

Semantisches Wissensmanagement im Unternehmen: Konzepte, Technologien, Anwendungen

Prof. Dr. Stefan Linus Zander

Kapitel 3.2: Datenmodellierung mit Semantic MediaWiki

Vorbemerkung

Ein Großteil unseres menschlichen Daseins und Wirken definiert sich über **Beziehungen** zu anderen "Dingen" (hier im weitest möglichen Sinne zu verstehen). Möchte man diese Gegenstandsbereiche in technischen Systemen möglichst präzise und "naturgetreu" abbilden, so braucht es **ausdrucksmächtige Beschreibungsformate** und **Datenstrukturmodelle**. Viele bekannte Datenstrukturmodelle und Formate sind zwar sehr effizient in ihrer maschinellen Verarbeitung, ihnen fehlt es aber an der notwendigen Ausdrucksmächtigkeit und Erweiterbarkeit, d.h., diese sind nicht in der Lage, die komplexen Beziehungen zwischen Dingen präzise und widerspruchsfrei (=unambiguitiv) abzubilden.

Beispiel

Versuchen Sie nachfolgend beschriebenen Sachverhalt in einem Datenmodell ihrer Wahl möglichst widerspruchsfrei und eindeutig abzubilden.

"In seiner konstituierenden Sitzung vom 25.03.2019 beschloss der Fachbereichsrat in Bezug auf Berichtspunkt Nr. 5 des Protokolls vom 17.02.2019 mit 10 'Ja'-Stimmen, 0 Enthaltungen und keiner Gegenstimme:

Prof. Dr. Kai Renz wird zum kommenden Sommersemester neues Mitglied des Stundenplanerteams."

Überlegungen zum Datenmodell

Bei der Durchführung des vorherigen Beispiels werden Sie sehr schnell feststellen, dass die Repräsentation in Form eines **konzeptuellen Graphen** mit **Knoten** und **Kanten** die Beziehungen zwischen den “Dingen” in einer natürlichen Darstellung am nächsten kommt.

Mit den Informationen aus "Kapitel 2: Technologische Grundlagen" können wir diesen Graphen weiter verfeinern.

- Ausschnitte aus Gegenstandsbereichen lassen sich am besten unter Reduzierung struktureller Heterogenität als **konzeptueller Graph** darstellen
- **URIs & IRIs** ermöglichen die eindeutige Identifizierung von “Dingen” und dienen als **Adressierungsschemata**
- Dinge sind **Information-** und **Non-Information Resources**
- Wir unterscheiden zwischen **Designator** und **Designatum**
- Zusammen mit dem Konzept der **Content Negotiation** ermöglichen sie eine Unterscheidung zwischen technischen- und real-weltlichen Repräsentationen
- Zur Beschreibung der Bedeutung einer Beziehung braucht es eine **maschinen-verarbeitbare Semantik**
- Die Semantik wird durch die Terme und logische Theorie, auf der eine **Ontologiesprache** definiert ist, festgelegt
- Beziehungen sollten durch **wohl-definierte Terme** aus bekannten Ontologien oder Vokabularen repräsentiert werden
- Ressourcen sollten durch URIs/IRIs von bekannten **Domänen** (bspw. DBpedia) identifiziert werden
- Beziehungen sollten als **First-Class Elemente** behandelt werden
- Sachverhalte werden in einem **Triple-Pattern** kodiert

Wie werden Informationen in SMW kodiert ?

Semantic MediaWiki encodes information in the form of **facts**

Example: Berlin has a population of 3,520,031 people

A **fact** can be represented in a **triple-based linguistic structure**:

1. The **subject** of a sentence, e.g., "Berlin"
2. The **predicate**, e.g., "has a population"
3. The **object**, e.g., "3,520,031 people"

These three parts can be split into a **relation** and a **statement**:

- **Relation** ~> Berlin has a population.
- **Statement** ~> The population is "3,520,031".

A statement consists of a **property** and its **value**:

- **Property** (i.e. predicate), e.g. "Has population"
- **Value** (i.e. object or literal), e.g. "3,520,031"

Adding a statement (property-value pair) to a page (the subject) is called **annotation**.

As a consequence, facts are encoded as **property-value pairs** on wiki pages.

Encoding Facts in Semantic MediaWiki

This **triple** can be expressed with elements from SMW's Knowledge Representation Framework¹:

1. **Subjects** are general **wiki pages** or **subobjects**²
 - e.g. a wiki page named `Berlin` in the main namespace
2. **Predicates** are **wiki pages** in the `Property` namespace
 - e.g., a wiki page named `has Population` in the `Property` namespace
3. **Objects** are **literals**, **wiki pages** or **subobjects**²
 - e.g., value `3,520,031` in case the object is a literal
 - e.g., a page named `Germany` in case the statement's object is another wiki page
 - e.g., an anonymous **blank node** in case the object is a subobject

Examples

(page – property – value):

Cologne	has population	1.017.155
Germany	has capital	Berlin
Spinoza	born on	24 Nov 1632

The **triple-based model** employed by Semantic MediaWiki is inherited from W3C's **Resource Description Framework (RDF)** specification.

A **page-property-value** triple resembles the **subject-predicate-object** triple pattern of RDF.

¹ KRF = Knowledge Representation Framework, i.e., a set of rules and description primitives for encoding information in a certain format.

² We will learn about subobjects in Lecture 4.

Part 2: How to Encode Information in Knowledge Graphs

Motivating Example

*Matthias Frank is an employee of the FZI Research Center for Information Technology working on the BigGIS project.
BigGIS is an ongoing research project started at April 2016 and deals with real-time big data and semantic technologies.*

Steps:

1. Identify **Instances** (Resources)
2. Identify **Concepts** (Classes)
3. Identify **Properties** and **Data** (Numbers, Literals etc.)
4. Represent as **Conceptual Graph** with **Stereotypes**
5. Transform into **Serialized Graphs** using MediaWiki syntax

Step 1: Identify Instances

*Matthias Frank is an employee of the FZI Research Center for Information Technology working on the BigGIS project.
BigGIS is an ongoing research project started at April 2016 and deals with real-time big data and semantic technologies.*

Step 2: Identify Concepts

*Matthias Frank is an **employee** of the FZI Research Center for Information Technology working on the BigGIS **project**.
BigGIS is an ongoing **research project** started at April 2016 and deals with real-time big data and semantic technologies.*

Step 3a: Identify Properties...

*Matthias Frank **is** an employee of the FZI Research Center for Information Technology **working on** the BigGIS project.
BigGIS **is** an ongoing research project **started at** April 2016 and **deals with** real-time big data and semantic technologies.*

Step 3b: ...and Datatypes

*Matthias Frank is an employee of the FZI Research Center for Information Technology working on the BigGIS project.
BigGIS is an ongoing research project started at April 2016 and deals with real-time big data and semantic technologies.*

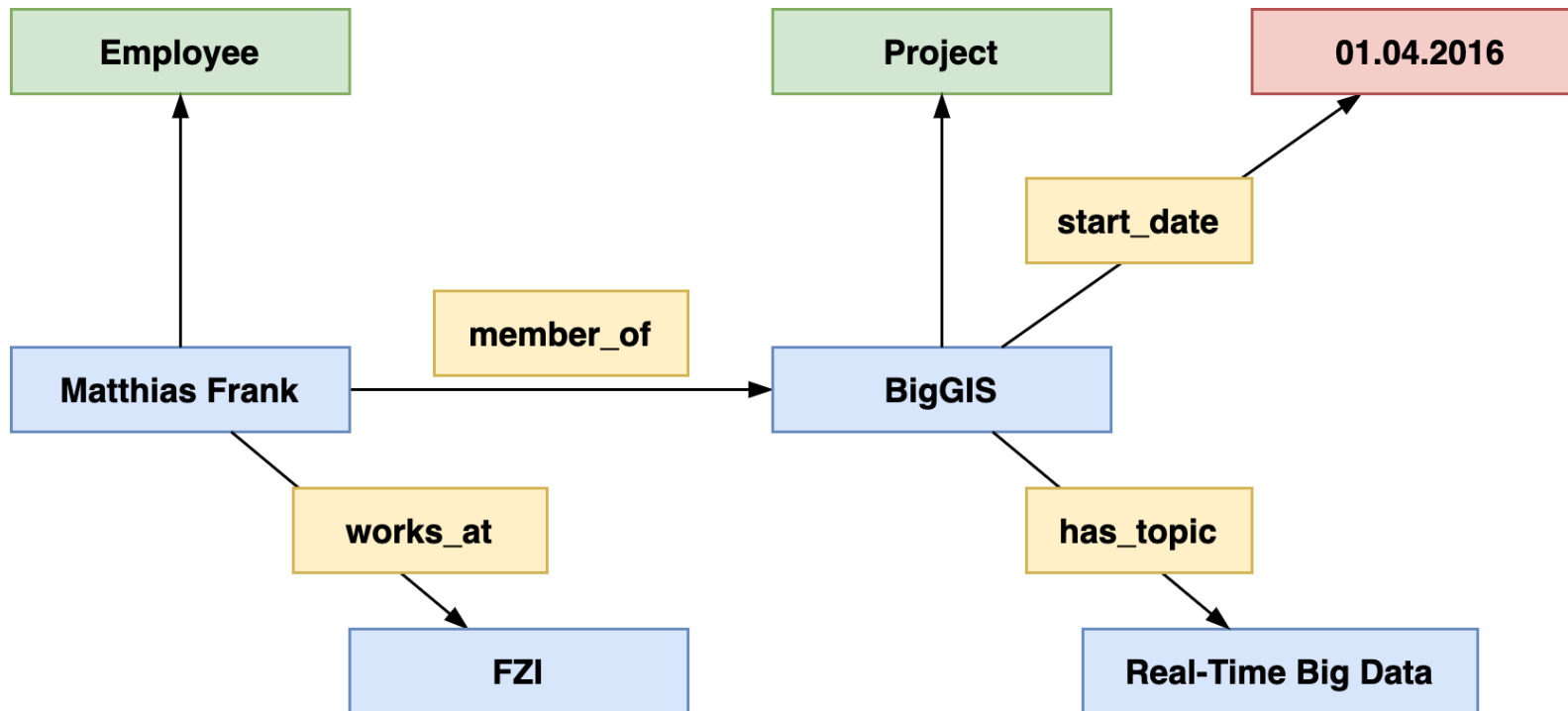
Instances versus Literals

Question: Should an entity be represented as a literal or instance ?

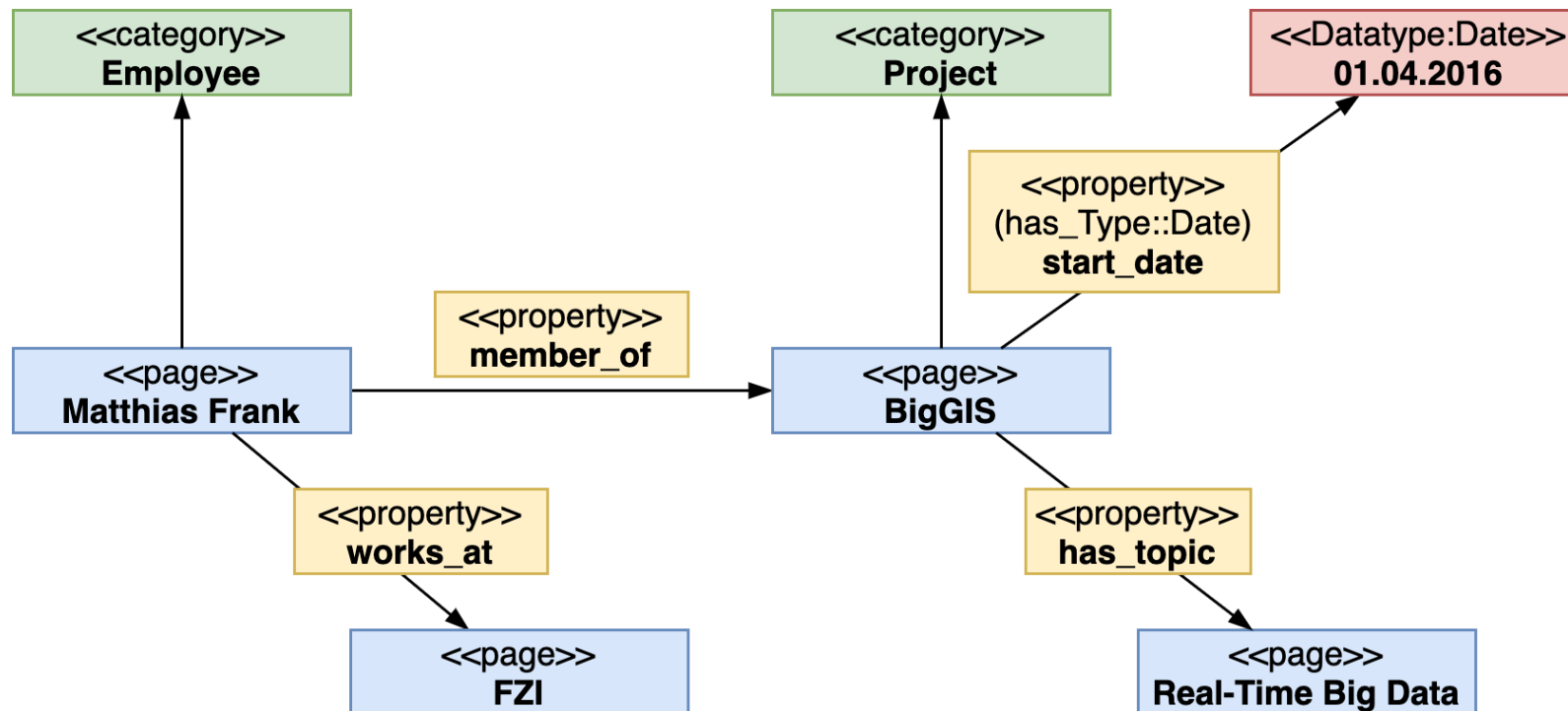
- Sometimes it is difficult to determine whether an entity should be represented as a literal (using the `text` datatype) or as an instance
 - e.g., should a research field be represented as a literal or as an individual page in the system?
- Since this is a **modelling issue** which depends on the situation at hand, there is no definite answer to this question
 - A **trade-off** in terms of performance versus flexibility and extensibility is necessary.
- Representing an entity as **instance** (datatype `page` in Semantic MediaWiki) allows for more flexibility and extensibility
 - e.g. the page of a research field can **list projects** or works in which this research field is used.
 - It can host **additional information** about the field that can be used in queries
- If **no additional information** is needed for an entity, it should be represented as a **literal**
 - This helps save space (disk and memory) and increases query performance on large knowledge bases

Step 4a: Representation as Conceptual Graph

*Matthias Frank is an employee of the FZI Research Center for Information Technology working on the BigGIS project.
BigGIS is an ongoing research project started at April 2016 and deals with real-time big data and semantic technologies.*



Step 4b: Conceptual Graph with Stereotypes



Step 5: Transformation into Serialized Graphs

It is **important** to **separate the facts** on their respective pages and determine the **direction** of a relation.

Syntax on the **Matthias Frank**-Page:

[[Matthias Frank]] is an employee of the [[works_at::FZI Research Center for Information Technology]] working on the [[member_of::BigGIS]] project.

[[Category:Employee]]

Syntax on the **BigGIS**-Page:

[[BigGIS]] is an ongoing research project started at [[start_date::April 2016]] and deals with [[has_topic::Real-Time Big Data]] and [[has_topic::Semantic Technologies]].

[[Category:Project]]

Important: Consider the Relationship Semantics

- SMW's Knowledge Representation Framework is a **directed Hypergraph**
- Therefore, the direction of a relation matters
- The direction of a relation bears certain **semantics** (**structural** and **model-theoretic**)
- These semantics are relevant in **inline queries**
 - e.g. `has_member` versus `is_member_of` – one can be used on employee pages, the other on project pages

Try for Yourself

*Bayern Munich is a professional football club based in Munich, Germany, that plays in the Bundesliga.
Bayern Munich was founded in 1900. They play at the Allianz Arena. The club has won 30 national league titles.*

Tasks

1. Identify all **instances**, **properties**, **classes**, **literals** etc.
2. Think about what **additional** classes, properties and literals are needed
3. Create the **conceptual graph**
4. Think about what **facts** will be represented on which page
5. Complement the above text using the **Semantic MediaWiki syntax** of the respective elements

Future Work

Here is a list of further, more advanced **modelling problems**:

- How can we state that Bayern Munich has won 30 Bundesliga Championship titles ?
- How can we express the years in which the Bundesliga titles were won ?
- How can we express in which season + year a Bundesliga title was won ?
- How can we express in which seasons Bayern Munich finished second ?
- How can we count all national and international titles won by a German football club ?

Hint: All the above modelling problems call for the use of **subobjects**!¹

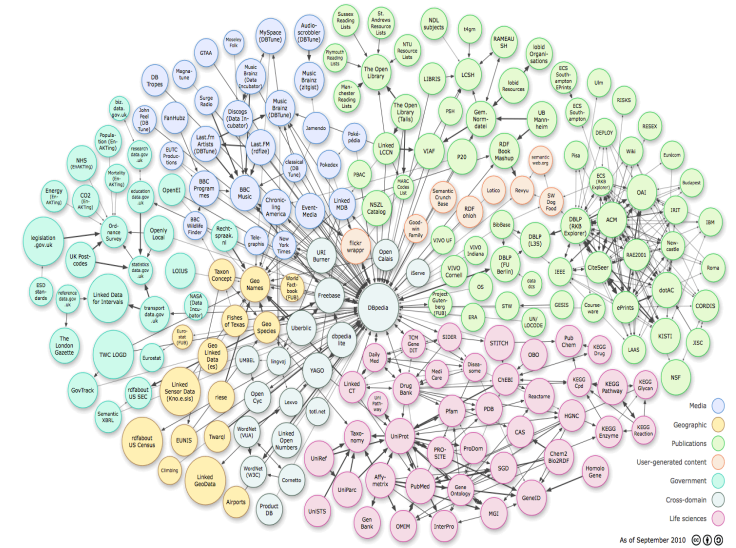
¹ We will learn about subobjects in [lecture 4](#) of this course.

Summary

Why Knowledge Graphs are useful

Knowledge graphs have many advantages

- Like most graph models, it more intuitively captures the way **we think about the world** as humans (as networks, not as tables), making it easier to design, capture, and query data.
- As a data model supported by W3C standards, it allows us to create **interoperable data and systems**, all using the same standard to represent and encode data while addressing **structural, schema and semantic heterogeneities**.



RDF versus Labeled Property Graphs:

There are many other methods that have been developed to handle this kind of complexity in RDF, including singleton properties and named graphs/quads. Additionally, an entirely different type of **non-RDF graph model**, denoted as **labeled property graphs**, allows users to attach properties directly to relationships.

However, labeled property graphs do not allow for interoperability at the same scale as RDF does – it is much harder to share and combine different data sets, and moving data from tool to tool is not as simple and straightforward.

Source: <https://enterprise-knowledge.com/rdf-what-is-it-and-why-do-i-need-it/>