UNIVERSITÄT BIELEFELD
Faculty of Technology

Data-driven ontology engineering with OTTR: benefits and challenges derived from practical experience

3rd OTTR user forum, 2022-05-11

Moritz Blum, Basil Ell
Bielefeld University
{mblum,bell}@techfak.uni-bielefeld.de

Data-driven ontology engineering with OTTR

- DiProMag Ontology for the semantic description of experiments: *production, characterization and prototypical application of magnetocaloric alloys*
- A-Box and T-Box should be populated: parallel development of templates and ontology
- manual and partly automated instantiation through domain-experts
  - propose extensions/changes of the ontology
  - understand the underlying structure
- semi-structured data is given - some experiments still exploratory without a fixed parameter space
  - measurement devices: input specification, settings, output
  - production process: sequence of operations applied to objects
  - goals and incentives

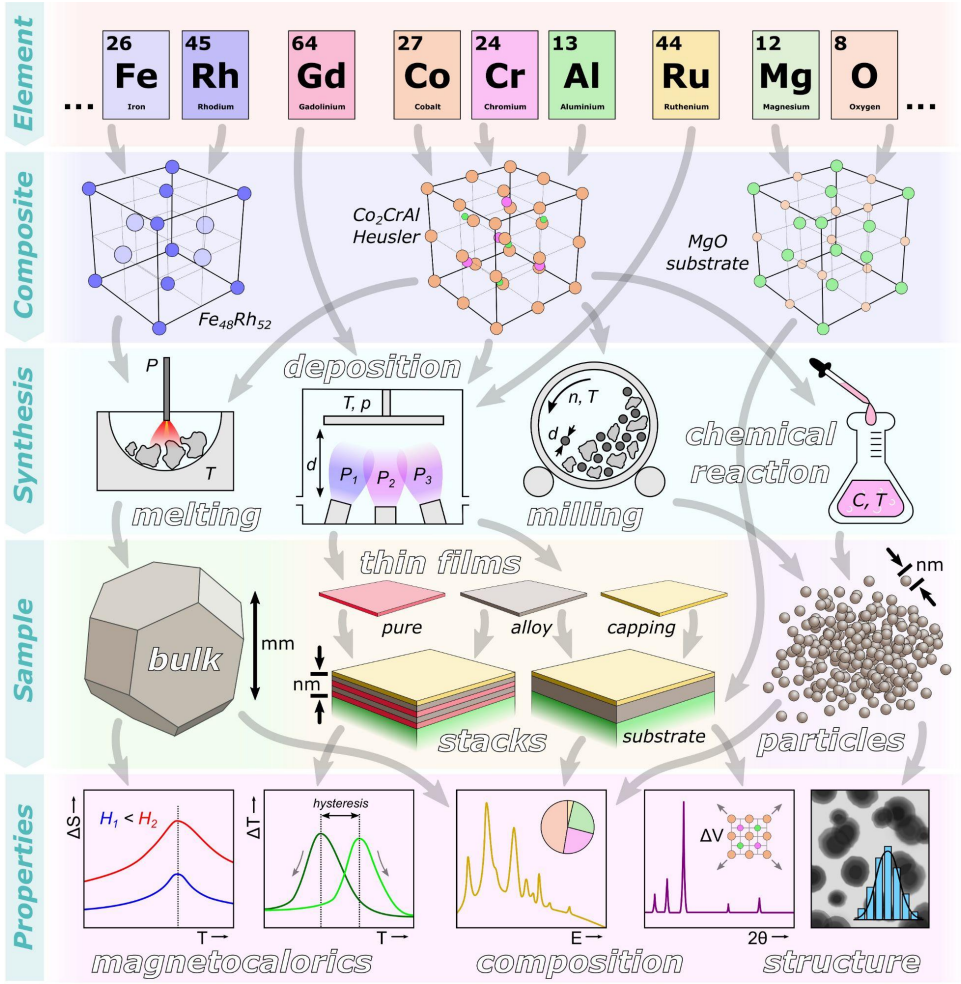
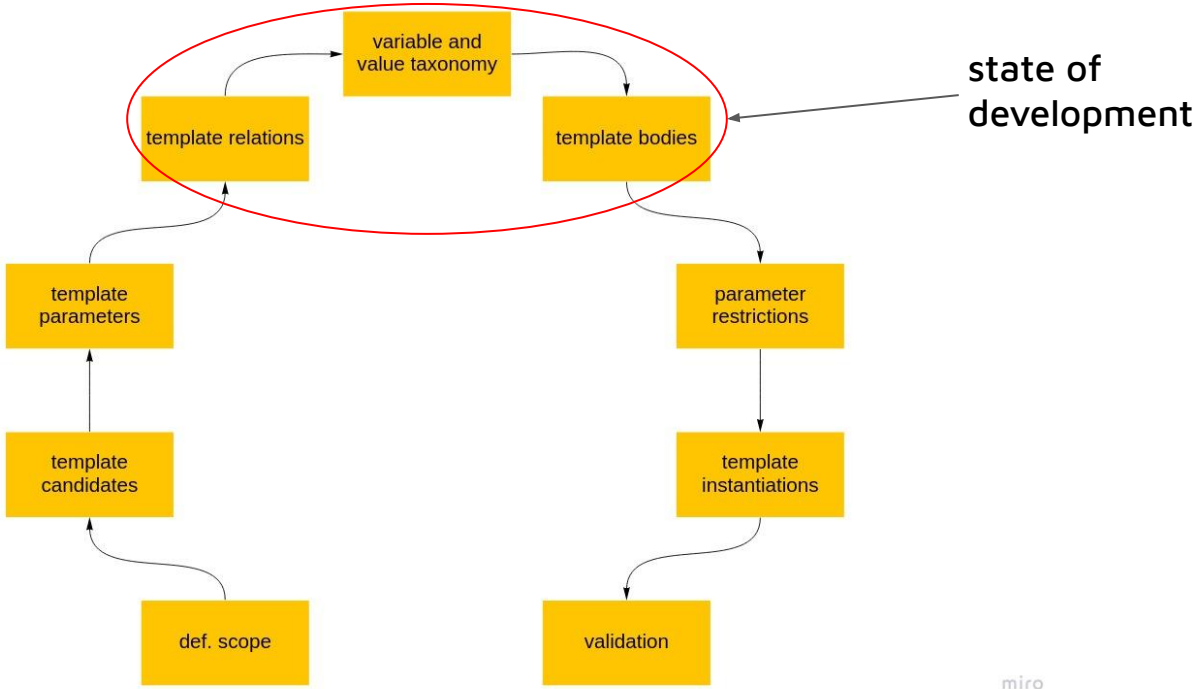


Fig: overview of the experiments carried out in two of our working groups

Read more about DiProMag on our project website: dipromag.de

Methodology

Iterative process that allows backward steps -> simplifies and improves communication with domain experts, enforces a bottom up / data-driven approach



UNIVERSITÄT BIELEFELD
Faculty of Technology

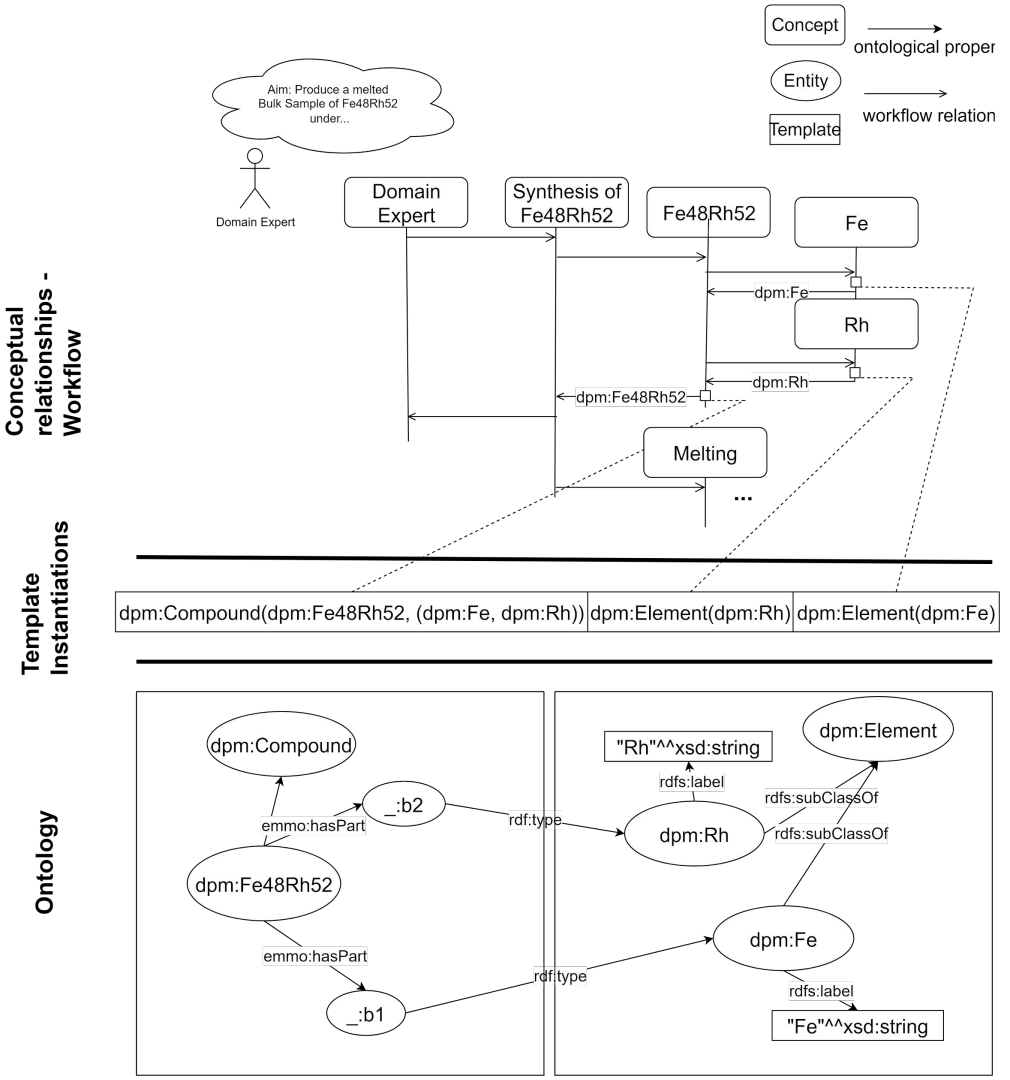
Communication with domain experts

- tables help to maintain an overview: datatype, parameter name, example
- documentation and variable naming
- coloring of rows to group concepts

dpm:Material[	
ottr:IRI ?material,	dpm:Co2CrAl0.0001Si
NEList<dpm:Element> ?elements,	(dpm:Co, dpm:Cr, dpm:Al, dpm:Si)
NEList<xsd:float> ?stoichiometry_portion_unit_at_percent,	(2,1,1,0.0001)
dpm:material_derived_from ?method_of_derivation,	dpm:addition_material_from
dpm:Material ?base_material,	dpm:Co2CrAl
dpm:User ?user,	dpm:BasilEll
xsd:date ?date,	“2022-05-11”^^xsd:date
xsd:string ?comment,	“Si for better… ”^^xsd:date
xsd:boolean ?verified]	“false”^^xsd:boolean

Workflows

- manual instantiation by domain experts requires workflows of template instantiations and naming conventions
- relations between templates define the workflows
- conceptual relationships are mirrored in the instantiated template signatures and the instantiated triples



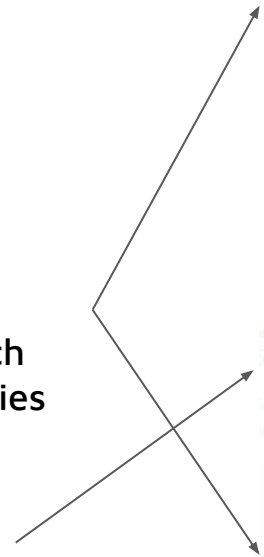
Experience

- strictly defined workflows reduce the requirement of naming conventions
- T-Box is distributed across templates vs. central T-Box template(s): duplication and knowing where → Protegé in parallel to maintain overview
- good experience with Semantic Media Wiki (SMW) as central platform for managing template definitions, instantiations, and documentation
- the OTTR SMW extension was developed and is maintained by us

SMW: OTTR Template definitions

the two separate templates are connected by instantiations which use the same entities

page name



Dpm:CompoundExample

Dpm | Discussion | ★ | Edit | History | Refresh

OTTR-Definition:

```
dpm:CompoundExample[ottr:IRI ?compound, NEList<dpm:Element> ?elements]::{
    zipMin | ottr:Triple(?compound, emmo:hasPart, ++(_:b1, _:b2)),
    zipMin | ottr:Triple(++(_:b1, _:b2), rdfs:type, ++?elements)
} .
```

Form Info:
The OTTR-Extension comes with an automated form creation, which simplifies the generation of instances of a template via input fields:

Create instance with form

Dpm:ElementExample

Dpm | Discussion | ★ | Edit | History | Refresh

OTTR-Definition:

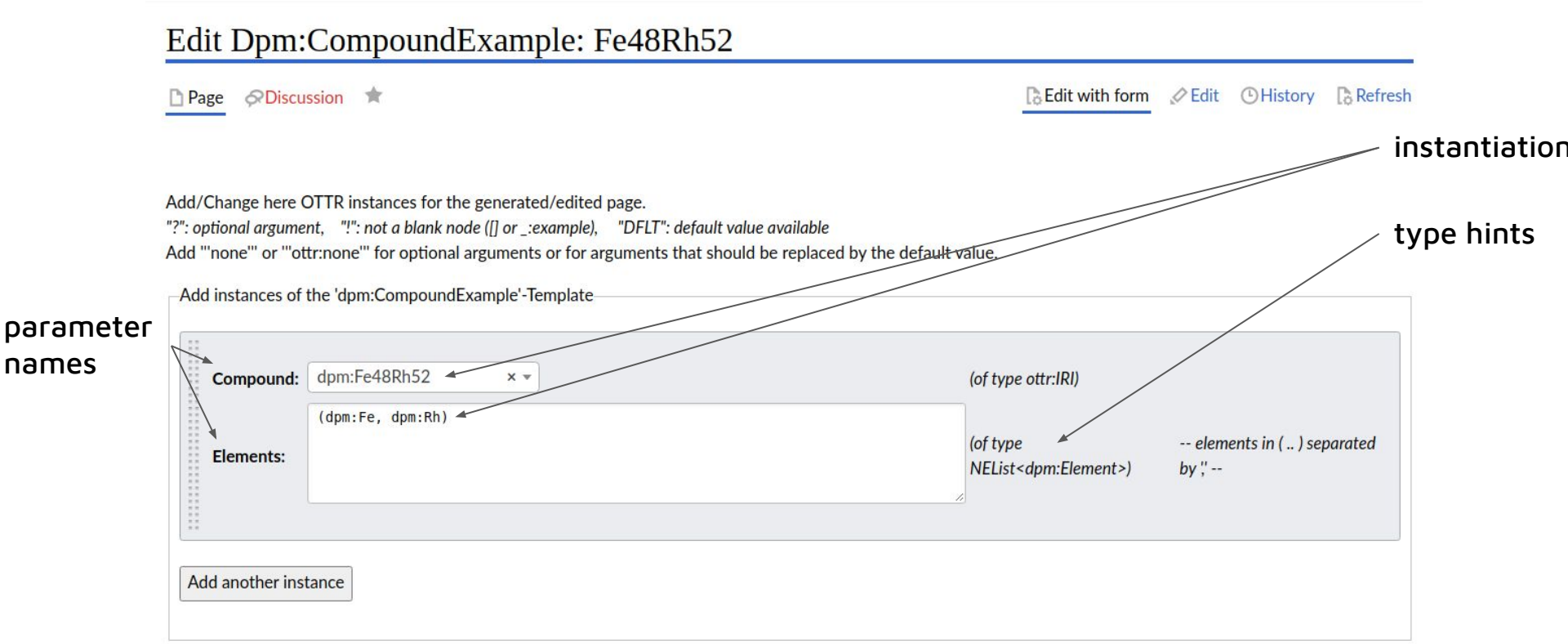
```
dpm:ElementExample[ottr:IRI ?element]::{
    ottr:Triple(?element, rdf:subClassOf, dpm:Element)
} .
```

Form Info:
The OTTR-Extension comes with an automated form creation, which simplifies the generation of instances of a template via input fields:

Create instance with form

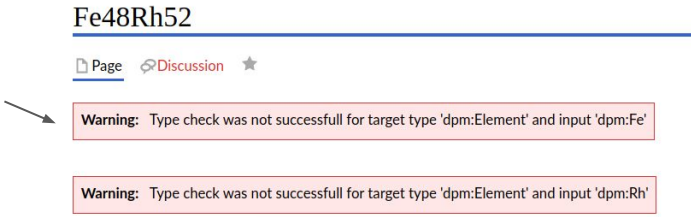
UNIVERSITÄT BIELEFELD
Faculty of Technology

SMW: Forms for OTTR Instantiations



SMW: OTTR Template Instances

type checks are only available for literals, not yet for classes



- templates are instantiated inside the Wiki by usage of the SMW template mechanism
- instance page shows triples, debug information, and SMW annotations like categories and properties defined on the template page
- retrieve data inside the wiki with #Ask

- *Number Init Triples:* **4**
- *Number Used IRIs:* **0**
- *Max Depth:* **2**
- *Used Templates:*
  - dpm:CompoundExample: **1**
  - ottr:Triple: **4**

Generated Triples: *(Needs sometimes 2x refreshes)*

	Subject	Predicate	Object
Fe48Rh52	Fe48Rh52	Emmo:hasPart	Ottr:blank:Fe48Rh52 0 BN-b1
Fe48Rh52	Ottr:blank:Fe48Rh52 0 BN-b2	Rdfs:type	Rh
Fe48Rh52	Fe48Rh52	Emmo:hasPart	Ottr:blank:Fe48Rh52 0 BN-b2
Fe48Rh52	Ottr:blank:Fe48Rh52 0 BN-b1	Rdfs:type	Fe

```
{{#ask:
 [[-Has subobject::+]] [[Predicate::Emmo:hasPart]] [[Subject::Dpm:Fe48Rh52]]
 |?Subject
 |?Predicate
 |?Object
 |format=table}}
```

	Subject	Predicate	Object
Fe48Rh52	Fe48Rh52	Emmo:hasPart	Ottr:blank:Fe48Rh52 0 BN-b1
Fe48Rh52	Fe48Rh52	Emmo:hasPart	Ottr:blank:Fe48Rh52 0 BN-b2

UNIVERSITÄT BIELEFELD
Faculty of Technology

Conclusion & Open Questions

- benefits from already existing structured data
- improved communication with domain experts and improved understanding by domain experts
- workflows or naming conventions are important to relate templates to each other
- SMW as a central platform for the collaboration

Open for discussion:

- How to ensure **correctness of the ontology**?
- How to manage **versioning** as changes to template signatures require changes inside the templates and a revision of the relations this template is involved in? If instances are created manually, such changes are work intensive.
- How to design templates if the **structure** of the modelled processes is **not yet fully explored**?