FEMS EUROMAT 23

Easy access to smart materials data and models using an ontology based data and model access approach

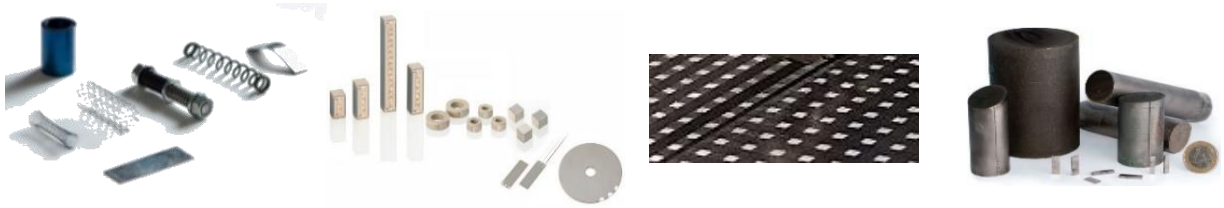
Jana Mertens
Technische Universität Berlin
Mechatronic Systems Lab
Jana.Mertens@tu-berlin.de
+49 (0) 30 314 21045

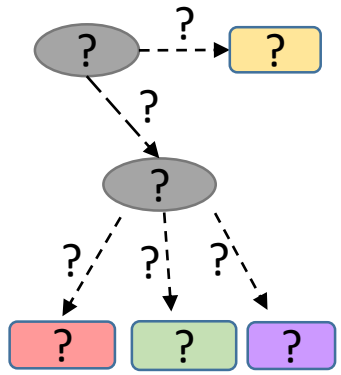
Sma*Di*
smart materials digital

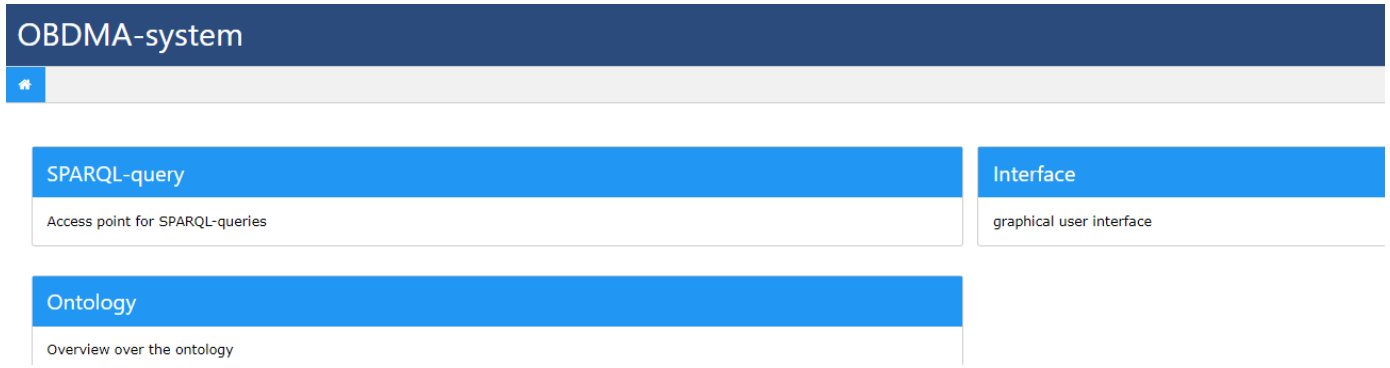
BMBF
supported academic joint project

Content

- Introduction
  - Smart materials
  - Ontology
- Working principle of the OBDMA system
- OBDMA system demonstrator
- Conclusion







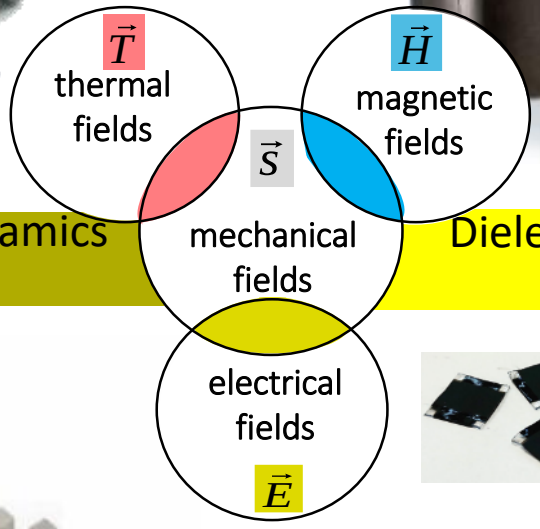
Introduction - Smart materials

Shape memory alloy
SMA

Magnetic shape memory alloy
MSMA

© Ingpuls

© ETO

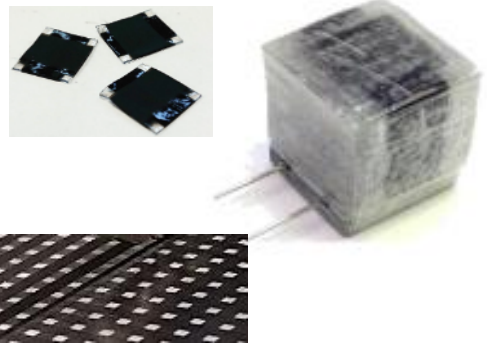


Piezoelectric ceramics
PC

Dielectric elastomer
DE

© PI Ceramic

© TUB



Project aim:

Easy, cross-scale data and model access for the four subclasses of smart materials : SMA, MSM, PC, DE

Solution: Ontology-based query answering system with integrated model access

Project partner:

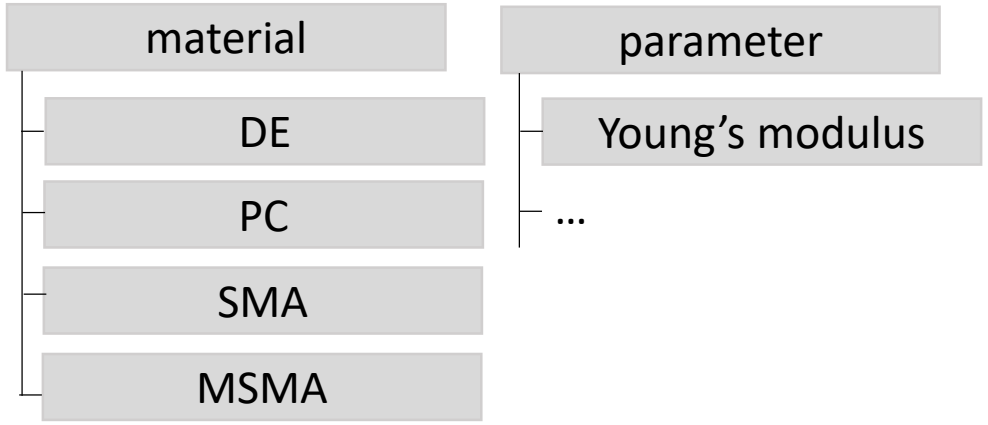
	material characterization and processing	material models up to part level
digitalization[l]	Universität zu Lübeck (UzL)	
SMA[a]	TU Chemnitz (TUC)	Fraunhofer IWU
MSMA[b]	Fraunhofer IWU	TU Berlin (TUB)
PC[c]	Fraunhofer IKTS	TU Ilmenau (TUI)
DE[d]	Fraunhofer IAP	TU Berlin (TUB)
	subclass sponsors: [a]Ingpuls, [b]ETO MAGNETIC, [c]PI Ceramic, [d]Wacker	**joint sponsor**: FESTO

Introduction - Ontology

Ontology

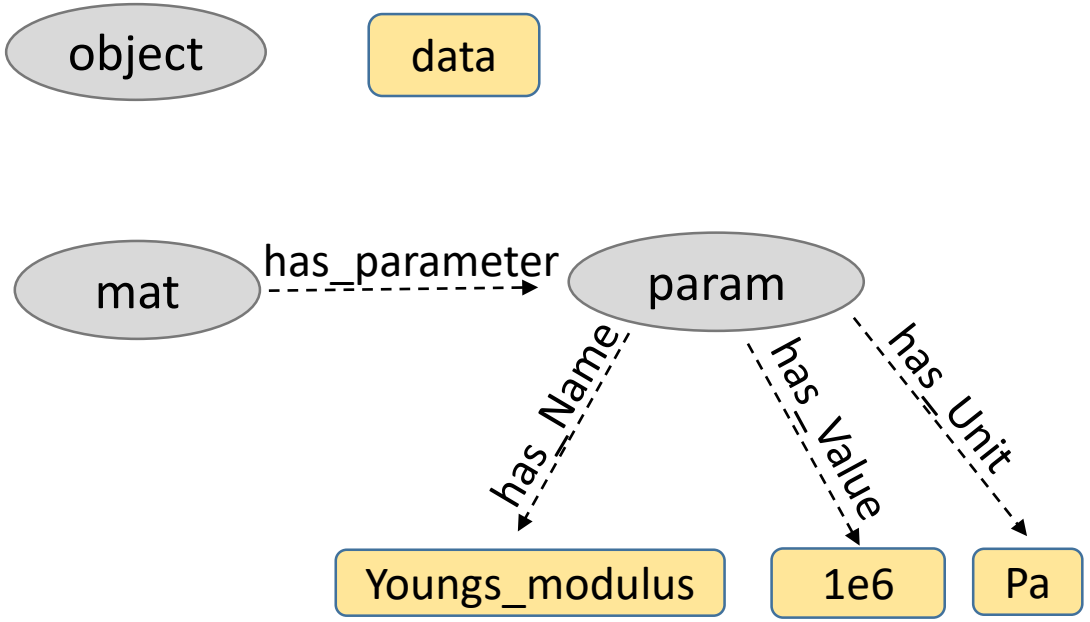
Terminological Box (TBox)

Predefined vocabulary in form of classes and relations.



Assertion Box (Abox)

Facts associated with TBox



Object relations:

is_a

has_parameter

Data relations:

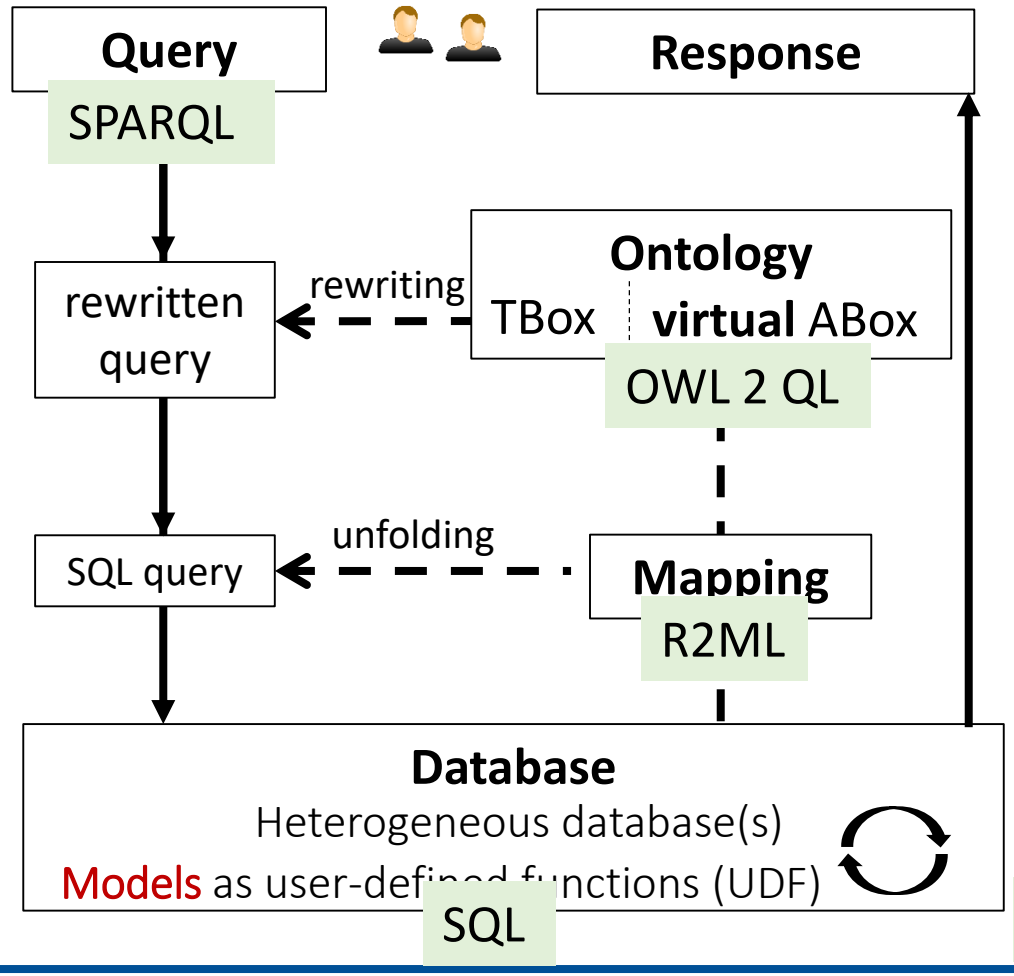
has_name

has_values

has_unit

Introduction - OBDMA

What is an ontology based data and **model** access (OBDMA)?



Main Task of the Ontology (Tbox)

- Predefine vocabulary (class names and relations)
- Define class hierarchy

Main Task of Rewriting

- Include TBox-knowledge in query

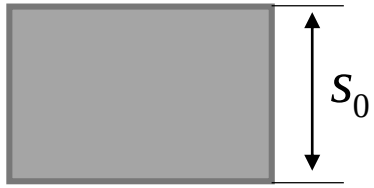
Main Task of Mapping

- Translation between TBox and Database
- Include additional knowledge, e.g. parameter can be calculated

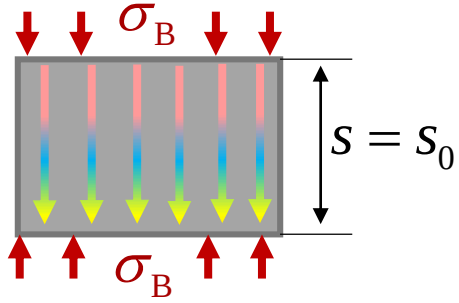
Use Case: maximum blocking stress

Blocking stress σ_B:

Actuator in initial situation



Actuator with physical field



SMA:
Limited by material fatigue

$\sigma_{B\text{-}max}$ **stored** in database

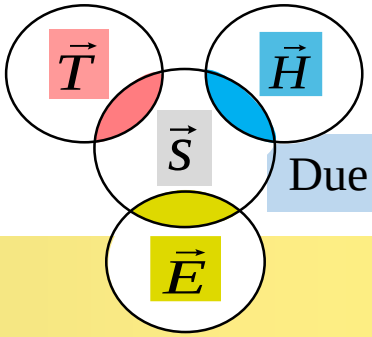
symbol	value	unit
$\sigma_{B,max}$	$500e^6$	Pa

MSMA:
Limited by Saturation

$\sigma_{B\text{-}max}$ is **calculated,**

$$\sigma_{B\text{-}max,h} = \sigma_{mag,max} + \sigma_{tw}$$

$$\sigma_{B\text{-}max,l} = \sigma_{mag,max} - \sigma_{tw}$$



Due two Hysteresis two values are required

PC

DE

Limited by electrical breaking field strenght E_{BFS}

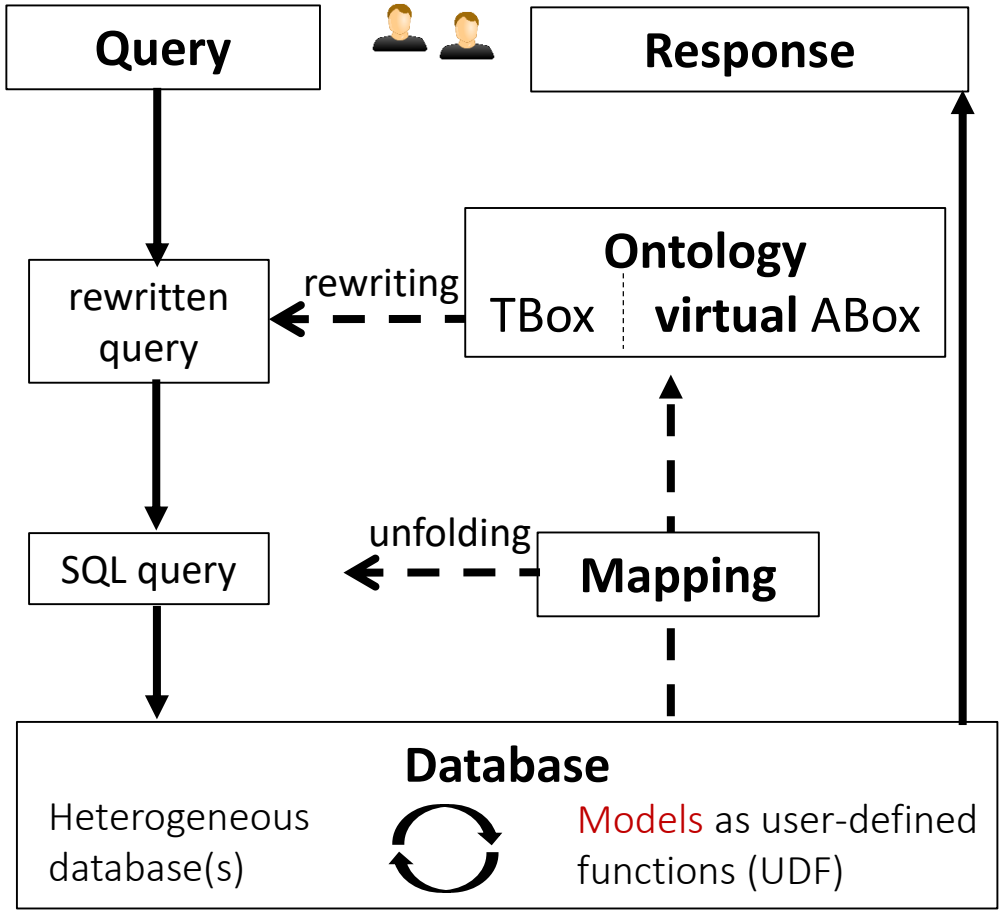
$\sigma_{B\text{-}max}$ is **calculated**

$$\sigma_{B\text{-}max} = \frac{d_{33,1}}{s^E_{33,1}} E_{BFS}$$

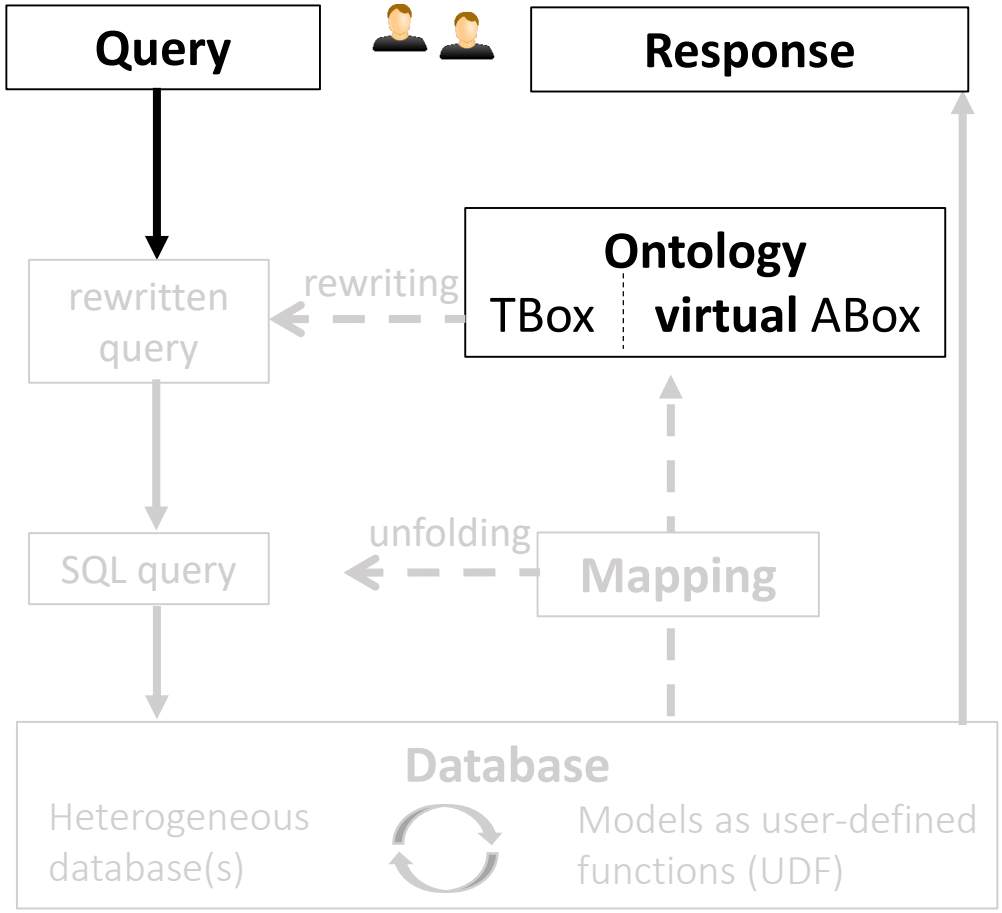
$\sigma_{B\text{-}max}$ is **calculated**

$$\sigma_{B\text{-}max} = \varepsilon_0 \varepsilon_r E_{BFS}^2$$

OBDMA system



OBDMA system



OBDMA system

Query

natural:

Give me the max. blocking stress with information about the material name, parameter symbol, value, and unit .

SPARQL

```
SELECT ?Mat ?Psym ?Value ?Unit
WHERE
{?param a :max_blocking_stress;
        :has_symbol ?Psym;
        :has_value ?Value;
        :has_unit ?Unit.
 ?m     :has_parameter ?param;
        :has_name ?Mat.}
```

Response

Mat	Psym	Value	Unit
?	?	?	?

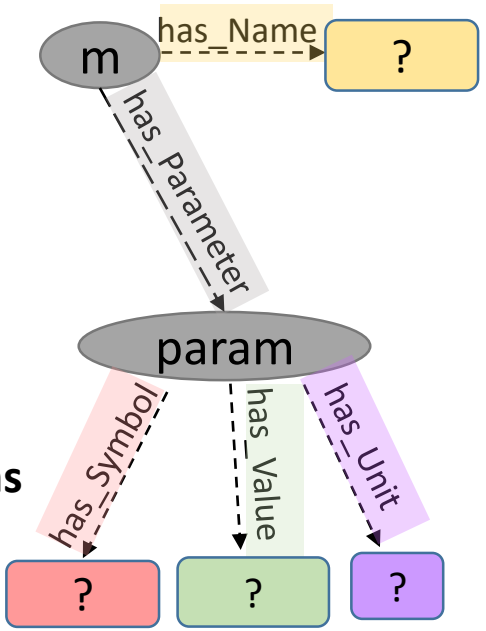
Ontology

TBox

Exemplary class hierarchy

- parameter
  - max_blocking_stress
    - max_blocking_stress_hold
    - max_blocking_stress_load

***Virtual** ABox*



Exemplary object and data relations

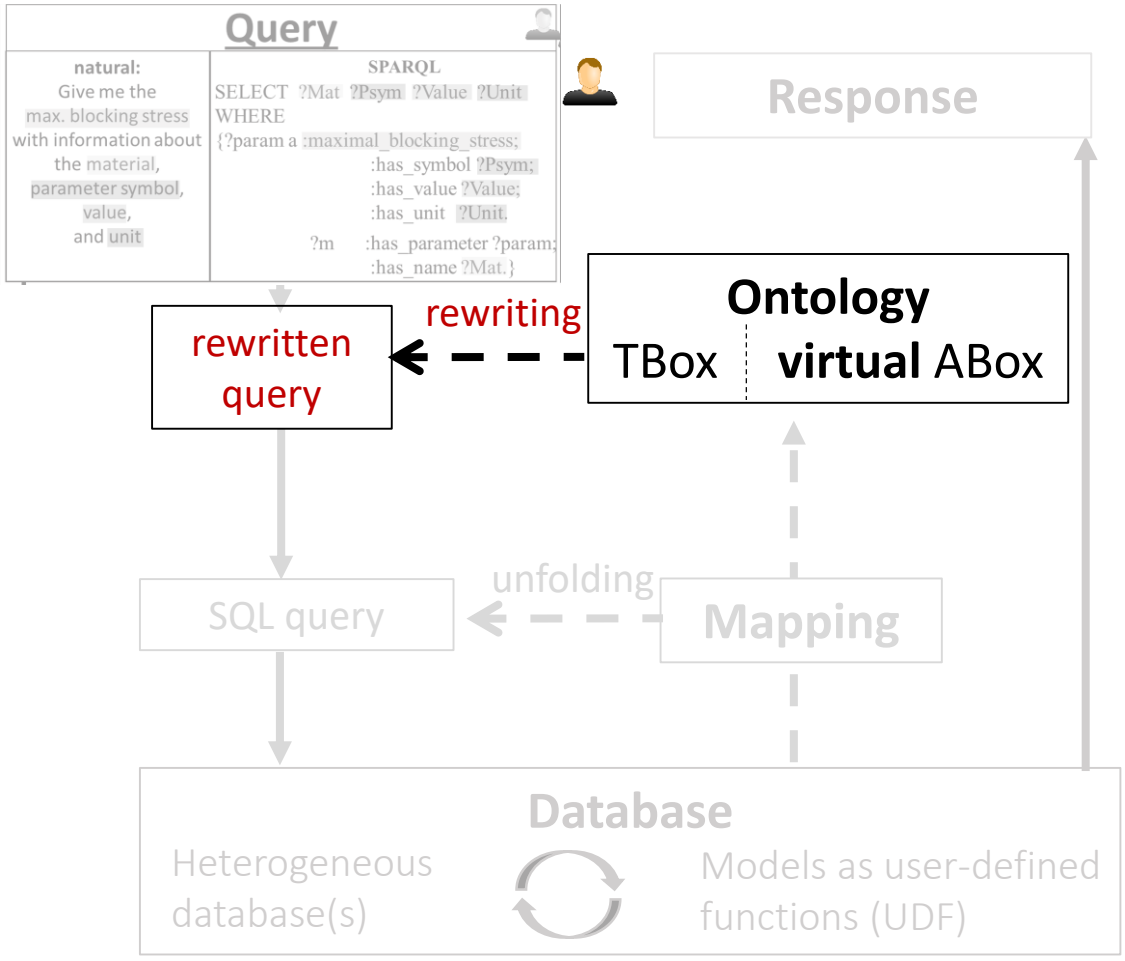
- has_Parameter
  - has_symbol
  - has_value
  - has_unit
  - has_name

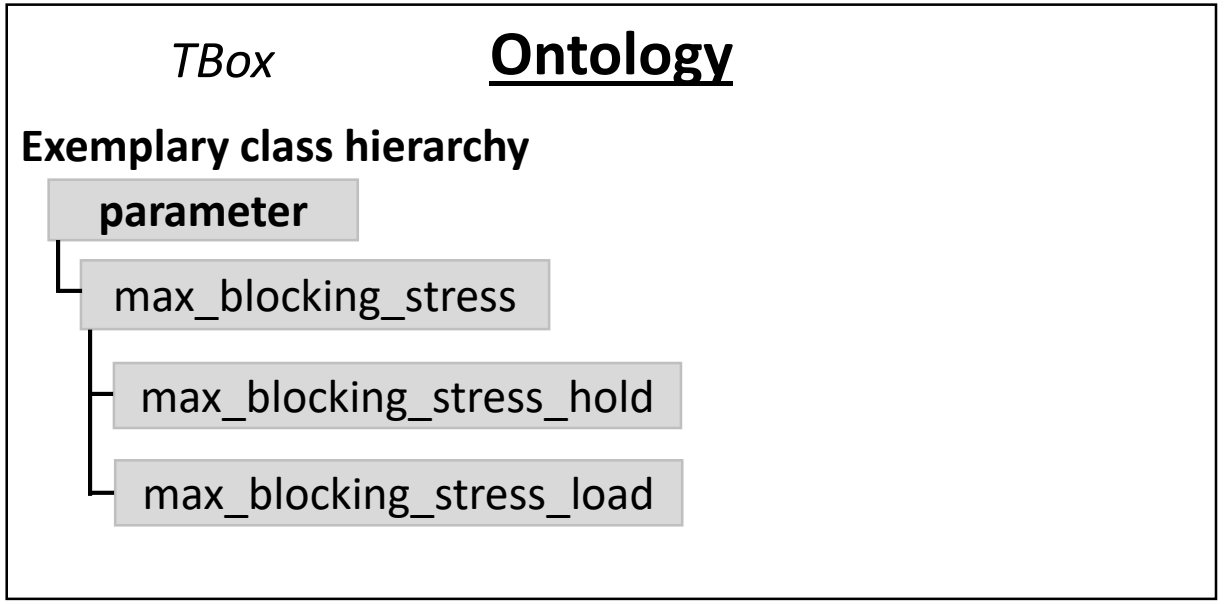
Database

Heterogeneous database(s)

Models as user-defined functions (UDF)

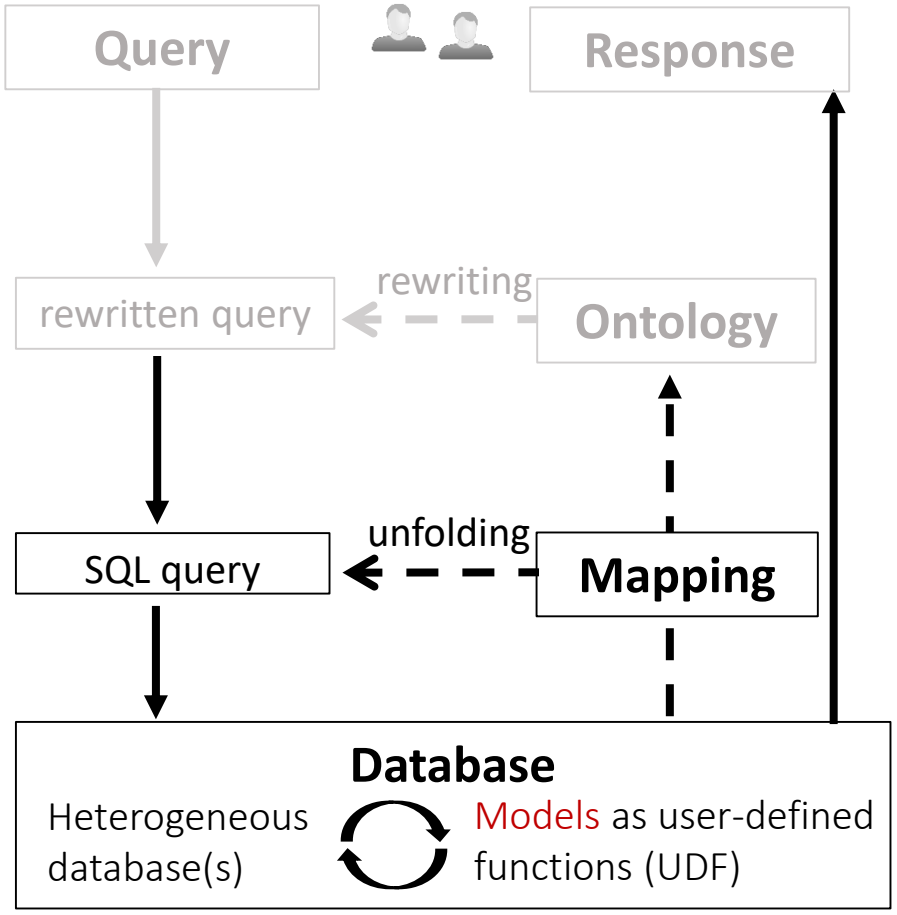
OBDMA system





→ **Rewritten Query:** Search for the **max_blockings_stress**, **max_blocking_stress_hold** and **max_blocking_Stress_load**.

OBDMA system



OBDMA system

Mapping

Translation for fixed values:

Tbox	SQL
max_blocking_stress	Search in parameter table for **param_id = Bl_stress_max**

Translation for calculated values:

Tbox	SQL
max_blocking_stress	Search in model table for mod_id= M_BSMax_DE
	Search for model input in **parameter table** for: **param_id = Perm_rel, param_id = Perm_vac param_id = Diel_str**

Database extension

Parameter table:

material	param_id	value	symbol	unit
NiTi	**Bl_stress_max**	$500e^{6}$	$\sigma_{B,max}$	Pa
-	**Perm_vac**	$8.9e^{-12}$	ε_0	F/m
EL2030	**Perm_rel**	3.2	ε_r	-
EL2030	**Diel_str**	$99e^{6}$	E_{BFS}	Pa

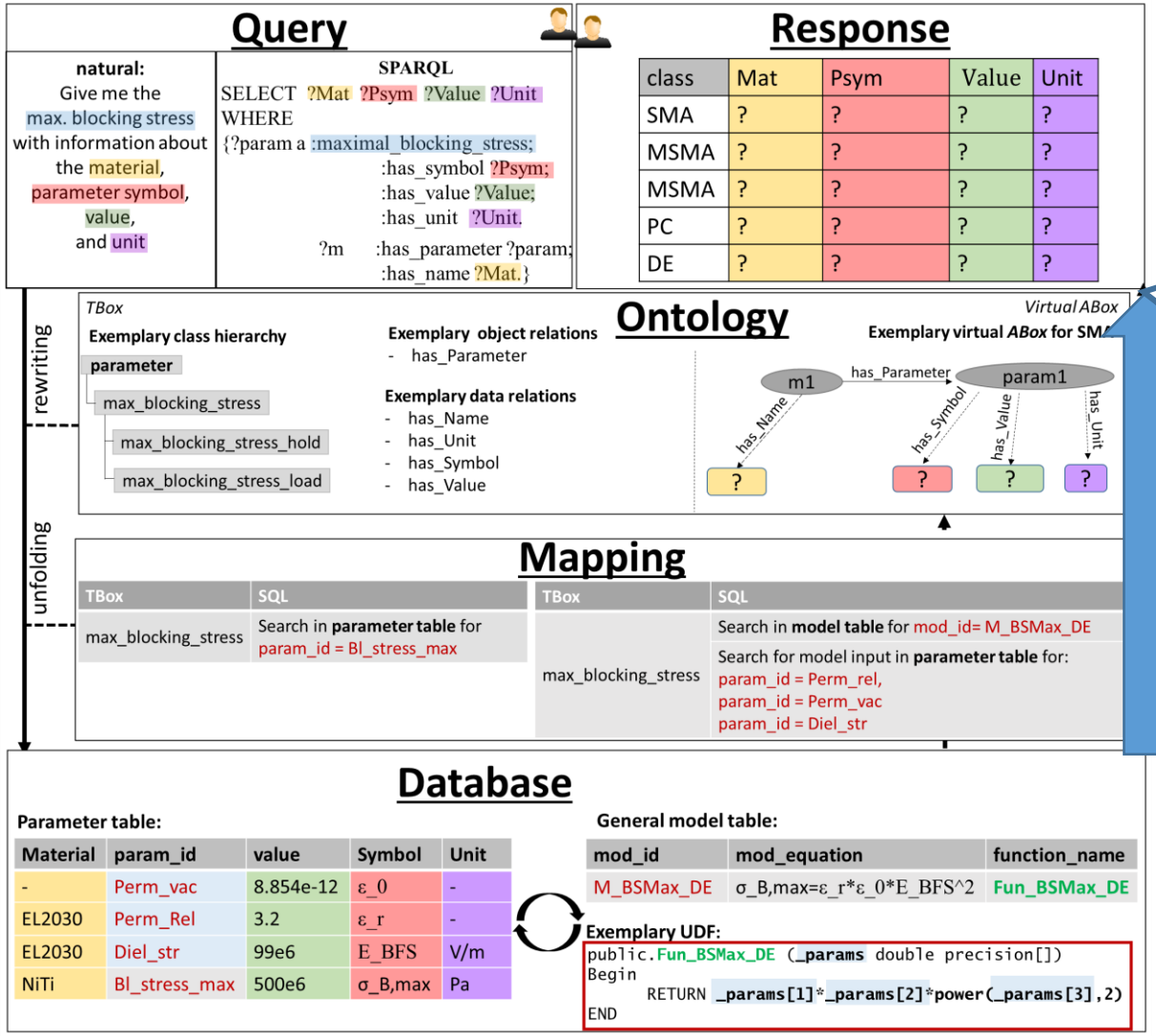
General model table:

mod_id	function_name
M_BSMax_DE	Fun_BSMax_DE

Exemplary UDF:

```
public.Fun_BSMax_DE (_params double precision[])
Begin
  RETURN _params[1]*_params[2]*power(_params[3],2)
END
```

OBDMA system



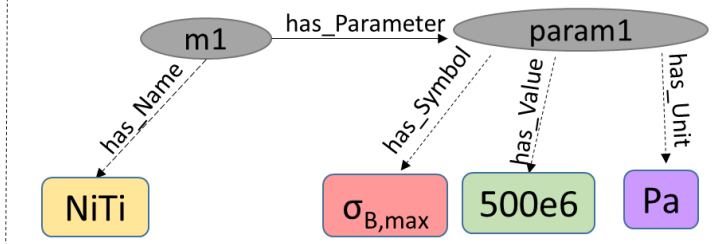
Response

class	Mat	Psym	Value	Unit
SMA	NiTi	σ_B,max	500e6	Pa
MSMA	stick	σ_B,max,l	1.6e6	Pa
MSMA	stick	σ_B,max,h	2.2e6	Pa
PC	M420	σ_B,max	1.9e6	Pa
DE	EL2030	σ_B,max	2.8e5	Pa

Ontology

Virtual ABox

Exemplary virtual *ABox* for SMA



Conclusion

Why we use OBD(M)A?

- Access to heterogeneous data bases
- Enables easy integration of workflows using user defined function „UDF“ in the database
  → Model access

System abilities

- Data and model based access to parameters and characteristic curves
- Exemplary implemented use cases with model access:
  - Interpolation of a specific characteristic value
  - Parameter identification
  - Parameter set conversion

Future works:

- Integration of more complex models, e.g. considering manufacturing processes
- Consideration of conditional queries (calculation with user-defined parameter values)

Thank you for your Attention!

GitHub - SmaDi-OBDMA/SmaDi-OBDMA-system

Plattform MaterialDigital