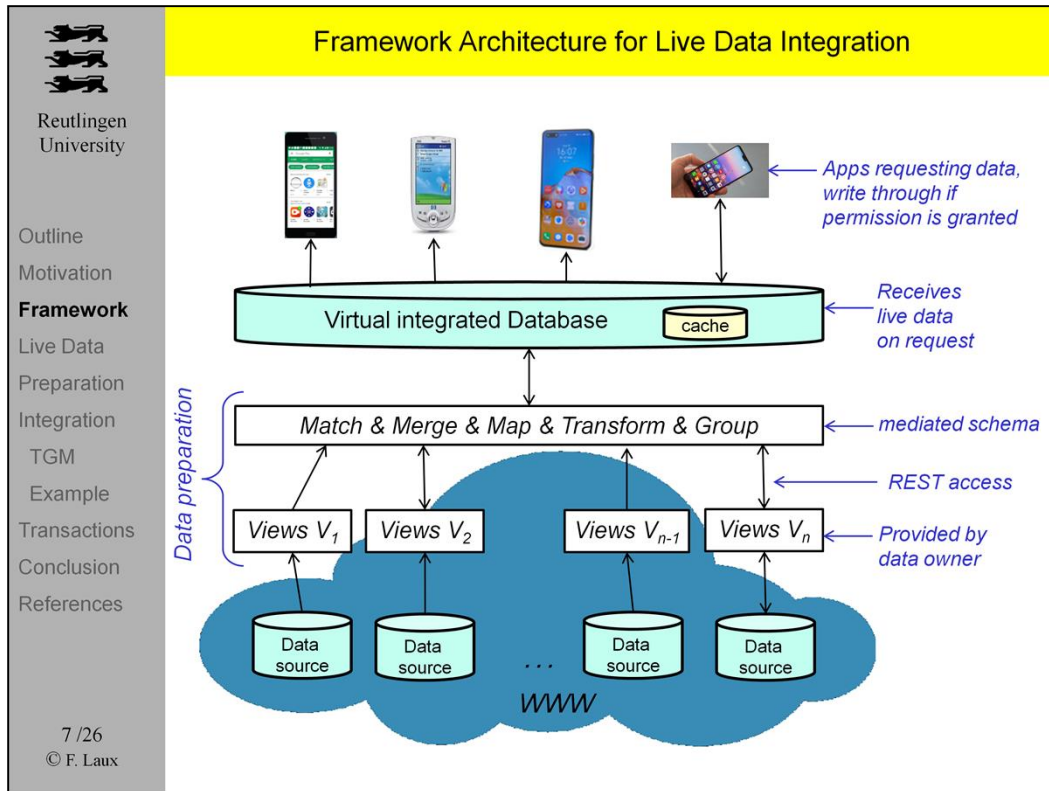
For a successful Live Data Integration the following list of requirements are essential:

Data from heterogeneous sources must be combined, this means, data has to be transformed to a compatible for integration.

The latest data (even real-time data) have to be selected and integrated on the fly. This results in increased network traffic.

We have to ensure high data quality. This makes data cleansing necessary.

Finally, distributed transaction support is desirable and necessary for data correction and amendment.



This picture is an overview for a mediated data sharing framework architecture.

It depicts from bottom to top data sources available in the WWW. This is convenient because we recommend to use mediated views of the data sources and access them via REST services.


The main effort of data preparation is placed into the mediated schema providing a homogeneous integrated view on the data. This requires matching, merging, mapping and transformations of data.

The resulting virtual integrated database does not store data but maintains a cache to reduce network traffic.

Many different Apps may use these Virtual Database for analysis and decision making.

Building such a virtual integrated database is a complex task that has to consider a variety of aspects. We will address these aspects in following slides in sequence even if the tasks are interrelated.

Data Sources for Live Data



Reutlingen
University

- Outline
- Motivation
- Framework
- Live Data**
- Preparation
- Integration
- TGM
- Example
- Transactions
- Conclusion
- References

8 / 26
© F. Laux

↪ *Heterogeneous sources with different data structures need to be integrated.*


↪ **Issues**

- ☞ Find quality sources with up-to-date (live) data suitable for the application purpose
- ☞ Disclose (hidden) semantics of data require cooperation with the data owners
 - ⇒ Data owners should provide mediated data views and semantic information for integration
- ☞ If two sources contain overlapping (redundant) data they usually do not match
 - ⇒ Different granularity, actuality, semantics
 - ⇒ Examples: Covid-19 Pandemic data vary between WHO, JHU, ECDC (European and national authorities (e.g. RKI, DGS) due to different collection and registration

Selecting trustable and up-to-date sources is most important. The resulting data quality heavily depends on the sources' data quality.

Heterogeneous sources with different data structures need to be integrated. Data semantics is available from meta data if the source is a database system or XML data. But there might be hidden semantics as well that require cooperation with the data owners.

If two sources contain overlapping (redundant) data they usually do not match. We have seen this in the Corona example. The reasons are manifold: different granularity, actuality, and semantics. For example, the income of an enterprise can be reported on different periods, before or after taxation, including or excluding "unearned income".



Reutlingen
University

Live Data [Crowe2017]

↪ *Ensure that only latest data (even real-time data) is collected*

↪ *Issues*

- ☞ get data on the fly using mediated views on the live data
 - ⇒ This needs cooperation with the source owner
- ☞ Reduce network traffic using ReadCheck [Crowe2017] validation and caching
- ☞ ReadCheck is a validator for freshness that checks if the requested data is (partially) in the cache and still up-to-date.
- ☞ ReadCheck combines ideas from Etag (Fielding and Reschke RFC 7232) and RVV [Laiho2010]

9 / 26
© F. Laux

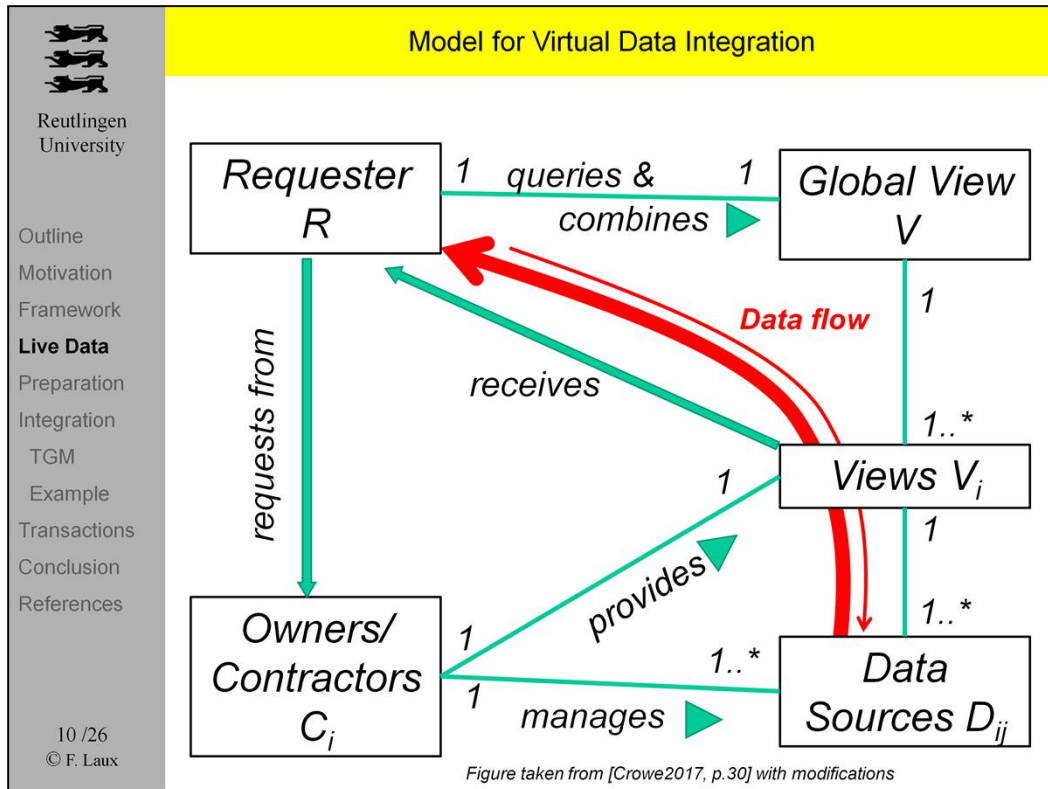
In order to avoid outdated information the idea of a virtual integration has been around for quite some time but have been rarely implemented. Crowe et al. propose in their paper to use mediated views provided by the data owners. They have demonstrated this idea with a prototypical implementation in their paper from 2017.

There is administrative need for cooperation with the source owners. The source schema should remain stable or the source admin should agree to provide stable views.

This ensures that always the latest available data is used.

But, this approach can result in extra network traffic and performance degradation. The network traffic can be reduced by using a kind of version or timestamp flag, called ReadCheck validator.

It is a validator for freshness that checks if the requested data is (partially) in the cache and still up-to-date. ReadCheck combines ideas from Fielding and Reschke (RFC 7232) on Etags for Web pages and the Row Version Validator (RVV) of Martti Laiho used in relational databases.



The conceptual model for Virtual Data Integration looks like this. It is taken from the paper of Crowe et al. and enhanced to show the data flow for queries and for update transactions.

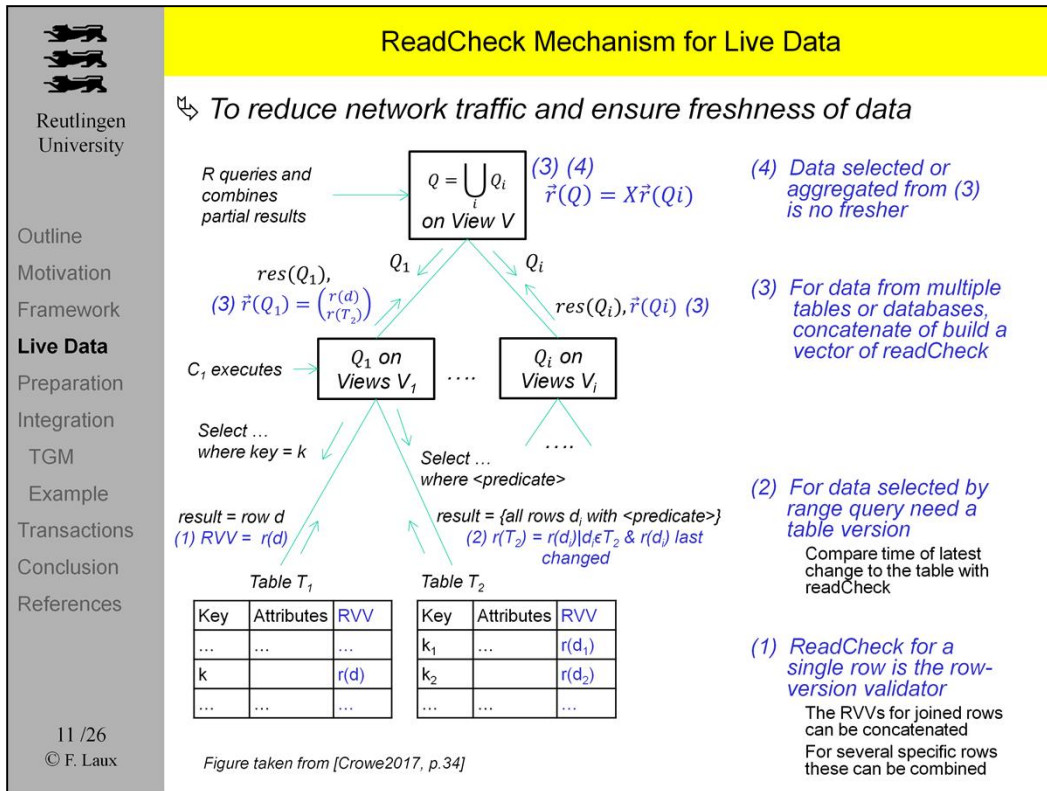
The requester R on the top left asks the data owners for data.

Each contractor C_i manages a set of data sources D_{ij} and provides a view V_i to the requestor. The view V_i is part of the contract (service level agreement) and ensures a degree of privacy.

The Views V_i are combined to an integrated global View V by the requester R . This global View is queried for analysis. No data is held by the requester except from some cached data from previous queries.

This allows to reduce network traffic if the cached data is still valid. This can be determined by the ReadCheck validator presented in the previous slide.

If permission is provided, the requester may write back transactions to correct data or to initiate actions.




The views provided on the source sites should maintain the previously mentioned readCheck validator.

On the level of tabular data we have a RVV for every row which is a monotonic increasing number like a signature for the data. When any data item changes its value, the RVV of that row increments its number.

The readCheck mechanism works in the following way:

- (1) In a query that reads a single row e.g. by key the RVV is used as readCheck value for that data. (see T1 on the bottom left)
- (2) If the query retrieves a range of rows or aggregates data the readCheck is the timestamp of the last change to the queried table. (see T2)
- (3) If data from multiple tables or databases is queried the readCheck forms a vector of each involved table and is included in the REST header as concatenated string.(see in the middle of the picture)
- (4) In case of any combination or aggregation of the previous situation the readCheck is combined from the contributing subqueries shown on the top of the picture.

 Reutlingen University Outline Motivation Framework Live Data Preparation Integration TGM Example Transactions Conclusion References 12 /26 © F. Laux	Data preparation
	<p>↪ <i>Prepare data to meet highest quality for its purpose</i></p> <p>↪ <i>Preparation steps [Sim2005] [Kemp2010] [Caf2009]</i></p> <ol style="list-style-type: none"> 1. Select data 2. Adjust measurement units 3. Harmonize semantics 4. Group and classify 5. Correct and amend data <p>☞ Steps 1 – 3 are mostly application neutral and should be realized by specific views</p> <p>☞ Steps 4 and 5 depend on the application and can be provided by the integration mapping</p> <p>↪ <i>Issues</i></p> <ul style="list-style-type: none"> ☞ Requires semantic knowledge (coding, type, granularity, etc.) for all steps ☞ Processing costs for step 4 ☞ Human decisions for step 5 required

We now turn to the data preparation task. Its goal is to achieve the best possible data quality.

The preparation process can be broken up into 5 steps, consisting of

- 1) Data Selection including the identification of relevant data
- 2) Adjust measurement units to be comparable
- 3) Harmonize semantics including the naming
- 3) Group and classify the data for aggregation
- 4) Correct and amend data for analysis

The first 3 steps are mostly application neutral. They could be realized by appropriate view definitions.


Steps 4 and 5 depend on the application and can be provided by the integration mapping

All these steps require a good understanding of the semantics of the data elements.

The grouping and classification is optional and adds some processing costs but can speed up a later analysis.

Data correction and amendment often need human decisions depending on the intended analysis methods. It is therefore mainly a manual process.

Data Preparation Example



Reutlingen
University

Outline

Motivation

Framework

Live Data

Preparation

Integration

TGM

Example

Transactions

Conclusion

References

13 / 26

© F. Laux

↳ **Step 1 (Covid-19 data from Web page)**

- ☞ Identify and retrieve data

↳ **Step 2**

- ☞ Get units and other meta information from HTML page for unit adjustments

```

<div role="row" class="tr_depth_0 " ...">
<div class="column_name td" role="cell" ...">
<div class="sc-AxjAm sc-fzqzLV egdybr">

<span>United States of America</span>
</div>
</div>
<div class="column_Cumulative_Confirmed td" ...">
...
<div class="sc-fznOgF fRrkWV 66.613.737">div>
<div data-id="bar" ...F">
.....

```

↳ **Step 3**

- ☞ Synonyms: apart from national language differences, the English names: cases, positive cases, reported cases, hospitalized, etc. could mean all the same or could mean different things.

↳ **Step 4**

- ☞ Group patients according cost factors (ABC analysis)
- ☞ For time series bin data into equidistant intervals

↳ **Step 5**

- ☞ Apparently incorrect: age < 0 or age > 130, interpolate missing data in time series.

In this example we start with data from the Web provided by the WHO.


The relevant data must first be isolated (selected) together with its meta-data if available. This includes coding and names for the data items.

In step 2 the measure units and other meta-data provided by the data owner are used to adjust all measures to the same unit.

The next step harmonizes all synonyms and homonyms providing unambiguous naming and data description (meta-data).

The 4th step depends on the intended analysis and can be considered as a analysis specific data preparation, e.g. ABC analysis or forecasts with time series.

The last step depends very much on human rationale and requires a human decision if data should be corrected or not. For time series missing values need to be complemented to make the algorithms work correctly.

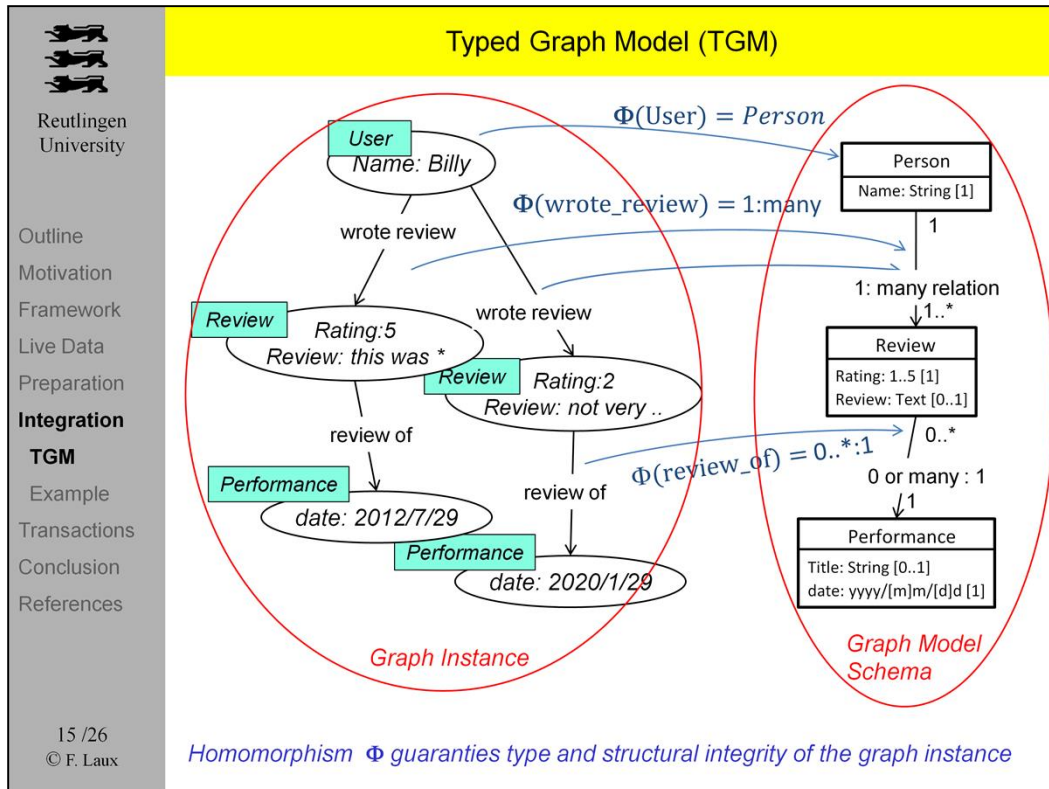
 Reutlingen University Outline Motivation Framework Live Data Preparation Integration TGM Example Transactions Conclusion References 14 /26 © F. Laux	Data Integration
	<p>↪ <i>Match and map data with compatible semantics</i></p> <p>↪ <i>Solution</i></p> <ul style="list-style-type: none"> ☞ The Typed Graph Model (TGM) [Laux2020] can help to identify, visualize, and map the data correctly ☞ TGM is flexible to support various data structures and visualizes the integration process. ☞ TGM provides clear quality criteria for the data mapping. [Laux2017] <p>↪ <i>Issues</i></p> <ul style="list-style-type: none"> ☞ The matching and mapping task is manual ☞ Choosing the best quality (freshness, reliability, precision) data is the task of the integration schema designer

Data integration aims to combine data from different sources and integrate it into a new global schema. It requires that the source data structures are matched or mapped to a target structure.

The Typed Graph Model – a Labeled Property Graph Model enhanced with data types - can help to identify, visualize, and map the data correctly. It is flexible enough to support various data structures and visualizes the integration process which helps to identify and resolve mapping problems.

In addition, the TGM provides quality criteria for data mapping.

The matching and mapping is a manual task and needs to choose best quality (freshness, reliability, precision) data. The integration schema designer has the responsibility to choose the best quality data.



We illustrate the TGM with an example taken from the paper of [Laux2020].


The pictures shows two data graphs representing an instance on the left and its corresponding graph schema on the right.

- On the left the example graph uses the property label notation as visualization showing a user named Billy with two reviews of two performances.
- The schema on the right uses UML for better visual clarity of both levels and defines the allowed structure of possible instance graphs.
- The function Φ maps the **User** to the data type **Person** which ensures that the **User** must have exactly one name. This is indicated by the number 1 in brackets. The **Review** itself is tied to the complex type **Review** with a mandatory Rating and an optional Review text. The **Performance** can have 2 properties, an optional title and a mandatory date. Even the date format is clearly prescribed by a format template.

The association cardinalities between **Person** and **Review** signify that a **Person** has at least one **Review**. The mapping Φ ensures that there are no **Users** without **Review**.

The homomorphism Φ preserves the structure between both graph levels. This means, that **wrote review** instances are tied to the 1 to many relation and therefore no second author is allowed to link to Billy's reviews.

A **Review** always refers to exactly one **Performance**, but, a **Performance** may have any number of **Reviews**, including none.



Reutlingen University


- Outline
- Motivation
- Framework
- Live Data
- Preparation
- Integration**
- TGM**
- Example
- Transactions
- Conclusion
- References

16 / 26
© F. Laux

Graph Mapping Patterns

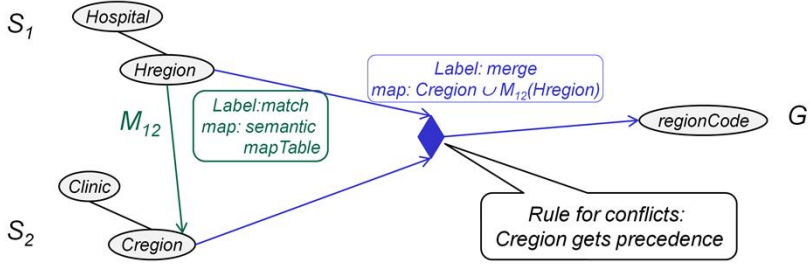
Match used in preparation steps 2 and 3

Given schema S and G. A 1:1- or renaming mapping is called a Match. The mapping preserves the semantics.



Merge used in preparation steps 1 - 4

Given mapping $M_{12}: S_1 \rightarrow S_2$, $G = S_1 \cup S_2$




In order to illustrate the modelling power and flexibility of the TGM we present a series of typical mappings that arise during schema integration and mediation.

We could call these typical situations **mapping pattern** as it reoccurs often and has a standardized solution.

The MATCH example helps with steps 2 and 3 to identify matching nodes and preserves semantics.

A match is essentially a 1:1 mapping of a data elements.

A MERGE unions two or more set of nodes which could be a necessary task in all steps 1 - 4. If two nodes are in conflict a conflict resolution is necessary.



Reutlingen University

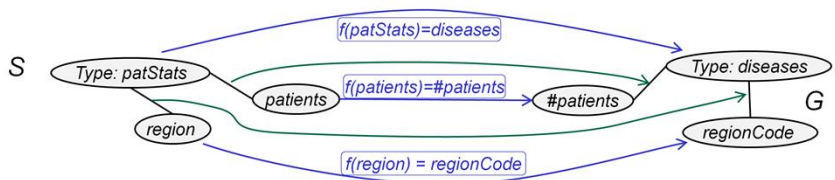
- Outline
- Motivation
- Framework
- Live Data
- Preparation
- Integration**
- TGM**
- Example
- Transactions
- Conclusion
- References

17 / 26
© F. Laux

Important Graph Mapping Types

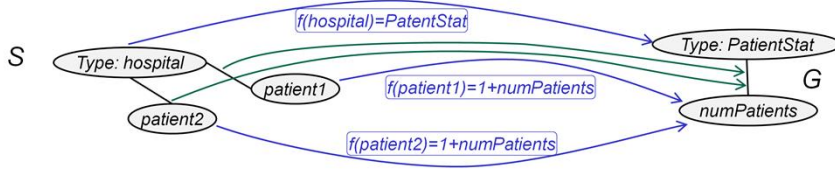
↳ **Isomorphism (Edge preserving injection) used for steps 2 and 3**

☞ Given two graphs $S=(V_1, E_1)$ and $G=(V_2, E_2)$
 $f: (V_1) \rightarrow (V_2)$ is injection and
 $\forall (v_1, v_2) \in E_1 \iff (f(v_1), f(v_2)) \in E_2$



↳ **Homomorphism (Edge preserving map) used for step 4**

☞ $f: (V_1) \rightarrow (V_2)$ is mapping and
 $\forall (v_1, v_2) \in E_1 \implies (f(v_1), f(v_2)) \in E_2$




A graph isomorphism is a edge preserving bijection i.e. a 1:1 mapping.

It helps to transform the source models into a target model and preserve structure and semantics.

This means not only matching nodes are mapped but matching edges as well.

A graph homomorphism is similar, it preserves edges, but allows that multiple nodes are mapped to the same target node. This can be used for data aggregation.



Reutlingen University

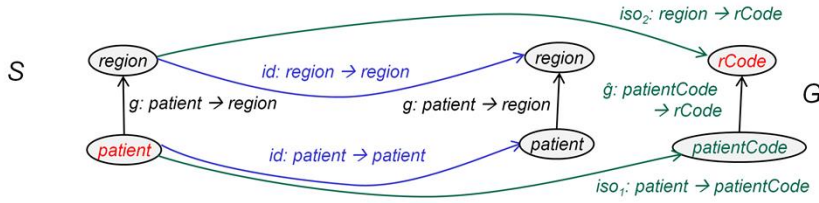
- Outline
- Motivation
- Framework
- Live Data
- Preparation
- Integration**
- TGM**
- Example
- Transactions
- Conclusion
- References

18 / 26
© F. Laux

Commutative Mappings

↪ **Commutative Mapping** used for quality control

- ☞ A function chain is called **commutative** if and only if $f_2 \circ f_1 = f_1 \circ f_2$, i.e. $f_2(f_1(x)) = f_1(f_2(x)) \forall x \in \text{dom}(f_1)$
- ☞ Example: $g \circ \text{id} = \text{id} \circ g$ (and more general $\hat{g} \circ \text{iso}_1 = \text{iso}_2 \circ g$)



- ☞ For a consistent mapping from **patient** to **rCode** it is irrelevant if the projection g to **region** is done first or the isomorphic mapping iso_1 to **patientCode**.

↪ **Desirable Mappings**

- ☞ Projection π , Homomorphism hom , and Isomorphism iso are good candidates for commutative mappings. (e.g. $\pi \circ \text{iso} = \text{iso} \circ \pi$)


When defining the mappings between two schemata special care has to be taken if a target node can be reached via different paths.

This happens for example when in the source schema two data items are related and both items are mapped to the same target node. In this case we can use the Merge pattern to resolve the conflict.

But if we preserve edges like in our examples we should have mappings that commute. If special mappings are use like projection and isomorphism then chances are good that we have commutativity.

Commutative mappings are an essential criterion for a consistent mapping.

Commutativity is important because it guaranties that we have the choice between alternative mapping paths and still end up with the same target data.



Reutlingen University

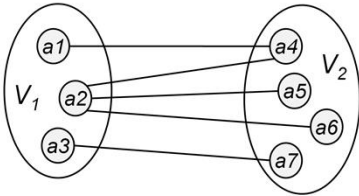
- Outline
- Motivation
- Framework
- Live Data
- Preparation
- Integration**
- TGM**
- Example
- Transactions
- Conclusion
- References

19 / 26
© F. Laux

Quality Criteria for TGM Schema Mapping (1/2)

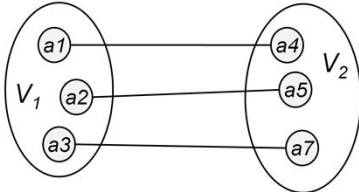
Bipartite Graph

- ☞ Let $G = (V, E)$ with $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$. If there are no edges within V_1 and V_2 then G is bipartite.
- ☞ Example 1:



Graph Matching quality criteria

- ☞ Let G be a bipartite Graph. A matching is a subset of edges where no two edges share an endpoint (node)
- ☞ Maximum matching = maximum number of vertices are matched
- ☞ Perfect matching = all vertices are matched (not merged)
- ☞ Example 2:



A bipartite graph is a graph where the nodes are separated in two disjoint sets with no edges within the subsets.


This is the case with graph matching and mapping if we remove the connections inside the local schemata.

If we consider such a bipartite graph we can define the following quality criteria:

A mapping between the source and target graphs is called maximum mapping if the number of mapped vertices is maximized.

If all nodes are matched we call this a perfect matching. This guaranties that all nodes (data elements) from the sources are matched and gives us a criteria of how well the mapping covers the integration task.

The following theorem in the next slide gives us a tool to decide if and when such a perfect match exists.



Reutlingen University

- Outline
- Motivation
- Framework
- Live Data
- Preparation
- Integration**
- TGM**
- Example
- Transactions
- Conclusion
- References

20 / 26
© F. Laux

Quality Criteria for TGM Schema Mapping (2/2)

↳ *Theorem of Hall (Marriage Theorem)*

☞ Let $G = (V_1 \cup V_2, E)$ be a bipartite Graph. In G exist a perfect matching if

$\forall U_1 \subseteq V_1: d(U_1) \geq |U_1|$.

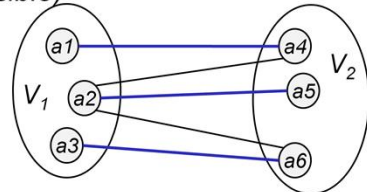
$d(U_1) := |\{v \in V_2 \mid u \in U_1 \wedge (u,v) \in E\}|$

general criteria for data integration coverage/completeness

↳ *Example 3 (perfect match possible)*

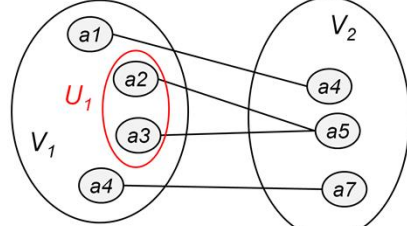
☞ All subsets U_1 of V_1 have $d(U_1) \geq |U_1|$

☞ $(a_1, a_4), (a_2, a_5), (a_3, a_6)$ is a (the only) possible perfect matching



↳ *Example 4 (no perfect match possible)*

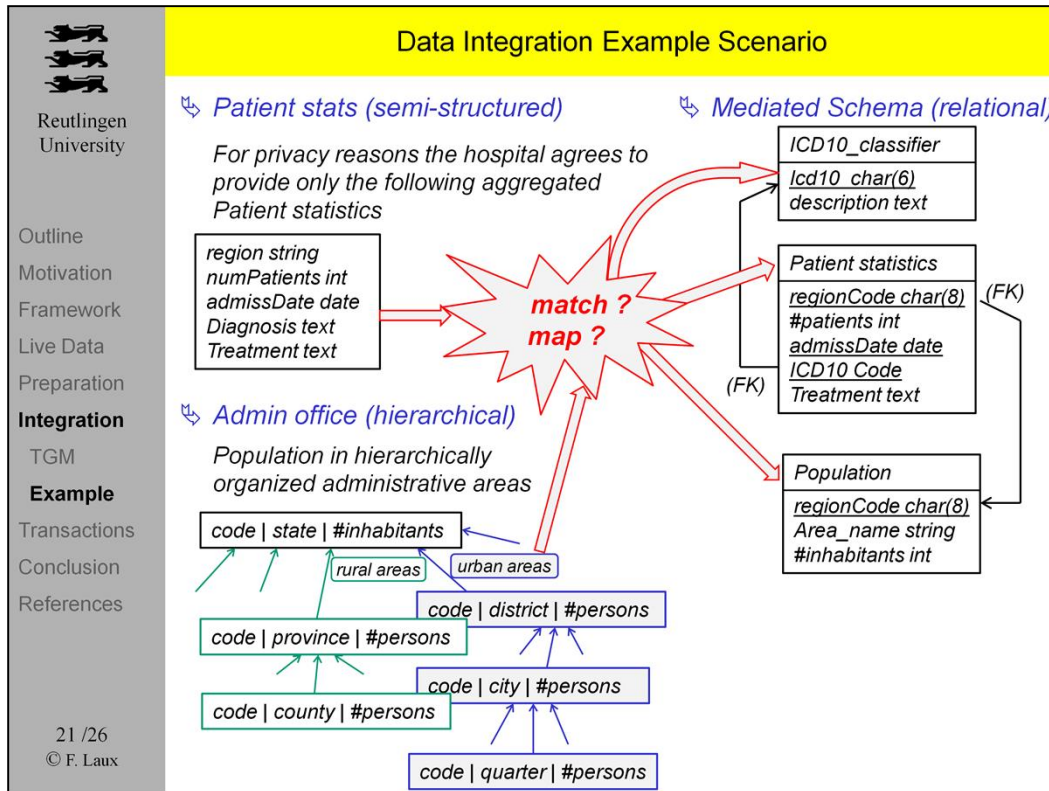
☞ Subset $U_1 = \{a_2, a_3\}$ has $d(U_1) = |\{a_5\}| = 1$, but $|U_1| = 2$.



The theorem of Hall states that if all subsets of the source nodes have more links to target nodes than the cardinality of the subset then there exists a perfect matching.

The perfect matching is a general criteria for the coverage or completeness of a data integration

If no perfect match exist a merge conflict can arise and a conflict resolution is necessary.



Let us take the pandemic corona situation again and illustrate our approach with an example scenario that we are going to solve now.

The WHO is monitoring the world health situation and reports on emergencies/epidemics. The data is provided in many different formats by national or regional autonomous actors like national offices or hospitals.

In this slide we use different visual rendering for the different data structures. The target data will be a relational schema.

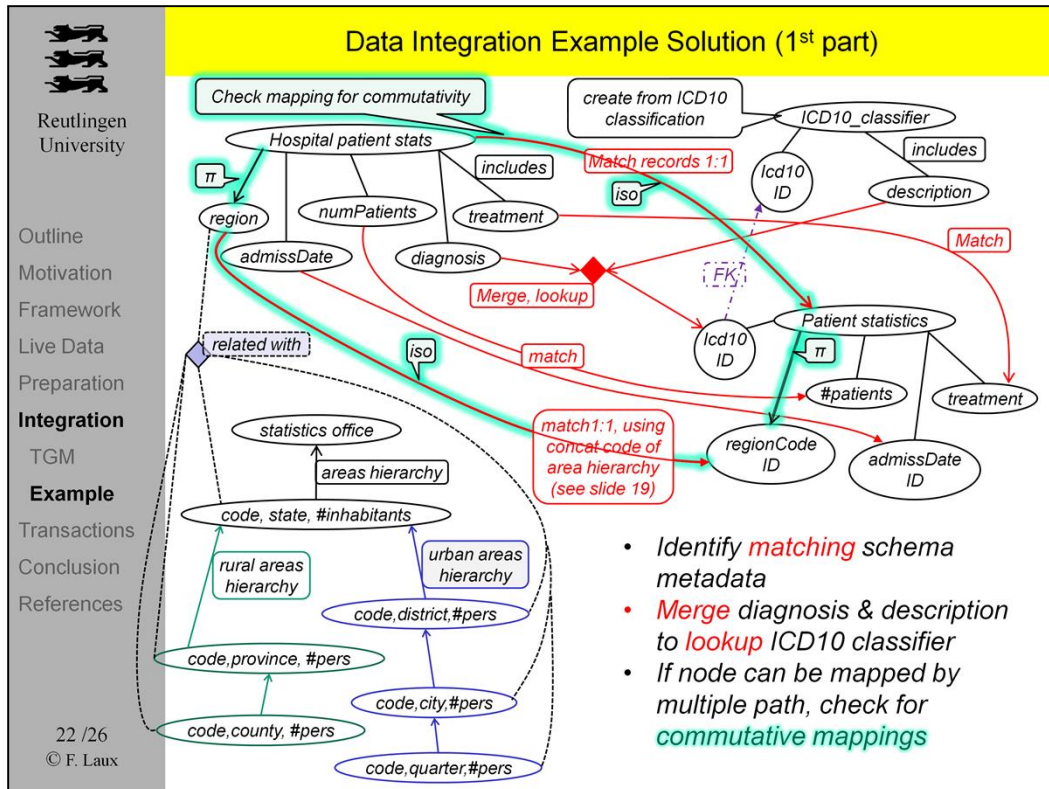
For simplicity we only show one hospital providing aggregated tabular data on patients data grouped by regions, admission date and diagnosis.

The statistics admin offices deliver demographic data in an hierarchical structure reflecting the administrative areas.

We want to create an integrated/mediated schema that combines both information using the international classifier for diseases (ICD).

Given the Integration Schema, the question is how do we find matches for the data items and how to transform them. The naming, coding, and semantics of the data is different for different sources.

Nevertheless, the WHO wants to report consolidated and comparable data.



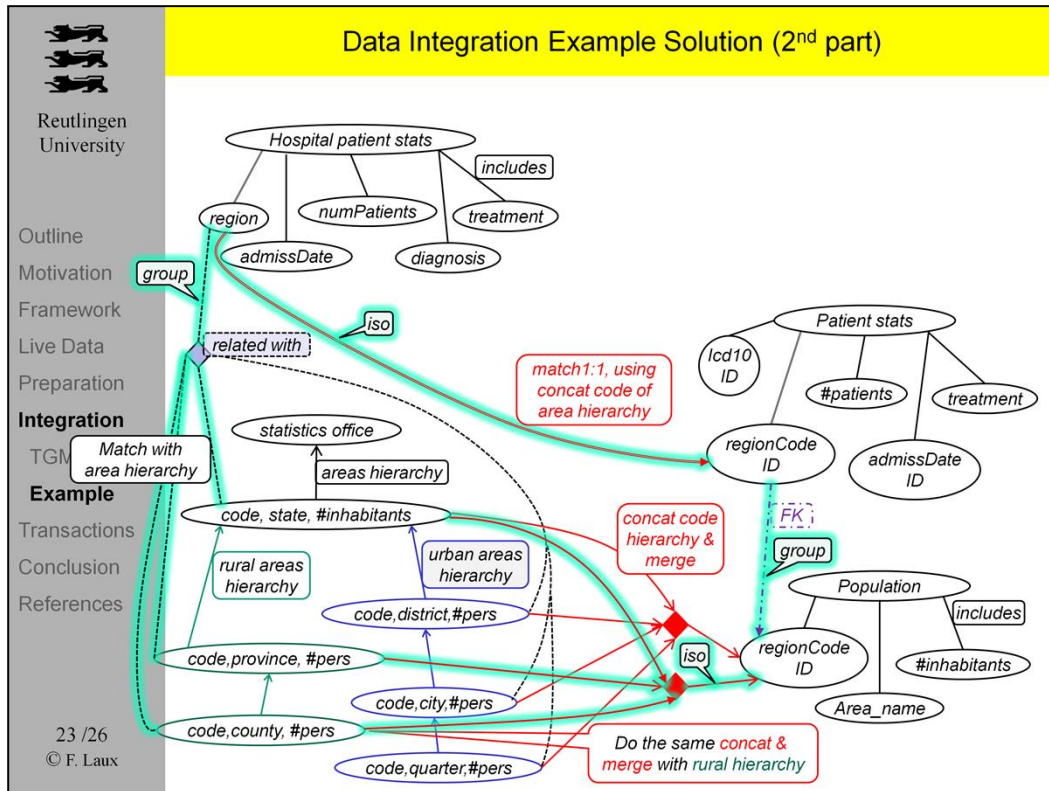
Equipped with the rules and techniques from the previous slides we are able to attack our integration example. In order to use the previously presented matching and mapping rules we first transform all data structures into a TGM.

Here we show the first part of our data integration example. The source and target schemata are already depicted as TGM. The graph of the population table is suppressed for the moment to not overload the picture.

First, we identify matching schema meta-data. The treatment field of the hospital patient statistics match with the treatment of the target schema on the right. Further matches are numPatients, admissDate, and region. Both match semantically with #patients, admissDateID, and regionCodeID of the target schema. The Hospital patient stats represents an isomorphism on the Patient statistics of the target.

Next we merge diagnosis from the hospital and description from ICD10 to lookup the ICD10 classifier.

We check the multipath mapping (green glow) between Hospital patient stats and regionCodeID. Both mappings are commutative because both projections π are compatible and the isomorphisms iso are 1-1 mappings. Hence their connection in series give the same result.




In the second part of the schema integration we focus on the population. We need to „flatten“ the hierarchy to a relational representation. This can be done by a concatenation and merge mapping to generate the regionCodeID. The same is done with the rural area hierarchy.

Finally we see that there are 2 paths (green glow) from region to regionCodeID of Population. One runs via the area hierarchies and the other is a direct match due to the isomorphism between region and regionCodeID.

This concludes our data integration example.

Transaction support



Reutlingen University

- Outline
- Motivation
- Framework
- Live Data
- Preparation
- Integration
- TGM
- Example
- Transactions**
- Conclusion
- References

24 / 26
© F. Laux

↪ *Some data require transactional guaranties for manual updates, corrections, and processing*

- ☞ These Apps need permission to write data

↪ *Solution*

- ☞ ReadCheck make distributed transactions feasible in loosely coupled environments
- ☞ ReadCheck can be used for an Optimistic Concurrency Control (OCC) like Row Version Verifying (RVV) [Laiho2010]

↪ *Issues*

- ☞ Update and insert transaction must operate on the source data
 - ⇒ only possible when data that is passed one-to-one to an App
 - ⇒ It does not work on aggregated data

Some data require transactional guaranties for manual updates, corrections, and processing. For instance if data is apparently incorrect or missing.

These Apps need permission to write data back to the source. If no permission exist, as work around the virtual integrated Database – apart from the cache – may have some tables that contain data that was missing or incorrect in the source data.


These data could be merged or replace source data when queried.

The already mentioned readCheck validator can be used to build an optimistic concurrency control (OCC) mechanism which is useful in a distributed, loosely coupled Transaction environment.

The mechanism works the same way as the RVV proposed by Laiho et al. During the validation of the transaction the RVV is check if it has changed since the beginning of the transaction.

Update and insert transaction must operate on the source data. But this is only possible when data that is passed one-to-one to an App.

It does not work on aggregated data.

 <p>Reutlingen University</p> <p>Outline Motivation Framework Live Data Preparation Integration TGM Example Transactions Conclusion References</p> <p>25 / 26 © F. Laux</p>	Lessons learned
	<p>↳ <i>Live Data Integration is possible with high quality and freshness if</i></p> <ul style="list-style-type: none"> ↳ Sources provide Live Views of data ↳ A mediated data schema is used for integration ↳ TGM helps to match, map, and merge data for integration ↳ Data is prepared in 5 steps for quality improvement <p>↳ <i>Key points for successful integration</i></p> <ul style="list-style-type: none"> ↳ Cooperation of data sources is necessary ↳ Careful semantic analysis and preparation of data is required to ensure quality data ↳ <i>The integration process is iterative as most aspects are interwoven</i>

Our conclusion is:

Live Data Integration is possible with high quality and freshness if first of all data sources are willing to provide Live Views of data.

Second, a mediated integration schema is used for a consolidated view on the data.

Third, the TGM helps to match, map, and merge the data which is prepared in 5 steps for quality improvement.

The key points for a successful integration are cooperation with the data sources, careful analysis of the data semantic and preparation of data is required to ensure the highest possible quality data

The integration process is iterative as most aspects are interwoven. For example if a data item turns out to often has wrong or missing values, then it is advisable to look for a more reliable data source and integrate it instead.

	References
 Reutlingen University	<p>[Crowe2017] M. Crowe et al., „Data Validation for Big Live Data“, <i>DBKDA 2017, Barcelona, Spain</i>, ISBN13: 978-1-61208-558-6</p>
Outline	<p>[Laux2017] F. Laux, “Using the Graph-Model for Schema and Data Mapping”, <i>Talk at DBKDAWEB/GraphSM, Barcelona, Spain</i>, URL: https://www.iaia.org/conferences2017/filesDBKDA17/FritzLaux_GraphModelForMapping.pdf</p>
Motivation	
Framework	
Live Data	<p>[Laux2020] F. Laux, “The Typed Graph Model”, <i>DBKDA 2020, Lisbon, Portugal</i>,</p>
Preparation	<p>[Sim2005] A. Simitsis et al., “Extraction-Transformation-Loading Processes“, in <i>Encyclopedia of Database Technologies and Applications, 2005</i>, ISBN13: 9781591405603, DOI: 10.4018/978-1-59140-560-3.ch041</p>
Integration	
TGM	
Example	
Transactions	<p>[Kemp2010] H.-G. Kemper, H. Baars, and W. Mehanna, <i>Business Intelligence – Grundlagen und praktische Anwendungen</i>, Vieweg+Teubner Verlag, 2010, ISBN13: 9783834807199</p>
Conclusion	
References	<p>[Caf2009] Michael J. Cafarella, <i>Extracting and Managing Structured Web Data</i>, PhD-Dissertation, University of Washington, 2009</p>
26 /26 © F. Laux	<p>[Laiho2010] M. Laiho and F. Laux, “Implementing Optimistic Concurrency Control for Persistence Middleware Using Row Version Verification,” <i>DBKDA 2010</i>, pp. 45-50, DOI: 10.1109/DBKDA.2010.25.</p>

In [Crowe2017] the term live data was introduced and a technical implementation for virtual integration was presented.

The paper of [Laux2020] introduced the TGM as an extension of the property graph model (PGM) that was used in the presentation [Laux2017] as a tool to help with integration of different data structures.

The contributions of [Sim2005], [Kemp2010], and [Caf2009] name and propose process steps for preparation and transformation of data.

[Laiho2010] presents a Row Version Verification (RVV) mechanism to avoid the „lost update“ problem in typical web applications (disconnected scenarios).