Patterns for Heterogeneous TBox Mappings to Bridge Different Modelling Decisions

Pablo R. Fillottrani[1,2] and C. Maria Keet[3]

[1] Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina, prf@cs.uns.edu.ar
[2] Comisión de Investigaciones Científicas, Provincia de Buenos Aires, Argentina
[3] Department of Computer Science, University of Cape Town, South Africa
mkeet@cs.uct.ac.za

14th International Extended Semantic Web Conference (ESWC'17)
Portoroz, Slovenia, 29 May - 1 June, 2017

Outline

Outline

Introduction

- Representing 'the same' knowledge in different ways
- Sometimes there's a right and a wrong way, but not always
- Typical (recurring) examples:
  - Should `Marriage`/`or married to` be a class or an object property?
  - `Colour` as a class or a data property?
  - `Employee` is a subclass of `Person` or does it `inhere in` a person?
- Different modelling decisions are made for different reasons; e.g., OBDA, aligning with a top-level ontology

Introduction

- Representing 'the same' knowledge in different ways
- Sometimes there's a right and a wrong way, but not always
- Typical (recurring) examples:
  - Should `Marriage`/`or married to` be a class or an object property?
  - `Colour` as a class or a data property?
  - `Employee` is a subclass of `Person` or does it `inhere in` a person?
- Different modelling decisions are made for different reasons; e.g., OBDA, aligning with a top-level ontology

$\Rightarrow$ **How to align these different modelling choices?**

Shortcomings

- Current mapping/alignment approaches mainly 1:1 [Euzenat and Shvaiko(2007), Otero-Cerdeira et al.(2015)]
- Here, we have to map small patterns

Shortcomings

- Current mapping/alignment approaches mainly 1:1 [Euzenat and Shvaiko(2007), Otero-Cerdeira et al.(2015)]
- Here, we have to map small patterns
- ODPs include "correspondence patterns" [Ritze et al.(2009), Scharffe and Fensel(2008), Scharffe et al.(2014)]
  - mostly *definition alignments* (e.g., $C \equiv \exists R.D$)
- No systematic way of aligning arbitrary lhs and arbitrary rhs of inclusion or equivalence

Aims

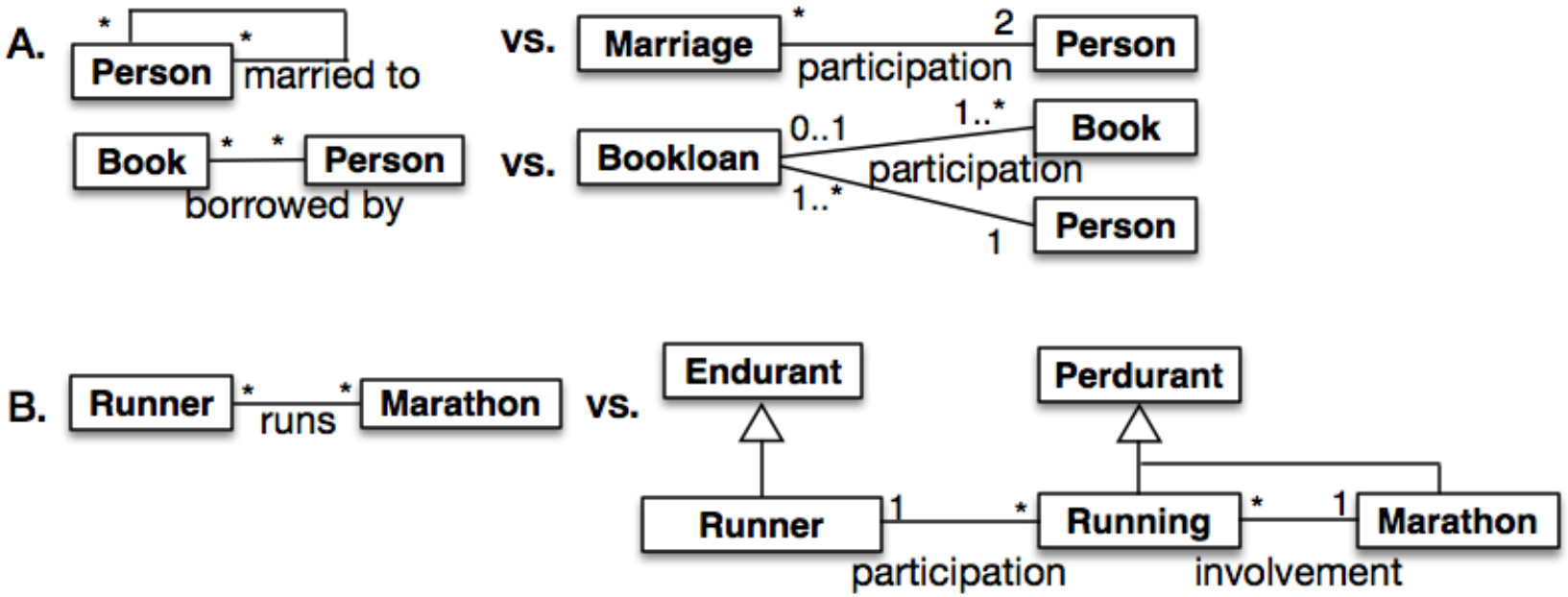
- Identify common modelling patterns
- Devise formalisation of OP and of OP alignment
- Formalise those common patterns and alignments
- Algorithm design for *automated finding* of the patterns and *checking correctness* of a possible pattern-based alignment

Outline

Introduction

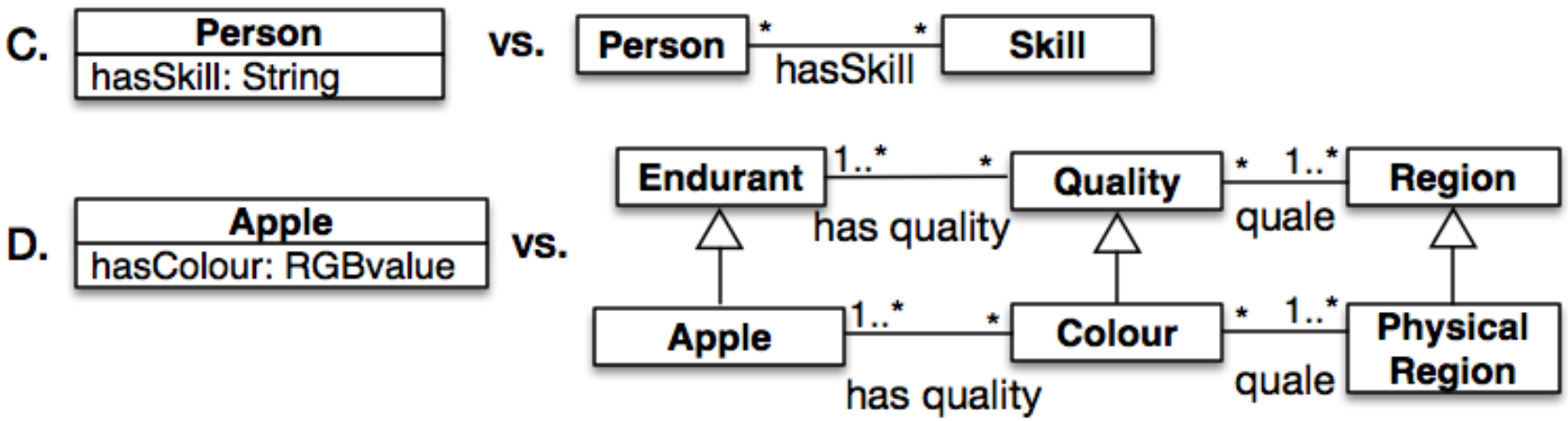
- 5 alternate modelling pattern pairs proposed in the paper
- 2 pattern pairs generalised from correspondence patterns
- Discuss only one in detail in this presentation: class vs object property

'Case A': class or object property? (and Case B with perdurants)



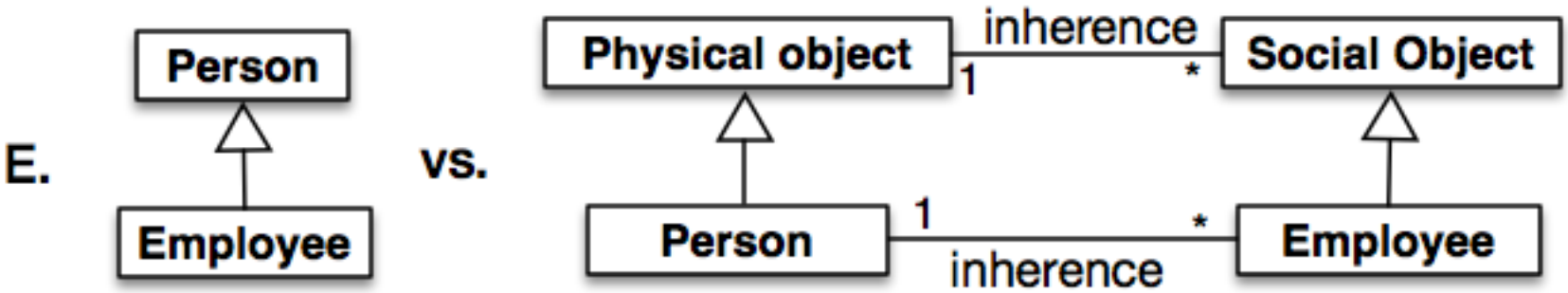
- Main issue: to reify or not to reify?
- And: are the more precise cardinality constraints needed?

Attributes or not?



- Class vs data property
- Object property vs data property (consequence of the former)
- Qualities vs data properties

Roles or subclasses?



Formalisation—preliminaries

- Language of pattern instantiation (OWL in this case)
- Language for patterns with vocabulary $\mathcal{V}$, meta-level (second-order) elements (or stereotypes)
- Ontology pattern, with name, elements from $\mathcal{V}$, pattern axiom components, pattern's full formalisations; e.g.:
  - pattern name: *basic all-some*
  - pattern elements: $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{R}_1$
  - pattern axiom components: $\sqsubseteq$, $\exists$
  - pattern's full formalisation $\mathcal{C}_1 \sqsubseteq \exists \mathcal{R}.\mathcal{C}_2$

  Example instantiation: $\texttt{Professor} \sqsubseteq \exists \texttt{teaches.Course}$
- Homogeneous mapping: subsumption or equivalence relating two homogeneous elements (e.g., `Teacher` in O and `Instructor` in O')

Definition (Ontology Pattern Alignment, *OPA*)

An *ontology pattern alignment OPA* consists of two ontology patterns, P and P', such that its signature Σ is a subset of the signature of the respective ontologies O and O', i.e., $\Sigma(P) \subseteq \Sigma(O)$ and $\Sigma(P') \subseteq \Sigma(O')$, and alignment axioms

- alignment pattern name;
- pattern elements;
- alignment patterns' context, consisting of:
  - O's pattern P
  - O''s pattern P'
- alignment pattern axiom component(s) from V_X;
- pattern alignment's formalisation, composed of:
  - a (possibly empty) set of mappings between homogeneous elements in P and P'
  - a set of axioms made from components in V_X connecting heterogeneous elements in P and P'

Class vs. Object Property (case A)

- *alignment pattern name*: *class-OP*
- *pattern elements*: $\mathcal{C}_1, \mathcal{C}_2, \mathcal{R}_1$ from O, $\mathcal{C}'_3, \mathcal{C}'_4, \mathcal{C}'_5, \mathcal{R}'_2, \mathcal{R}'_3$ from O'
- *alignment patterns' contexts*:
  - *pattern P in O*: $\exists \mathcal{R}_1.\mathcal{C}_2 \sqsubseteq \mathcal{C}_1$ and $\exists \mathcal{R}_1^-.\mathcal{C}_1 \sqsubseteq \mathcal{C}_2$;
  - *pattern P′ in O′*: $\exists \mathcal{R}'_2.\mathcal{C}'_4 \sqsubseteq \mathcal{C}'_3$, $\exists \mathcal{R}'^-_2.\mathcal{C}'_3 \sqsubseteq \mathcal{C}'_4$, $\exists \mathcal{R}'_3.\mathcal{C}'_5 \sqsubseteq \mathcal{C}'_3$, $\exists \mathcal{R}'^-_3.\mathcal{C}'_3 \sqsubseteq \mathcal{C}'_5$, $\mathcal{C}'_3 \sqsubseteq (\exists \mathcal{R}'_2)$, and $\mathcal{C}'_3 \sqsubseteq (\exists \mathcal{R}'_3)$.
- *pattern's full formalisation*:
  - *homogeneous mappings*: between $\mathcal{C}_1$ and $\mathcal{C}'_4$ and between $\mathcal{C}_2$ and $\mathcal{C}'_5$, which may be subsumption or equivalence relations.
  - *heterogeneous alignments*: $\exists \mathcal{R}_1 \sqsubseteq \mathcal{C}'_3$, $\exists \mathcal{R}_1^- \sqsubseteq \mathcal{C}'_3$, $\mathcal{C}'_3 \sqsubseteq \exists \mathcal{R}_1 \sqcap \exists \mathcal{R}_1^- \sqcap (\leq 1\, \mathcal{R}_1) \sqcap (\leq 1\, \mathcal{R}_1^-)$.

Related to the examples–case A

- $\mathcal{C}_1 \equiv \mathcal{C}_2$ for `Person`, $\mathcal{C}'_4 \equiv \mathcal{C}'_5$ for `Person` as well, and the pattern associates the role `married to` ($\mathcal{R}_1$) with the class `Marriage` (instantiating $\mathcal{C}'_3$).
- There must be an equivalence or a subsumption mapping between the `Person` classes in both ontologies
- Note that due to cardinality constraints in P' that are not in P, the alignment *de facto* generates a new property that is a sub property of $\mathcal{R}_1$.

Outline

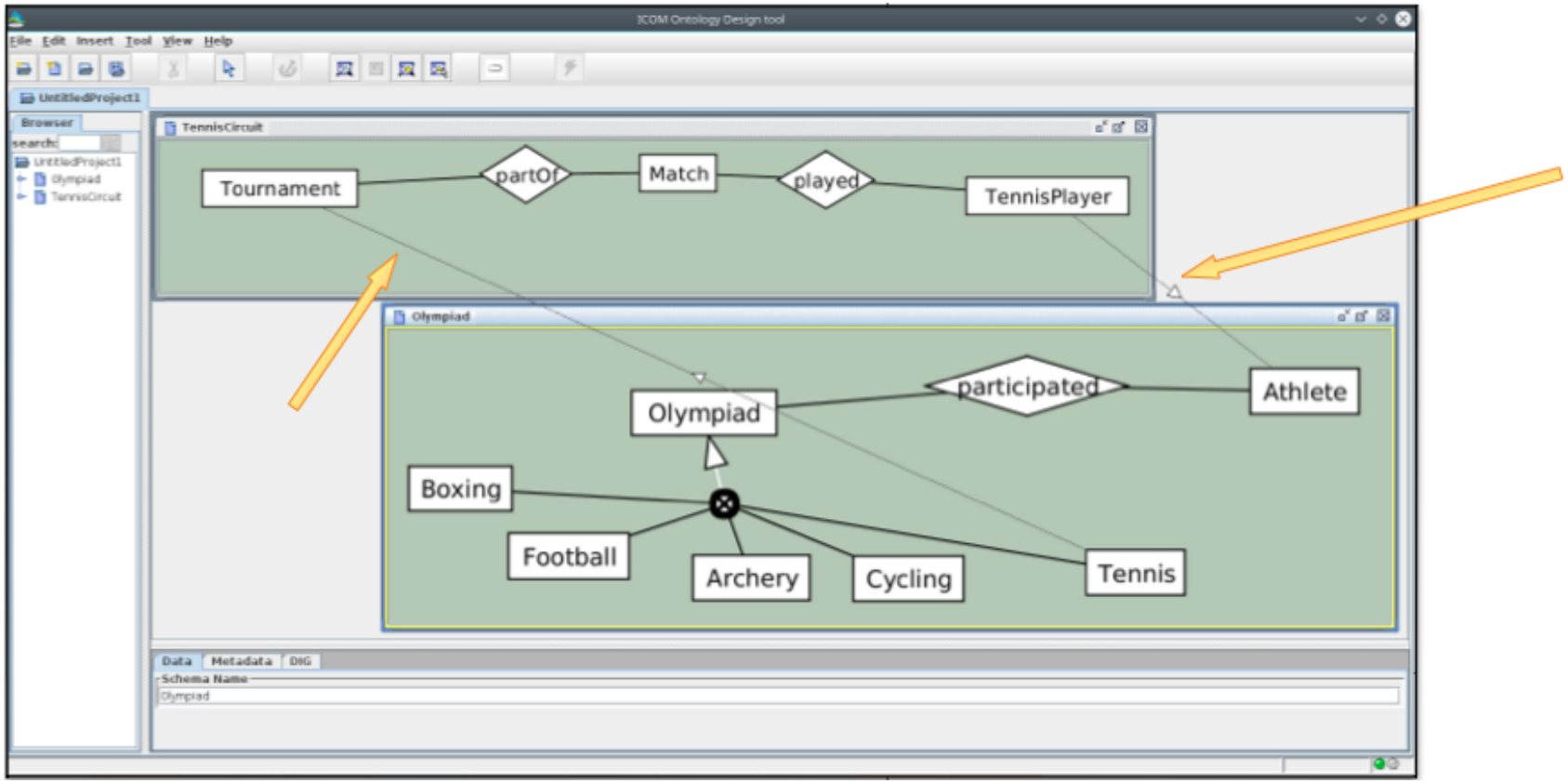
Introduction

- First algorithm to search for all possible matching instantiations in any pair of ontologies (taking into account existing homogeneous mappings)
- Uses OWLink's IsOPSatisfiable for checking OP satisfiability and GetSubClasses with 'direct' flag
- Running time $m^2 r^3 c$, with $m =$ no. homogeneous mappings, $r =$ no. OPs and c no. of classes
- Second algorithm checks whether a proposed instantiation follows the formalised pattern properties

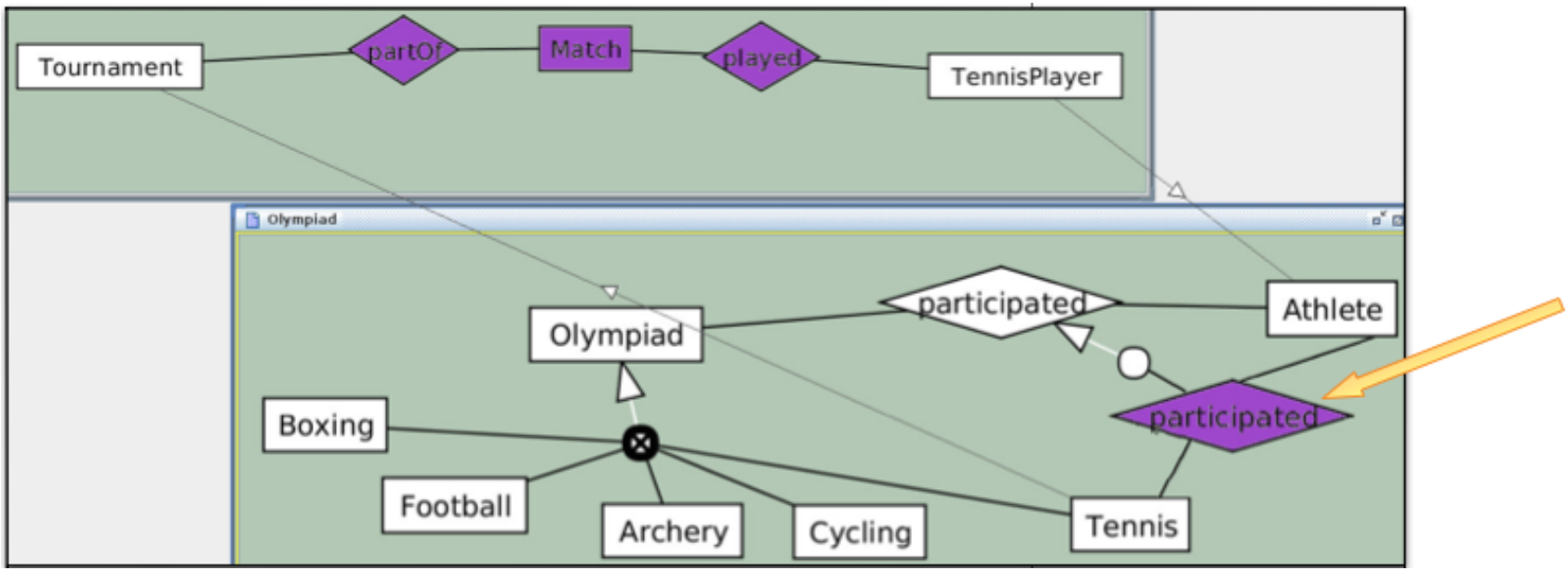
Introduction

- First algorithm to search for all possible matching instantiations in any pair of ontologies (taking into account existing homogeneous mappings)
- Uses OWLink's IsOPSatisfiable for checking OP satisfiability and GetSubClasses with 'direct' flag
- Running time $m^2 r^3 c$, with $m =$ no. homogeneous mappings, $r =$ no. OPs and c no. of classes
- Second algorithm checks whether a proposed instantiation follows the formalised pattern properties
- Paper presents algorithms for pattern A; others can be developed analogously.
- If the pattern references a FO (pattern B, D, and E): shorten the algorithm execution time by restricting the search of candidates to only the descendants of the FO's elements in the pattern.

Example: mapping and searching ('Case A': Class↔OP)



Example: checking and accept/reject alignment



Outline

Discussion

- Layered approach with OPs, as in [Scharffe and Fensel(2008)], but extends this to *patterns* (> 1 element on *both* sides)

Discussion

- Layered approach with OPs, as in [Scharffe and Fensel(2008)], but extends this to *patterns* (> 1 element on *both* sides)
- The algorithms have been designed such that one can do a local search compared to brute force exhaustive searches across the whole ontology
- Combinations resulting in undecidability (notably: due to the regularity constraint)—our patterns remain within OWL 2 DL

Discussion

- Layered approach with OPs, as in [Scharffe and Fensel(2008)], but extends this to *patterns* (> 1 element on *both* sides)
- The algorithms have been designed such that one can do a local search compared to brute force exhaustive searches across the whole ontology
- Combinations resulting in undecidability (notably: due to the regularity constraint)—our patterns remain within OWL 2 DL
- Various possible extensions: *replacing* a pattern in a single ontology for TDD's refactoring, preparation for FO alignment, or from domain ontology to conceptual model for OBDA

Conclusions

- First detailed formalisation of an ontology design pattern
- 12 patterns for 6 pattern alignments motivated from modelling viewpoint, formalised
- Efficient local search and pattern matching algorithms that can propose possible pattern alignments
- Currently being implemented in ICOM and generalised SUGOI

References I

Jerome Euzenat and Pavel Shvaiko.
Ontology Matching.
Springer, 2007.

Lorena Otero-Cerdeira, Francisco J. Rodríguez-Martínez, and Alma Gómez-Rodríguez.
Ontology matching: A literature review.
Expert Syst. Appl., 42:949–971, 2015.

Dominique Ritze, Christian Meilicke, Ondrej Svab-Zamazal, and Heiner Stuckenschmidt.
A pattern-based ontology matching approach for detecting complex correspondences.
In Pavel Shvaiko et al., editors, *Ontology Matching OM-2009*, 2009.

Francois Scharffe and Dieter Fensel.
Correspondence patterns for ontology alignment.
In A. Gangemi and J. Euzenat, editors, *Proc. of EKAW'08*, volume 5268 of *LNAI*, pages 83–92. Springer, 2008.

Francois Scharffe, Ondrej Zamazal, and Dieter Fensel.
Ontology alignment design patterns.
Knowledge and Information Systems, 40:1–28, 2014.

Thank you!

Questions?