



University of
Rome
Tor Vergata

GENOMA

GENeric Ontology Matching Architecture

Maria Teresa Pazienza⁺, Roberto Enea⁺, Andrea Turbati⁺

⁺ART Group, University of Rome Tor Vergata, Via del Politecnico 1, 00133 Rome, Italy

Contacts: {pazienza, turbati}@info.uniroma2.it , roberto.enea@gmail.com

#AI4 AI*IA
2015
CONFERENCE

Ferrara, Italy, 23rd - 25th September 2015

Ontology development became a very frequent task.
In creating a new ontology two possible approaches exist:

1. *starting from a shared vocabulary* (as the FOAF ontology when describing people-related terms) and then enriching the new domain with specific info
2. *defining everything from scratch.*

As a consequence, using more than one ontology in a new task can be extremely difficult, since *different resources* of distinct ontologies (classes/instances/properties identified by *different lexicalizations*) can be used to identify the *same concept*.

The Ontology Matching Problem

While humans naturally deal with such an *ambiguity* and try to understand the similarity between these resources, systems cannot easily do the same.

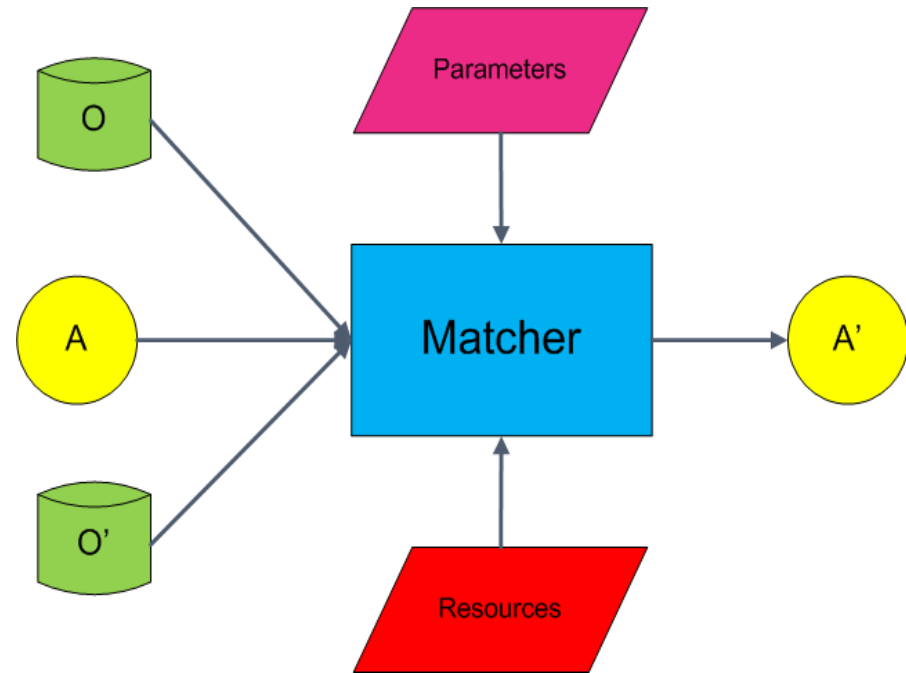
This means that even when using a formal representation for an ontology (with a given serialization), the richness of the natural language is still an issue!



The Ontology Matching Problem

Ontology Matching as the branch of the Ontology Engineering that aims in finding similarities between *resources* of two or more ontologies.

The input to the framework consists in two or more ontologies from which, using a matcher (single or multiple one), an *alignment matrix* is produced.



Main differences among existing O.M. tools are:

- the *size of the ontologies* they are able to manage
- the *formal language* in which the ontologies should be written (mostly RDF and one of its serialization, such as RDF/XML or N-Triples)
- which *resources they are able to compare* (classes, instances, properties)
- the *natural languages* in which the two ontologies should be written (or if they are able to compare ontologies written in different languages);
- the *cardinality* of the output alignment for each resource it returns (*1:1 or n:m*)
- *open source or proprietary* (a common problem when dealing with any software tools)
- if they *use external data or not* (this can affect their license as well);
- adoption of just *syntactic matching* or also *some sort of semantic matching approach*

O.M Tools: Comparison

	Falcon	ASMOV	Cupid	iMap	GLUE	COMA++
Specific for big ontologies	Yes	No	No	No	No	No
Input	RDF/RDFS/ OWL	OWL	XML	XML, DB Schemas	XML, DB Schemas	RDF/RDFS/ OWL
Resources matched	Classes + Properties + Instances	Classes + Properties	XML Elements	XML, DB schema Elements	XML, DB schema Elements	Classes + Properties
Specific Natural languages	No	English	English	No	No	English
Output alignment	1:1	m:n	1:1	1:1	1:1	1:1
Open source	Yes	No	No	No	No	Yes
Linguistic resources	No	WordNet	WordNet	No	No	WordNet
Type of matching	Linguistic Structural	Linguistic Structural Extensional	Linguistic Structural	Parallel searcher + similarity estimator	Statistical Approach, Machine learning	Linguistic

A generic architecture

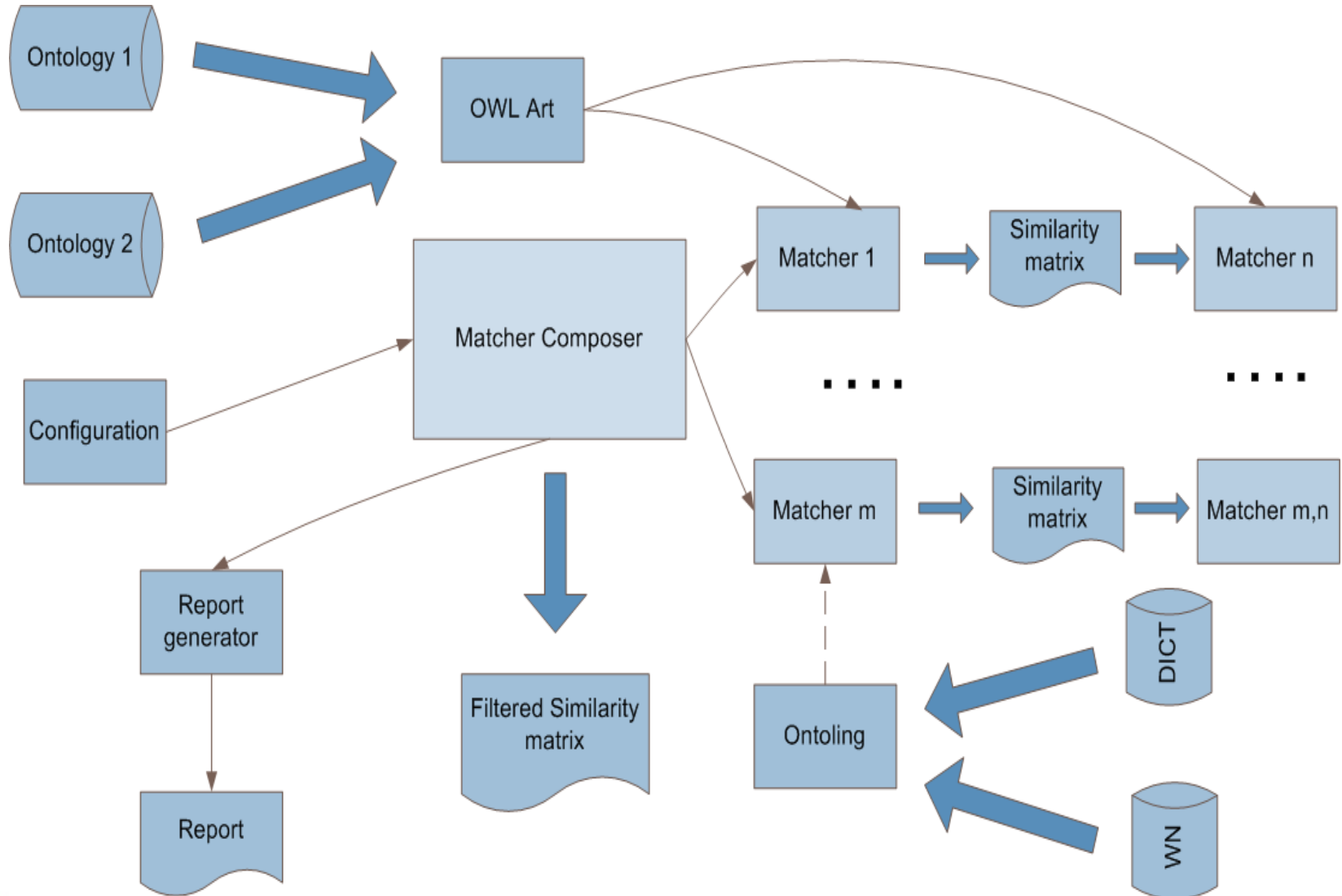
to help either experts or beginners
to assemble matchers in different modalities
and choose the best combination
given a specific application context.

Gen.O.M.A (**Generic Ontology Matching Architecture**) or GENOMA is a sort of meta architecture helping in developing new specific architectures for the ontology matching task.

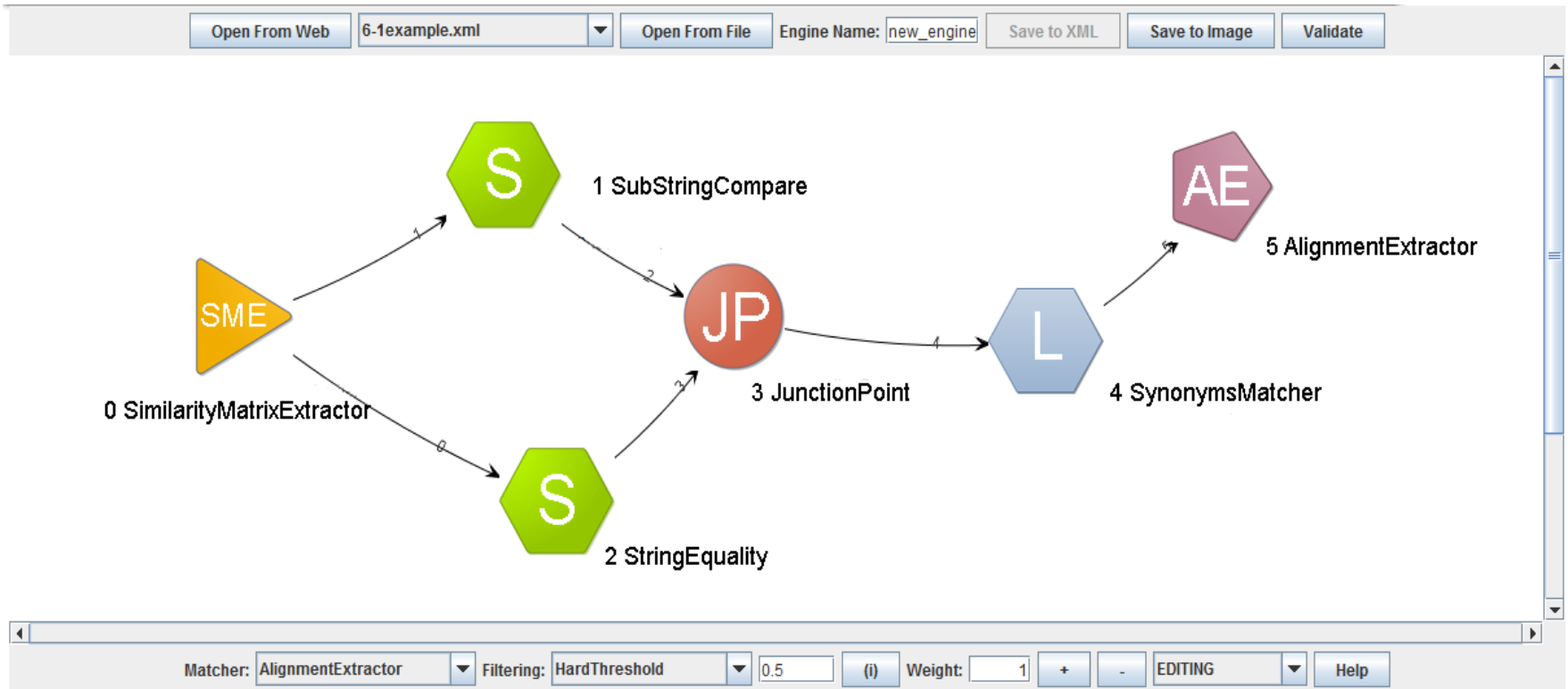
Some of its main features:

- WYSIWYG tool for the composition of the matching architecture
- Reusability of the matchers/use of newly created ones
- Expandability of the matcher's dictionary
- Deployable on distributed systems

GENOMA : Architecture



To help the user in creating, visualizing and validating an architecture for an O.M. tool, GENOMA provides an easy to use and complete **User Interface**

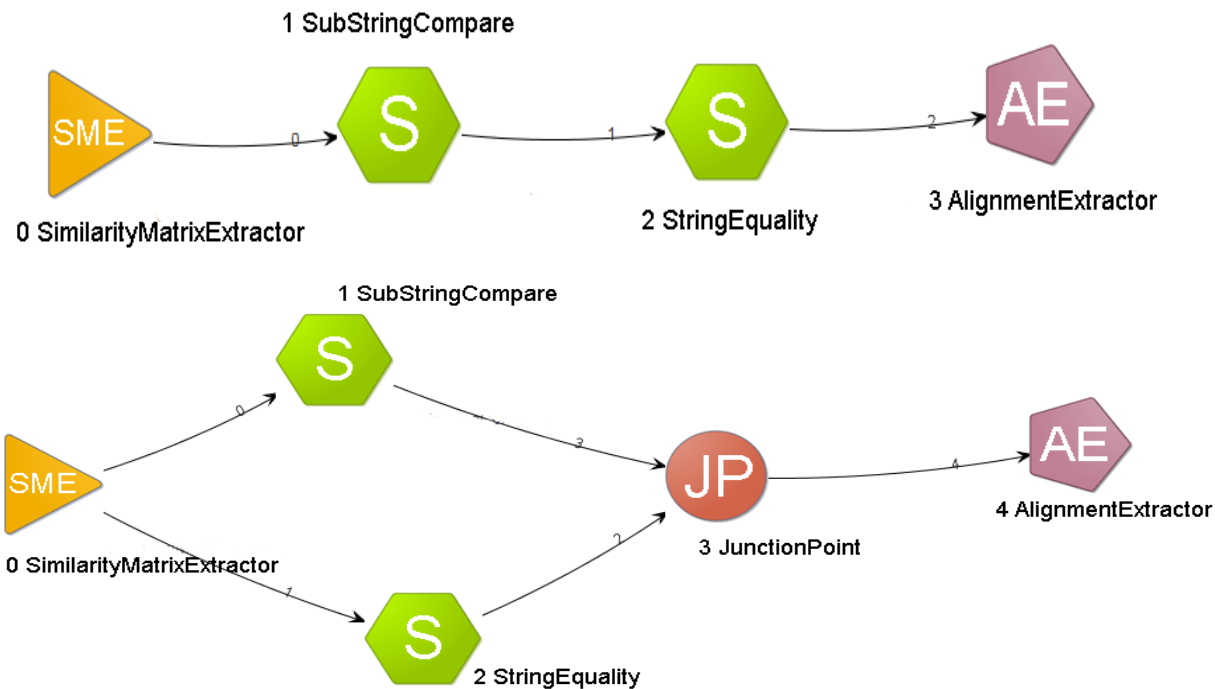


Creating a new architecture is completely interactive and ***mouse oriented*** activity

- By selecting the desired matcher, the user specifies the parameters values (or accepts the default ones)
- Then link the matchers in several possible deployment configurations.

All this is achieved easily by using just a few clicks of the mouse

By first, in defining an architecture, the user decides whether to adopt a parallel or a series approach given two matchers. Two base configurations:



GENOMA is an architecture that helps developing, deploying and validating complex and totally customizable O.M. architecture and tools. These architectures can be changed any time, to find the better one given a specific domain or application needs, leaving the user the complete freedom of experimenting while defining its architecture.

download from:

<http://bitbucket.org/aturbati/ontology-matching-architecture>

