Part 3
Knowledge Graphs

1 Part 0 - Course Organization

2 Part 1 - The Reuse Problem

3 Part 2 - State of the Art

4 Part 3 - Knowledge Graphs

5 Part 4 - Entity Base

6 Part 5 - The iTelos Methodology

7 Part 6 - KG Evaluation and Exploitation

Part 3.1
Data Integration

1 Data Integration

2 Knowledge Bases

3 Knowledge Graphs

Data integration (1)

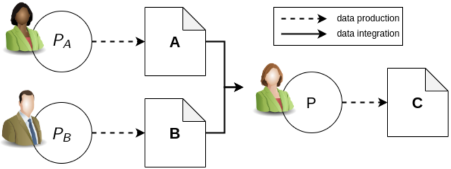
- The previous lecture presented the available resources for each layer considered by the stratification of the information.
- The language, schema and data value resources, described above, show how different types of information can be represented.
- Nevertheless, to enhance the diversity as a feature included in data, we need to **integrate** such resources.

Data integration (2)

- The information stratification leads to **different types of Data Integration** (DI).
  - Integrate data of the **same information layer**.
    - integrate different language data;
    - two or more, data schema, or ontologies;
    - two or more (data value) dataset.
  - Integrate data of **different information layers**.
    - Integrate one dataset with a new language (different from the one used to represent its data).
    - Integrate two datasets by using a third data schema (or ontology) different from the single data schema adopted in the two datasets.
    - Integrate an ontology with a language, to produce multilingual knowledge resources.

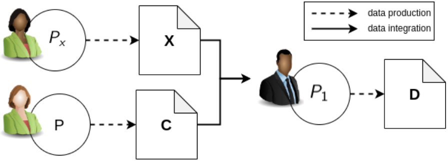
Data adaptation & evolution (1)

- In all the above cases, the DI can be defined in two different phases, which are **strictly related to the data reuse problem**.
  - **Data Adaptation**: this phase defines the first time that two resources, A and B, need to be integrated.
    - The resources A and B, have been created for specific purposes P_A and P_B, and they have not been modified, and/or, integrated with any other resource, to satisfy a different purpose.
    - A and B are then integrated, following a new purpose P, thus producing C as integration output. **Reuse of A and B.**



Data adaptation & evolution (2)

- **Data Evolution:** in this phase the result of the adaptation integration of A and B, is in turn integrated to satisfy a new purpose P_1 (or an extended version of the P).
  - The adaptation integration output C, is integrated with new resources to satisfy a new purpose P_1, thus creating the result of the evolution integration D.
    Reuse focused on C.



Data Integration - State of the art

- The DI was born to exploit the information present (stored, represented, modelled) in different data bases, or provided by different data sources.
- However, as we saw previously, the data heterogeneity issues cause multiple problems to be addressed:
  - language heterogeneity;
  - semantic heterogeneity;
  - format heterogeneity;

Data Integration - State of the art

- We can divide the evolution of DI in two main periods in time, where the DI has been used for **different purposes and in different contexts**.
  - **Corporate DI** (90s, early 2000s) with the use of **Knowledge Bases (KBs)**.
  - **Semantic Web DI** (from early 2000s, to the present day) with the use of **Knowledge Graphs (KBs)**.

Part 3.2
Knowledge Bases

1 Data Integration

2 Knowledge Bases

3 Knowledge Graphs

Knowledge Bases (1)

- In the 90s the availability of the data starts to increase significantly;
- nevertheless, the impact of using data for digital solutions (applications and services), still remains **in the hands of tech companies**.
- However, the need of such companies evolves, **from the data production to the integration of data** previously created for different purposes.
- In restricted contexts, like the private companies systems, the solution was the development of **Knowledge Bases (KB)**.

What is a Knowledge Bases ?

- ”The **knowledge base is a collection of interlinked descriptions of entities** (real-world objects, events, situations or concepts) that enables storage, analysis and reuse of this knowledge in a **machine-interpretable** way. As a result, it empowers search engines and other content retrieval applications to interpret text and match it to advanced queries.” [26]

- Most people are familiar with traditional, relational databases. There are cells and tables filled with letters and numbers. Years of refinements and optimizations have ensured that organizations can manage phenomenal amounts of data. But as the American author Clifford Stoll said it best: '**Data is not information, information is not knowledge**'.

- Knowledge bases, on the other hand, abstract away from a simple database to create an organized collection of data that **is closer to how the human brain organizes information**. Knowledge bases add a semantic model to the data, which includes a formal classification with **classes, subclasses, relationships and instances** (ontologies and dictionaries), on one hand, and rules for interpreting the data, on the other.

[26] Ontotext/Knowledge Hub

Knowledge Bases (2)

- In other words a KB, or a KB System (KBS), is a way to **abstract the information** about entities **contained into a database**.
  - Notice that, considering the stratification of the information a KB is data schema layer resource.
  - For this reason the data schema resource are also called **Knowledge Resources** (and by consequence Knowledge information layer).

- KBs are often represented by ontological models.

- Such models have been used to integrate multiple database, with the objective to serve a unique purpose (application/service)

Data Virtualization

- The usage of KBs within the corporate DI, has been implemented by specific DI techniques categorized as **data virtualization techniques**.
- This name ”virtualization” comes from the fact that **the original data (databases), to be integrated are not touched (modified)**, but a virtual model (abstraction) of that is created to implement the DI.
  - **Virtualization strategies**: aim at providing a unique interface for two, or more, data sources, for accessing the data without extraction and transformation data.
- The most known set of virtualization strategies are called:

  Ontology Based Data Access (OBDA).
  - They are all based on the modeling of **ontologies and vocabularies**, interfacing the different data sources.

OBDA - Local As a View

- **LAV - "Local As a View"** : this family of OBDA techniques assumes to query data as they are provided by the data sources.
  - **single-ontology** : a single ontology is queried to access all the data sources.
  - **multiple-ontology** : an ontology for each data source can be queried to access the data.
  - **hybrid approach** : as the multiple-ontology technique there are more ontologies, but in this case the queries are uniformed by a unique vocabulary used to query the data.

DI Virtualization strategies - GAV

- **GAV** - **"Global As a View"** : this family of OBDA techniques assumes to query data from the point of view of the application (services, or users) that needs to perform the query, by exploiting an application-specific ontology modeling.

DI Virtualization strategies - Summary

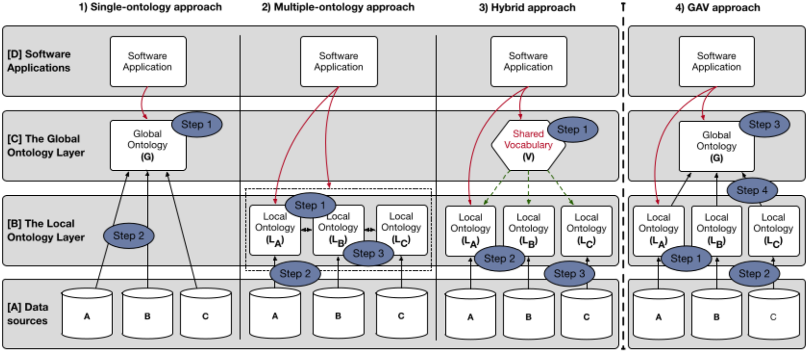


Figure 1: Three variants of OBDI from [75]: (1) single-ontology, (2) multiple-ontology, (3) hybrid, and an additional OBDI variant (4) Global-as-View (GAV).

(Explanation: Red arrows indicate access from an application to data, black arrows represent transformation/virtual access to the data; dotted green arrows represent implicit relations between involved ontologies, and numbered items show the sequence of system development. The dotted rectangle refers to the federation of local ontologies. Section 5.1 explains the additional OBDI variant (4) *Global-as-View* (GAV).)

DI Virtualization strategies - Limitations [27]

- In a GAV approach, changes in information sources, or adding a new information source, requires **revisions of a global schema and mappings** between the global schema and source schemas.
- In a LAV approach, automating query reformulation has **exponential time complexity with respect to query and source schema definitions**.

[27] Xu, Li, and David W. Embley. "Combining the best of global-as-view and local-as-view for data integration." Information systems technology and its applications, 3rd international conference ISTA'2004. Gesellschaft für Informatik eV, 2004.

Part 3.3
Knowledge Graphs

1 Data Integration

2 Knowledge Bases

3 Knowledge Graphs

Knowledge Graphs (1)

- Then, at the end of 90s, early 2000s, **the World Wide Web starts to be browsable!**
- Unlike the corporate data, the WEB requires a different need.
- A **semantic representation of multi-purpose information** which does not depend on previous databases or applications, but instead on **the human point of view**.
  - What has been defined by Tim Berners Lee as **the Semantic Web**. [28]
- Such a need, together with the wider (global) context considered, respect to the restricted corporate one, led to DI (and data representation) solutions more flexible and interoperable.
  - **The Knowledge Graphs (KGs)**

[28] Berners-Lee, Tim, James Hendler, and Ora Lassila. "Web Semantic." Scientific American 284.5 (2001): 34-43.

Knowledge Graphs (2)

- A knowledge graph is a **graph-based data model that represents knowledge as a collection of interconnected entities**, where nodes represent entities, and edges represent relationships between entities.
    - It is designed to capture complex relationships and interconnections in data, providing a more expressive and flexible representation.
- The main difference between KGs and KBs is that the latter is used to abstract (virtualise) **already existing databases/data sources**, while KGs are used as a **different data representation model**, used to enhance the data semantic.

What is a Knowledge Graph ?

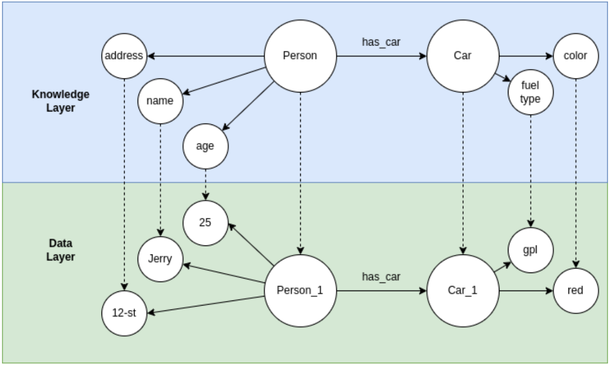
- A Knowledge Graph K, can be defined as follows:

$$\mathbf{KG = (E, D, R, A)}$$

Where:

- **E**: is the set of real-world objects types, called *Entity Types* (or ETypes).
- **D**: is the set of real-world objects representations, called *Entities*. The Entities are ETypes instantiation.
- **R**: is the set of properties used to denote the ETypes. The elements of R, can be properties related to a single EType, called *data properties*, or properties used to define relations among different ETypes, *called object properties*.
- **A**: is the set of property values denoting the attributes of the Entities. Each attribute, associated to one and only one property, instantiates the relative data/object property.

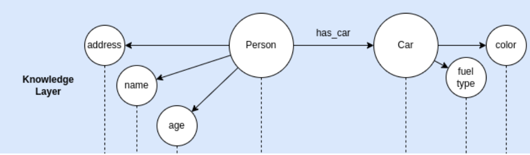
KG example



- E = {Person, Car}
- D = {Person_1, Car_1}
- R = {address, name, age, color, fuel type, has_car}
- A = {12-st, Jerry, 25, red, gpl}

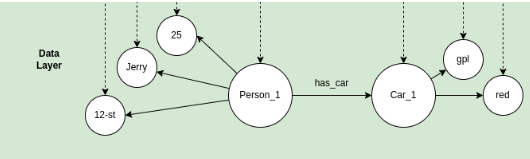
Knowledge Layer

- The KG's Knowledge Layer is composed by the elements of E (ETypes) plus the element of R (properties definition).
- It defines the KG's structure (or schema).
- It is usually defined using an ontology modeled to represent the information to be maintained in the KG.



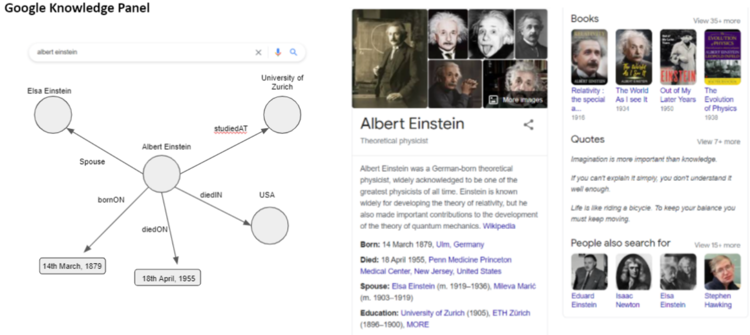
Data layer

- The KG's Data Layer is composed by the elements of D (Entities) plus the element of A (attributes definition).
- It contains the data values instantiating the KG's structure.



KG-based Apps - examples

- Google Knowledge Panel



Albert Einstein

Theoretical physicist

Albert Einstein was a German-born theoretical physicist, widely acknowledged to be one of the greatest physicists of all time. Einstein is known widely for developing the theory of relativity, but he also made important contributions to the development of the theory of quantum mechanics. Wikipedia

Born: 14 March 1879, Ulm, Germany

Died: 18 April 1955, Penn Medicine Princeton Medical Center, New Jersey, United States

Spouse: Elsa Einstein (m. 1919–1936), Mileva Marić (m. 1903–1919)

Education: University of Zurich (1905), ETH Zürich (1896–1900), MORE

Books View 35+ more

Relativity : the special a... 1916

The World As I see It 1934

Out of My Later Years 1950

The Evolution of Physics 1938

Quotes View 7+ more

Imagination is more important than knowledge.

If you can't explain it simply, you don't understand it well enough.

Life is like riding a bicycle. To keep your balance you must keep moving.

People also search for View 15+ more

Eduard Einstein

Isaac Newton

Elsa Einstein

Stephen Hawking

Figure: Mohit M., A guide to Knowledge Graphs, Aug 30, 2021

KG-based Apps - examples

- InteropEHRate EU project

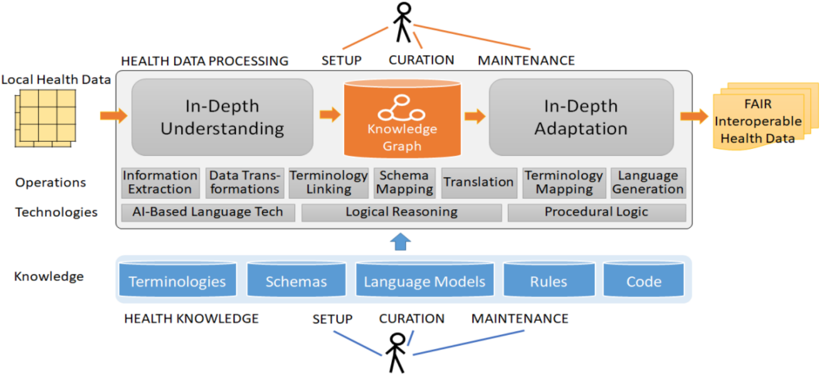


Figure 1: High-level architecture of the InteropEHRate Health Services and the way they are overseen by a human data manager.

KG-based Apps - examples

While there are several small-sized and domain-specific KGs, on the other hand, we also have many huge-sized and domain-agnostic KGs that contain facts of all types and forms.

- **DBpedia**: is a crowd-sourced community-based effort to extract structured content from the information present in various Wikimedia projects.
- **Freebase**: a massive, collaboratively edited database of cross-linked data. Touted as "an openly shared database of the world's knowledge". It was bought by Google and used to power its own KG. In 2015, it was finally discontinued.
- **OpenCyc**: is a gateway to the full power of Cyc, one of the world's most complete general knowledge base and commonsense reasoning engines.
- **Wikidata**: is a free, collaborative, multilingual database, collecting structured data to provide support for Wikimedia projects.
- **YAGO**: huge semantic knowledge base, derived from Wikipedia, WordNet, and GeoNames.

Data Materialization

- The usage of Knowledge Graphs increased a lot within the data integration community, thanks to their suitability within different domain of interest.
- Unlike the DI virtualisation strategies, the usage of KGs requires the creation and manipulation of the data.
- For this reason the **Knowledge Graph Construction (KGC)** is a DI technique included in the **data materialization** category.
  - **Materialization strategies**: are based on ETL procedures used to extract, and transform, the data to be integrated from the respective data sources.

KGC

- DI based on KGC is a process that involves different sub activities defined as follows:
  - Data collection/extraction.
  - Schema definition/alignment.
  - Data cleaning & formatting.
  - Entity recognition & identification.
  - Data mapping.

DI Materialization strategies - KGC limitations

- **Missing of** standard **methodologies** for KG generation;
  - the KGs produced are often, too application-specific, causing an increase of the KG's evolution cost.
- **Missing of frameworks** (tools and application) covering the whole KGC process.
- **Technical skills required** (data management and knowledge modeling) for the KGC process implementation;
  - the KG's final user (who is usually the domain expert) usually doesn't have such expertise.