Foundations of accessing data through ontologies

Maria Keet

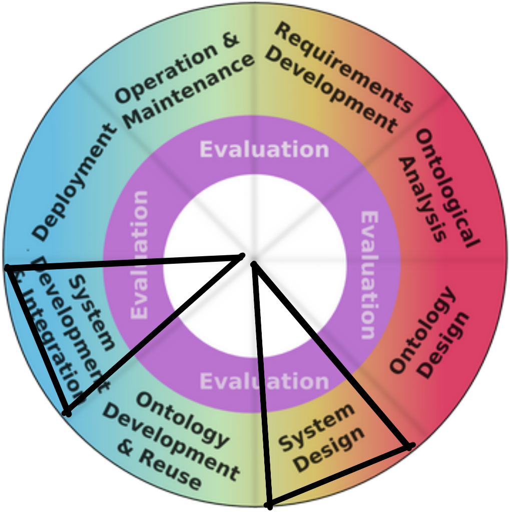
email: mkeet@cs.uct.ac.za

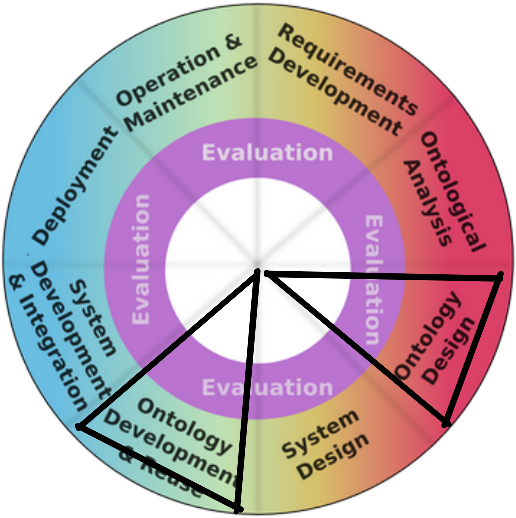
home: http://www.meteck.org

Department of Computer Science
University of Cape Town, South Africa

ISAO 2023, 10-14 July 2023, Sherbrooke, Canada

Outline


Requirements Development
Ontological Analysis
Ontology Design
System Design
Ontology Development & Reuse
System Development & Integration
Deployment
Operation & Maintenance
Evaluation
Evaluation
Evaluation
Evaluation


Requirements Development
Ontological Analysis
Ontology Design
System Design
Ontology Development & Reuse
System Development & Integration
Deployment
Operation & Maintenance
Evaluation
Evaluation
Evaluation
Evaluation

Outline

Why bother?

- Put the knowledge to use

- Add meaning to data

Why bother?

- Put the knowledge to use

⇒ For what use?

- Add meaning to data

⇒ What does that offer?

Why bother?

- Put the knowledge to use

$\Rightarrow$ For what use?

e.g.: data integration across database, RDBMS back-end OO frontend, ...

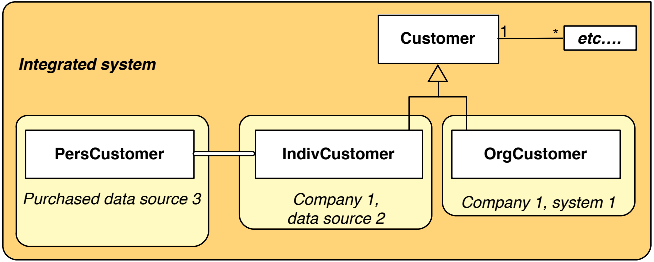
- Add meaning to data

$\Rightarrow$ What does that offer?

e.g.: easier and faster data access, infer more cf plain queries

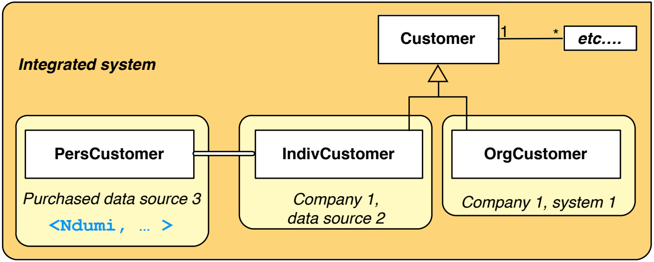
An example

- Subsumption & equivalence for integration



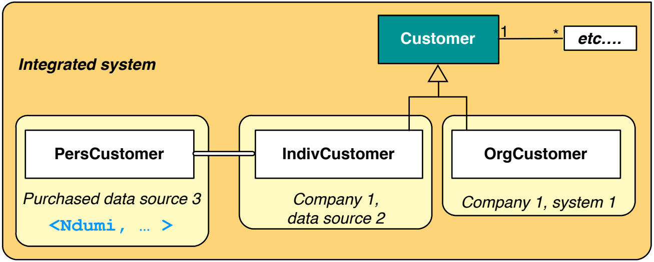
An example

- Subsumption & equivalence for integration
- Data, say `PersCustomer(Ndumi)`



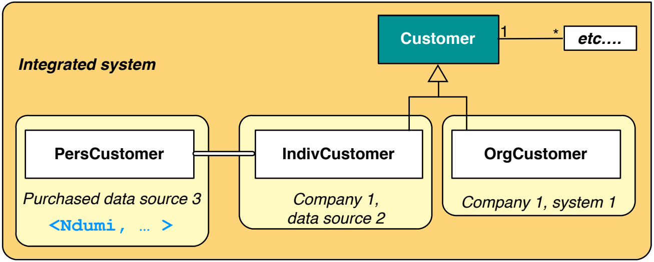
An example

- Subsumption & equivalence for integration
- Data, say `PersCustomer(Ndumi)`
- Query: "retrieve all customers"



An example

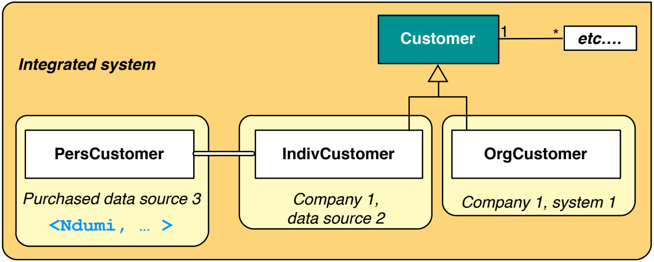
- Subsumption & equivalence for integration
- Data, say `PersCustomer(Ndumi)`
- Query: “retrieve all customers”
- Plain DB answer: {}



An example

- Subsumption & equivalence for integration
- Data, say `PersCustomer(Ndumi)`
- Query: “retrieve all customers”
- Plain DB answer: {}

⇒ Ontology-enhanced DB system: {`Ndumi`}

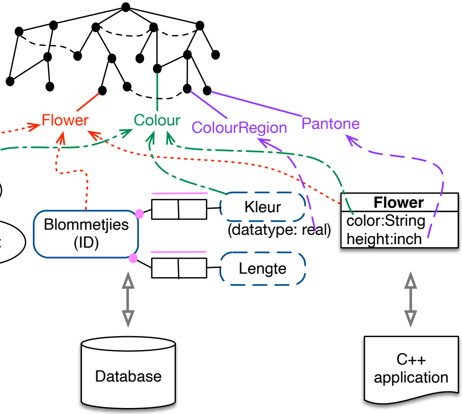


Connecting the knowledge to the data

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Conceptual model
shows what is stored in that particular application

Implementation
the actual information system that stores and manipulates the data

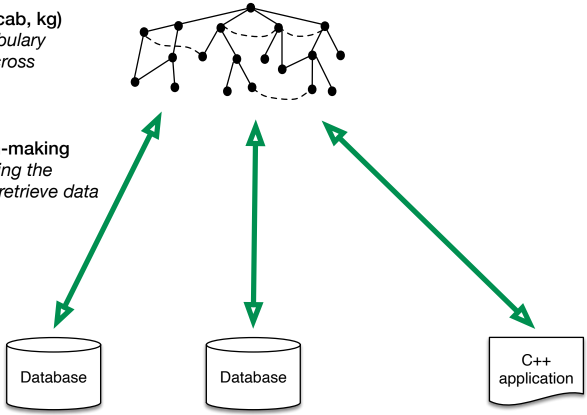


Connecting the knowledge to the data

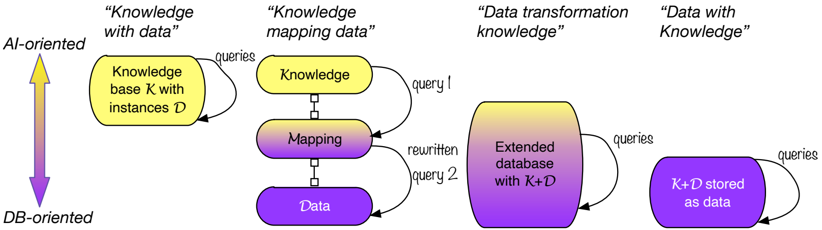
Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Queries for decision-making
formulate queries using the knowledge graph to retrieve data

Implementation
the actual information system that stores and manipulates the data

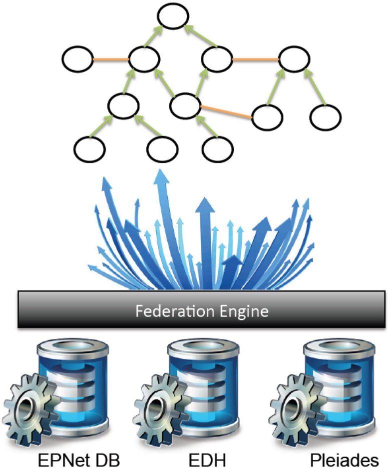


Knowledge-to-Data Pipeline options



Fillottrani, P.R., Keet, C.M. KnowID: An architecture for efficient Knowledge-driven Information and Data access. Data Intelligence, 2020, 2(4): 487-512.

"Knowledge mapping data": OBDA system EPnet [Calvanese et al.(2016)Calvanese, Liuzzo, Mosca, Remesal, R



Ontology or logic-based conceptual data model

Linking elements from the ontology to queries over the data source(s)

Mappings

The federation engine operates at the physical or relational schema layer

Data sources

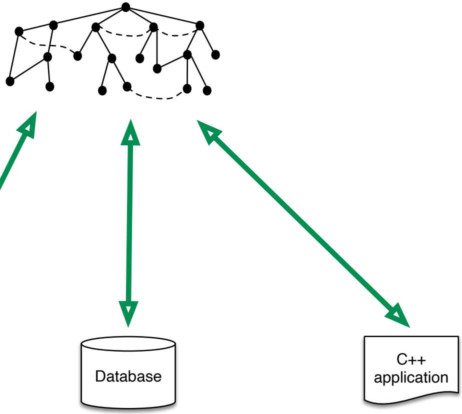
Typically relational databases and RDF triple stores

“Knowledge mapping data”: OBDA

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Queries for decision-making
formulate queries using the knowledge graph to retrieve data

Implementation
the actual information system that stores and manipulates the data

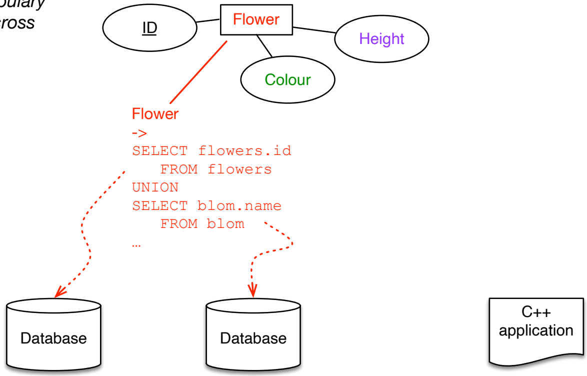


“Knowledge mapping data”: OBDA

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Mapping layer
links each entity to a query over the data source(s)

Implementation
the actual information system that stores and manipulates the data

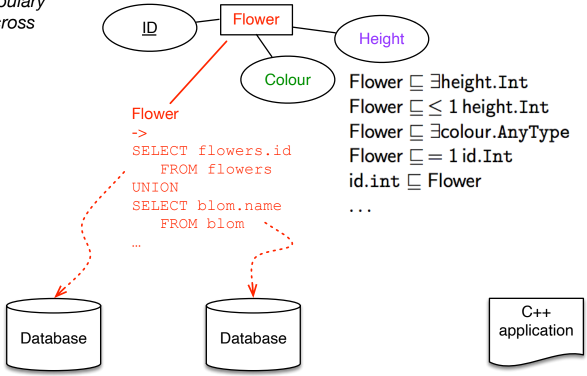


“Knowledge mapping data”: OBDA

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Mapping layer
links each entity to a query over the data source(s)

Implementation
the actual information system that stores and manipulates the data



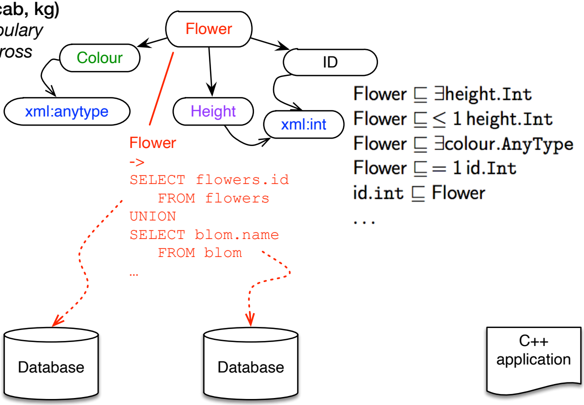
$\text{Flower} \sqsubseteq \exists \texttt{height.Int}$
$\text{Flower} \sqsubseteq \leq 1\, \texttt{height.Int}$
$\text{Flower} \sqsubseteq \exists \texttt{colour.AnyType}$
$\text{Flower} \sqsubseteq = 1\, \texttt{id.Int}$
$\texttt{id.int} \sqsubseteq \text{Flower}$
$\ldots$

"Knowledge mapping data": OBDA

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Mapping layer
links each entity to a query over the data source(s)

Implementation
the actual information system that stores and manipulates the data



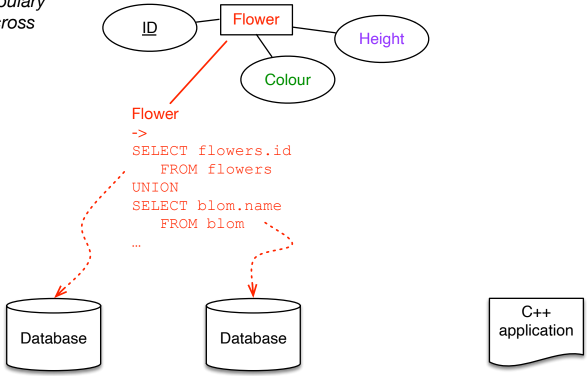
Flower $\sqsubseteq \exists$height.Int
Flower $\sqsubseteq \leq 1$ height.Int
Flower $\sqsubseteq \exists$colour.AnyType
Flower $\sqsubseteq = 1$ id.Int
id.int $\sqsubseteq$ Flower
...

"Knowledge mapping data": OBDA

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Mapping layer
links each entity to a query over the data source(s)

Implementation
the actual information system that stores and manipulates the data



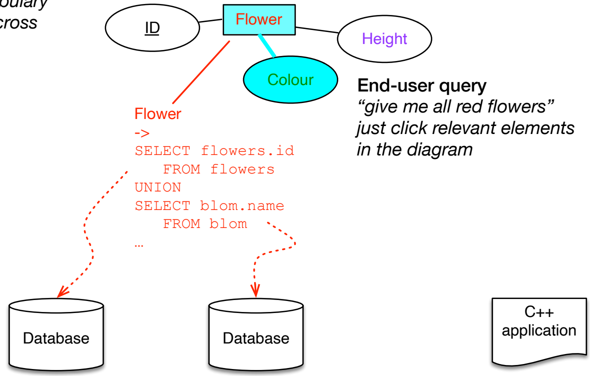
"Knowledge mapping data": OBDA

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Mapping layer
links each entity to a query over the data source(s)

Implementation
the actual information system that stores and manipulates the data

End-user query
"give me all red flowers" just click relevant elements in the diagram



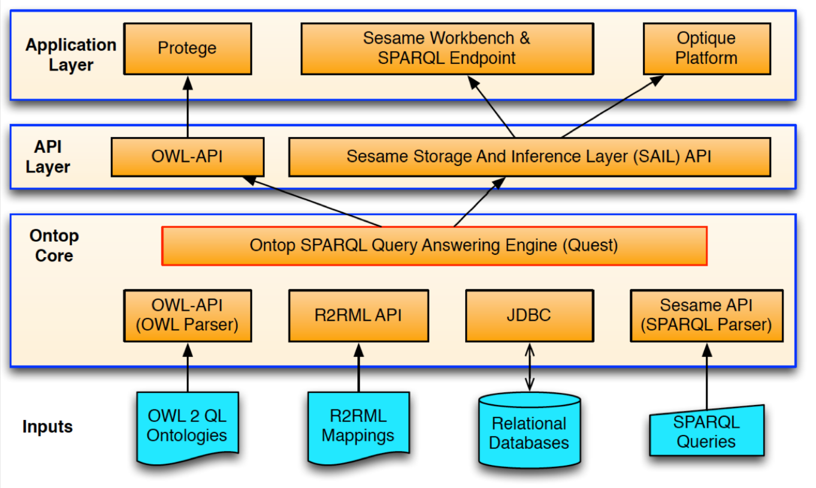
"Knowledge mapping data": OBDA

- OBDA with Ontop
  [Calvanese et al.(2017)Calvanese, Cogrel, Komla-Ebri, Kontchakov, L
  now more elaborate and more robust
- More recent case studies: Statoil, EPnet
  [Calvanese et al.(2016)Calvanese, Liuzzo, Mosca, Remesal, Rezk, and
  (early attempts: e.g., Genomics data with horizontal gene transfer
  [Calvanese et al.(2010)Calvanese, Keet, Nutt, Rodríguez-Muro, and

"Knowledge mapping data": OBDA

- OBDA with Ontop
  [Calvanese et al.(2017)Calvanese, Cogrel, Komla-Ebri, Kontchakov, L
  now more elaborate and more robust
- More recent case studies: Statoil, EPnet
  [Calvanese et al.(2016)Calvanese, Liuzzo, Mosca, Remesal, Rezk, and
  (early attempts: e.g., Genomics data with horizontal gene transfer
  [Calvanese et al.(2010)Calvanese, Keet, Nutt, Rodríguez-Muro, and
- Downsides
  - The mapping layer: cumbersome construction and maintenance
  - Low expressiveness for ontology language
  - Limitations on types of queries

An OBDA system with Ontop [Calvanese et al.(2017)Calvanese, Cogrel, Komla-Ebri, Kontc

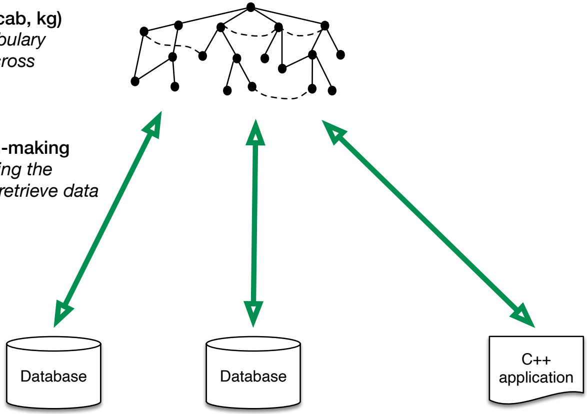


"Data-transformation-knowledge" example: KnowID

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Queries for decision-making
formulate queries using the knowledge graph to retrieve data

Implementation
the actual information system that stores and manipulates the data

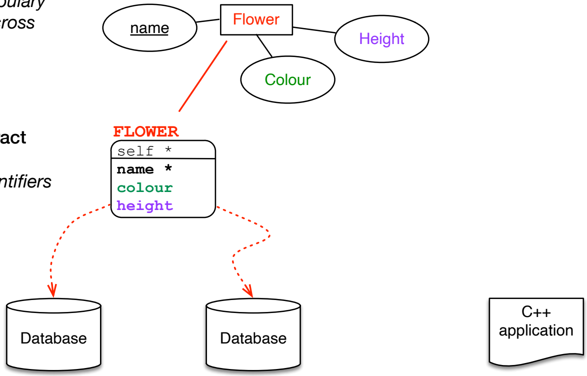


“Data-transformation-knowledge” example: KnowID

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Transformation via abstract relational model
with additional virtual *identifiers*

Implementation
the actual information system that stores and manipulates the data

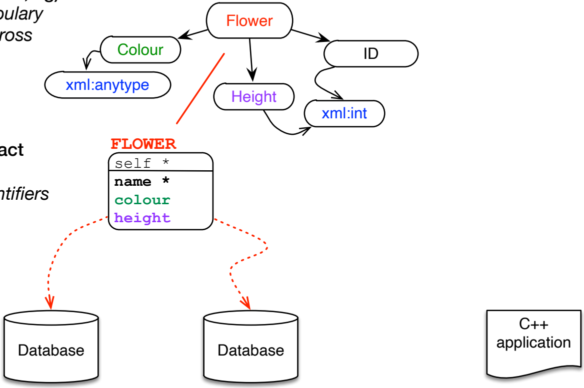


"Data-transformation-knowledge" example: KnowID

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

Transformation via abstract relational model
with additional virtual *identifiers*

Implementation
the actual information system that stores and manipulates the data

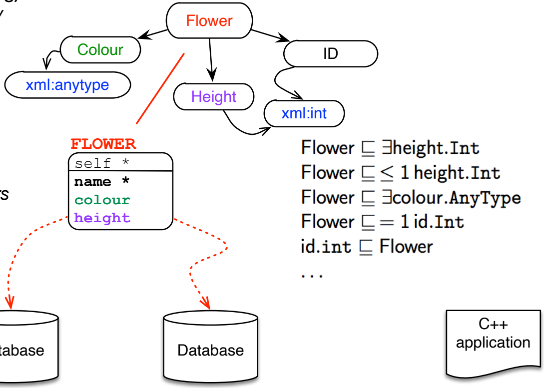


"Data-transformation-knowledge" example: KnowID

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

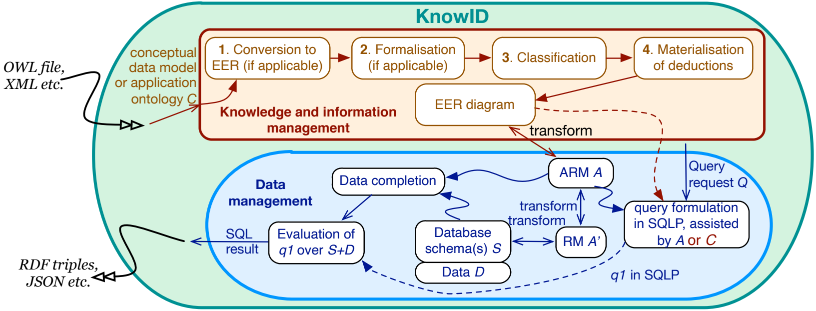
Transformation via abstract relational model
with additional virtual *identifiers*

Implementation
the actual information system that stores and manipulates the data



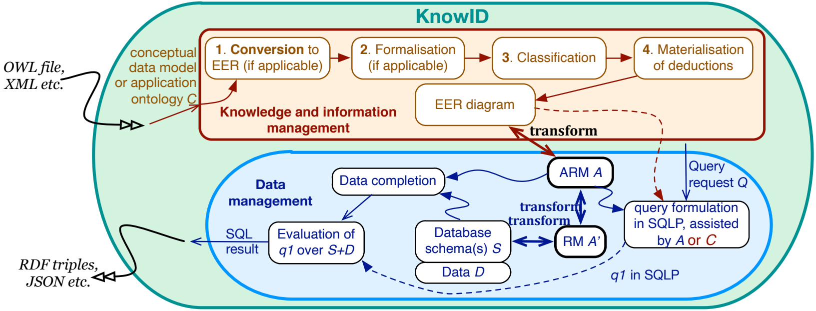
$\mathsf{Flower} \sqsubseteq \exists \mathtt{height}.\mathtt{Int}$
$\mathsf{Flower} \sqsubseteq\, \leq 1\, \mathtt{height}.\mathtt{Int}$
$\mathsf{Flower} \sqsubseteq \exists \mathtt{colour}.\mathtt{AnyType}$
$\mathsf{Flower} \sqsubseteq\, = 1\, \mathtt{id}.\mathtt{Int}$
$\mathtt{id}.\mathtt{int} \sqsubseteq \mathsf{Flower}$
$\ldots$

Knowledge-driven Information and Data access (KnowID)



Fillottrani, P.R., Keet, C.M. KnowID: An architecture for efficient Knowledge-driven Information and Data access. Data Intelligence, 2020, 2(4): 487-512.
Fillottrani, P.R., Jamieson, S., Keet, C.M. Connecting knowledge to data through transformations in KnowID: system description. Künstliche Intelligenz, 2020, 2020, 34, 373-379.

Knowledge-driven Information and Data access (KnowID)



Fillottrani, P.R., Keet, C.M. KnowID: An architecture for efficient Knowledge-driven Information and Data access. Data Intelligence, 2020, 2(4): 487-512.
Fillottrani, P.R., Jamieson, S., Keet, C.M. Connecting knowledge to data through transformations in KnowID: system description. Künstliche Intelligenz, 2020, 2020, 34, 373-379.

Knowledge-driven Information and Data access (KnowID)

- There's more on the 'knowledge and information management' module [Fillottrani and Keet(2020)]:
  - Swap between EER, UML, ORM [Keet and Fillottrani(2015), Fillottrani and Keet(2014), Braun et al.(2023)Braun, Fillottrani, and Keet]
  - DL (OWL) with reasoner at the back-end
  - Tool: `crowd 2.0` (beta) `http://crowd.fi.uncoma.edu.ar:3335/` [Braun et al.(2020)Braun, Gimenez, Cecchi, and Fillottrani]
  - More in the pipeline, such as integrating NOMSA for summarisation and modularisation of ontologies
- Querying with SQLP: SQLP requires less time for understanding and authoring queries, with no loss in accuracy [Ma et al.(2018)Ma, Keet, Oldford, Toman, and Weddell]
- Data Completion TBD

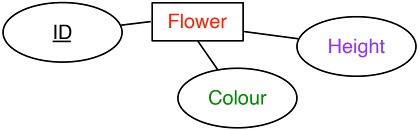
Outline

Three key factors for choosing an approach

- **Stability of the data:** do they (i) continuously change a lot throughout the database or (ii) intermittently, rarely, or append-only?
- **Stability of the schemas:** do they (i) remain unchanged once the system is set-up or (ii) will they have to change based on changing business needs and usage optimisations?
- **Type of queries posed over the data:** are they (i) at most (unions of) conjunctive queries (UCQs) or (ii) also other types of SQL queries (with or without paths)?

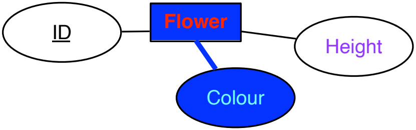
Queries with OBDA models vs FO-inspired ontologies

Ontology (or controlled vocab, kg) *provides the common vocabulary and constraints that hold across the applications*



Queries with OBDA models vs FO-inspired ontologies

Ontology (or controlled vocab, kg)
provides the common vocabulary and constraints that hold across the applications

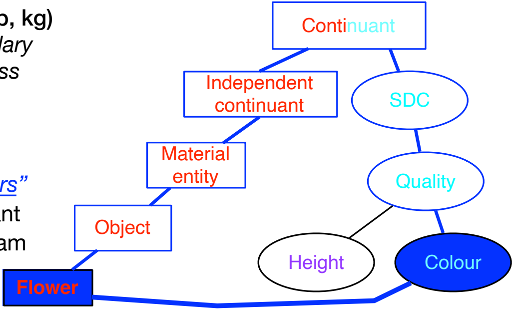


End-user query
"give me all red flowers"
just click relevant elements in the diagram

Queries with OBDA models vs FO-inspired ontologies

Ontology (or controlled vocab, kg) *provides the common vocabulary and constraints that hold across the applications*

End-user query
"give me all red flowers"
"*just*" process relevant elements in the diagram



How to answer queries efficiently?

1. $(\mathcal{T}, \mathcal{A})$ have exactly one model $\mathcal{I}$: then $Q(\mathcal{A}, \mathcal{T}) = Q(\mathcal{I})$

slides 18, 19, 22 with thanks to David Toman

How to answer queries efficiently?

1. $(\mathcal{T}, \mathcal{A})$ have exactly one model $\mathcal{I}$: then $Q(\mathcal{A}, \mathcal{T}) = Q(\mathcal{I})$

   . . . this is probably what you assume to be happening

slides 18, 19, 22 with thanks to David Toman

How to answer queries efficiently?

1. $(\mathcal{T}, \mathcal{A})$ have exactly one model $\mathcal{I}$: then $Q(\mathcal{A}, \mathcal{T}) = Q(\mathcal{I})$
   ... this is probably what you assume to be happening
2. $(\mathcal{T}, \mathcal{A})$ have many models, say $\mathcal{I}_j$ $(j \in J)$:
   Option I: restrict $\mathcal{T}$ to make it feasible: *(simple) Horn theories*

slides 18, 19, 22 with thanks to David Toman

How to answer queries efficiently?

1. $(\mathcal{T}, \mathcal{A})$ have exactly one model $\mathcal{I}$: then $Q(\mathcal{A}, \mathcal{T}) = Q(\mathcal{I})$

   ... this is probably what you assume to be happening

2. $(\mathcal{T}, \mathcal{A})$ have many models, say $\mathcal{I}_j \quad (j \in J)$:

   Option I: restrict $\mathcal{T}$ to make it feasible: *(simple) Horn theories*

   $\Rightarrow$ canonical models (and small ones!)

   $\Rightarrow$ works well only for positive queries

slides 18, 19, 22 with thanks to David Toman

How to answer queries efficiently?

1. $(\mathcal{T}, \mathcal{A})$ have exactly one model $\mathcal{I}$: then $Q(\mathcal{A}, \mathcal{T}) = Q(\mathcal{I})$
   ... this is probably what you assume to be happening
2. $(\mathcal{T}, \mathcal{A})$ have many models, say $\mathcal{I}_j \quad (j \in J)$:
   - Option I: restrict $\mathcal{T}$ to make it feasible: *(simple) Horn theories*
     - $\Rightarrow$ canonical models (and small ones!)
     - $\Rightarrow$ works well only for positive queries
   - Option II: restrict Q to make it feasible: those for which it doesn't matter which model is used

slides 18, 19, 22 with thanks to David Toman

How to answer queries efficiently?

1. $(\mathcal{T}, \mathcal{A})$ have exactly one model $\mathcal{I}$: then $Q(\mathcal{A}, \mathcal{T}) = Q(\mathcal{I})$

   ... this is probably what you assume to be happening

2. $(\mathcal{T}, \mathcal{A})$ have many models, say $\mathcal{I}_j$ $(j \in J)$:

   Option I: restrict $\mathcal{T}$ to make it feasible: *(simple) Horn theories*

   $\Rightarrow$ canonical models (and small ones!)

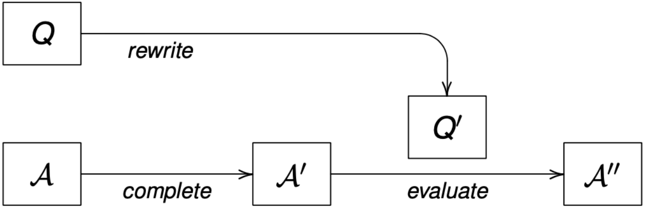
   $\Rightarrow$ works well only for positive queries

   Option II: restrict Q to make it feasible: those for which it doesn't matter which model is used

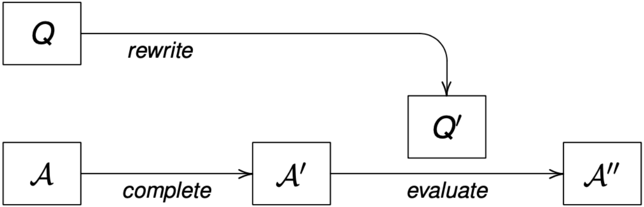
   $\Rightarrow$ e.g., safe queries in Codd's relational model

slides 18, 19, 22 with thanks to David Toman

Option I



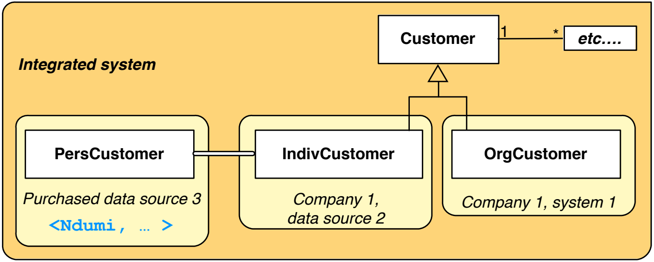
Option I



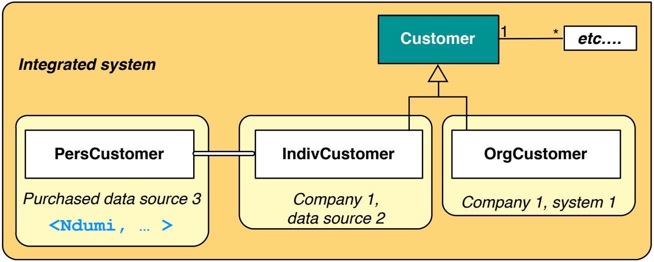
v1.0: *rewrite*: incorporate $\mathcal{T}$ into Q,
complete: an identity ($\mathcal{A}' = \mathcal{A}$)
... [Calvanese et al.]

v2.0: *rewrite*: rewrite independently of $\mathcal{T} \cup \mathcal{A}$,
complete: incorporate $\mathcal{T}$ into $\mathcal{A}$
... [Lutz et al.]

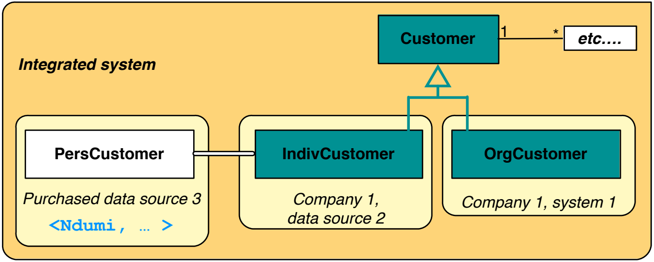
An example – Revisited with query rewriting



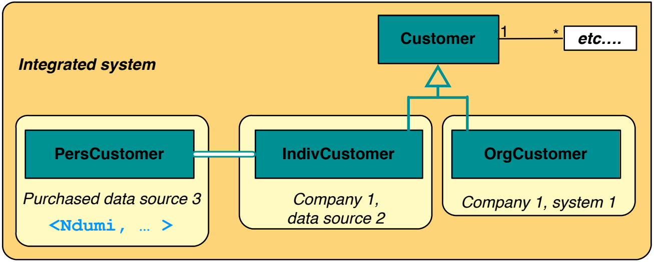
An example – Revisited with query rewriting



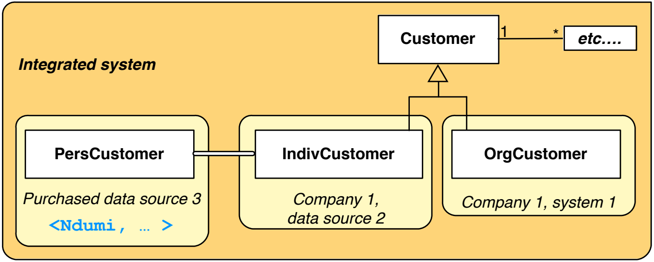
An example – Revisited with query rewriting



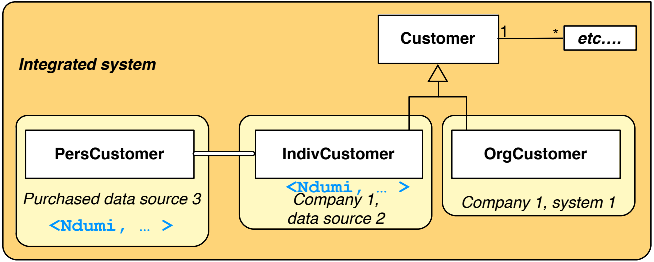
An example – Revisited with query rewriting



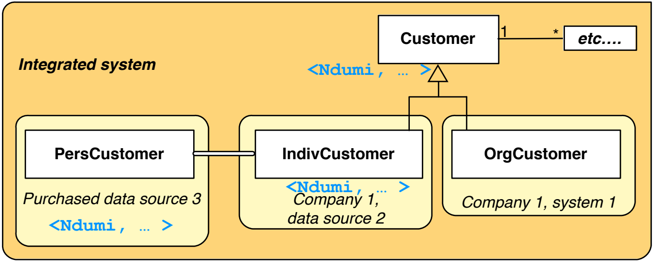
An example – Revisited with data completion



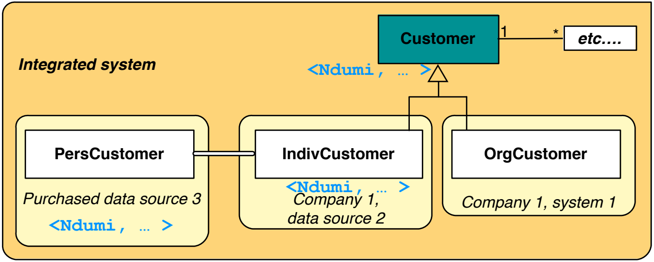
An example – Revisited with data completion



An example – Revisited with data completion



An example – Revisited with data completion



One more note on rewriting vs data completion

	v1.0 (query rewriting)	v2.0 (data completion)
Queries	rewriting is exponential in $\|Q\|$	data only grows polynomially in $\|\mathcal{A}\|$
Updates	applies to original data	needs rematerialize data completion

we always can devise a way where one system wins over the other

Key distinguishing features of varying computational cost

Feature	K ⓐ D	$K \Leftrightarrow D$	$D \bowtie K$	D ⓐ K
World	OWA	OWA+CWA	CWA	CWA
Language for $\mathcal{K}$	OWL	OWL	relational, DL	relational
Language for $\mathcal{D}$	OWL	relational	relational	relational
Query language	SPARQL	SPARQL + SQL (fragment)	SQLP	SQL
Automated reasoning	yes	yes	yes	depends on system
Reasoning w.r.t. data	no separate approach	query rewriting	data completion	data completion
Mapping layer	no	yes	no	no
Transformations	no	no	yes	yes
Entity recasting	no	yes	no	yes
Syntactic sugar	available	available	possible	possible

Outline

Summary

1. Approaches
2. Digging deeper to compare
3. Knowledge mapping data with OBDA: A system
   - The mapping layer
   - Query answering
4. A 'simple' example

References I

G. Braun, P. R. Fillottrani, and C. M. Keet.
A framework for interoperability between models with hybrid tools.
Journal of Intelligent Information Systems, 60:437–462, 2023.

Germán Braun, Christian Gimenez, Laura Cecchi, and Pablo Fillottrani.
`crowd`: A Visual Tool for Involving Stakeholders into Ontology Engineering Tasks.
Künstliche Intelligenz, 2020.

D. Calvanese, P. Liuzzo, A. Mosca, J. Remesal, M. Rezk, and G. Rull.
Ontology-based data integration in epnet: Production and distribution of food during the roman empire.
Engineering Applications of Artificial Intelligence, 51:212–229, 2016.

Diego Calvanese, C. Maria Keet, Werner Nutt, Mariano Rodríguez-Muro, and Giorgio Stefanoni.
Web-based graphical querying of databases through an ontology: the WONDER system.
In Sung Y. Shin, Sascha Ossowski, Michael Schumacher, Mathew J. Palakal, and Chih-Cheng Hung, editors, *Proceedings of ACM Symposium on Applied Computing (ACM SAC'10)*, pages 1389–1396. ACM, 2010.
URL `http://www.meteck.org/files/SAC10wonderCKNRS.pdf`.
March 22-26 2010, Sierre, Switzerland.

Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao.
Ontop: Answering SPARQL queries over relational databases.
Semantic Web Journal, 8(3):471–487, 2017.

P. R. Fillottrani and C. M. Keet.
KnowID: An architecture for efficient knowledge-driven information and data access.
Data Intelligence, 2(4):487–512, 2020.

References II

Pablo Rubén Fillottrani and C. Maria Keet.
Conceptual model interoperability: a metamodel-driven approach.
In A. Bikakis et al., editors, *Proceedings of the 8th International Web Rule Symposium (RuleML'14)*, volume 8620 of *LNCS*, pages 52–66. Springer, 2014.
August 18-20, 2014, Prague, Czech Republic.

S. Garcia-Vallvé, E. Guzman, M.A. Montero, and A. Romeu.
HGT-DB: a database of putative horizontally transferred genes in prokaryotic complete genomes.
Nucleic Acids Research, 31(1):187–189, 2003.

C. Maria Keet and Pablo Rubén Fillottrani.
An ontology-driven unifying metamodel of UML Class Diagrams, EER, and ORM2.
Data & Knowledge Engineering, 98:30–53, 2015.
doi: 0.1016/j.datak.2015.07.004.

Weicong Ma, C. Maria Keet, Wayne Oldford, David Toman, and Grant Weddell.
The utility of the abstract relational model and attribute paths in sql.
In Catherine Faron Zucker, Chiara Ghidini, Amedeo Napoli, and Yannick Toussaint, editors, *Proceedings of the 21st International COnference on Knowledge Engineering and Knowledge Management (EKAW'18))*, volume 11313 of *LNAI*, pages 195–211. Springer, 2018.
ISBN 978-3-030-03667-6.
12-16 Nov. 2018, Nancy, France.

Thank you!

Questions?

- My textbook on ontology engineering (aimed at computer scientists)
- Free pdf (and slides and exercises) at `https://people.cs.uct.ac.za/~mkeet/OEbook/`
- Also available in paperback: