

# An Intelligent Ontology Alignment Tool Dealing with Complicated Mismatches

Efthymios Chondrogiannis<sup>1</sup>, Vassiliki Andronikou<sup>1</sup>, Efstathios Karanastasis<sup>1</sup> and Theodora Varvarigou<sup>1</sup>

<sup>1</sup> National Technical University of Athens, 9 Heroon Politechniou Str, 15773, Athens, Greece  
{chondrog, vandro, ekaranas}@mail.ntua.gr, dora@telecom.ntua.gr

**Abstract.** Ontologies comprise an explicit specification of a domain conceptualisation. However, their design, structure and elements may significantly vary, even for a specific concept, given their subjective nature, which highly depends on their purpose and the view from which they are developed. Especially in cases that the ontologies serve as a means for linking different data sources and enabling meaningful information exchange, a variety of mismatches needs to be handled. This paper presents a web-based ontology alignment tool which supports the whole alignment process with particular focus in complex mismatches. This tool facilitates ontologies alignment by suggesting possible correspondences which are automatically recalculated as the alignment process progresses.

**Keywords:** Semantic Web, Ontologies Alignment, Correspondence Patterns, Mapping Languages, Matching Techniques

## 1 Introduction

The most widely used definition of an ontology in computer science was formed by Gruber in 1993, based on which it constitutes an explicit specification of a conceptualisation [1]. In fact, an ontology constitutes an agreement (possibly incomplete) about a conceptualisation [2], the design of which is driven by the purpose it serves as well as the background of experts developing it. Hence, ontologies published within a domain – even for the same topic and purpose – may differ in the structure and meaning of elements.

The purpose of ontologies alignment is to alleviate the differences among semantically overlapping ontologies, by specifying correspondences among them. Bridging such a gap constitutes an especially challenging issue for the Semantic Web community, taking into account the variety of mismatches which may be encountered [4], while it can also trigger research in relevant fields such as query and results translation, data integration, ontology evolution, etc.

In our previous work [3], we have presented a framework for accessing EHRs' data located at different data sources through a common interface (Reference Model). A crucial step in this process is the mapping of the Reference Model with the one used within each Healthcare Entity for capturing patients' data; a quite complicated process

due to the differences expected, as a result of the EHRs independent development and the poor adaptation of standards by healthcare entities. For instance, the coding schemas used for capturing hematological examinations may differ. Also the model for the hematological examinations may represent the outcome differently among healthcare entities, for example as three independent properties in one entity (value, unit and range of normal values) and as two properties in another entity (value and its unit as one parameter and normal value range as the other one), with direct impact in the structure and meaning of the elements defined. Nevertheless, the fact that ontologies alignment takes place during the *design* enables domain experts to participate in the process and improve the quality of the mappings manually specified as well as those suggested through any automated process.

Given the complexity and the variety of the mismatches across ontologies, the available tools dealing with their alignment provide only partial solutions, whereas, to our knowledge, there is no publicly available tool that can adequately handle such mismatches which *involve more than one element from each ontology*. In this work we present a web application that can be used for ontologies alignment purposes. The tool covers all process-driven requirements specified by Granitzer et al. [16] enabling the end users to simultaneously examine ontologies, handle suggested mapping rules, define new ones, and, finally, export them in a separate file of the desired format.

The paper is structured as follows. In section 2 we present related work in the domain of ontologies alignment, while section 3 focuses on the parameters required when specifying a correspondence among entities. In section 4, we provide an overview of the web application which has been developed and a detailed description of the functionalities it provides. The evaluation of the proposed Ontologies Alignment Tool (OAT) follows in section 5, while section 6 summarises the work presented and new challenges to be covered in our future work.

## 2 Related Work

For specifying the correspondences among ontological elements, a variety of ontologies alignment tools exists, such as SAMBO, Falcon-AO, OPTIMA, COMA 3.0 and AgreementMaker.

SAMBO [13] is a system for aligning and merging biomedical ontologies. It provides an interface for interaction with domain experts, while it uses a variety of techniques for detecting similarity among elements including terminological and structural matchers, domain knowledge (UMLS [34]) as well as data obtained by life science literature. The 1:1 correspondences detected are filtered based on a threshold specified and presented to the end user for specifying the final alignments. Falcon-AO [14] is another ontology matching system, an important part of which is the automatic ontology matching component. It utilises two light-weight linguistic matchers, an iterative structural matcher for detecting 1:1 correspondences, while it has adopted the divide-and-conquer strategy for mapping large ontologies. The graphical user interface (GUI) allows users to set matching parameters and manage alignments detected.

OPTIMA [30] is a general purpose tool for ontology alignment. The tool provides a user interface for visualisation and analysis of ontologies. It exploits the structural and lexical similarity between the schemas for detecting possible alignments, enabling users to save the detected alignments in XML. COMA 3.0 [32] is a schema and ontology matching tool. The user is able to load schemas or ontologies through its interface, while it uses linguistic and structural matchers for detecting equivalent correspondences among terms. The user can also participate in the alignment process and specify the strategy that should be used (i.e., concrete matchers) for producing more accurate results than the "default" ones. Moreover, users are able to manually specify complex correspondences such as when two elements should be combined and a data transformation should be applied.

AgreementMaker [31] is another tool that can be used for mapping large ontologies. It is equipped with a graphical user interface through which the user is able to examine ontologies presented as a tree and the mappings produced from its matchers as well as to define their own mappings. During the alignment process, it allows the application of one or more matching techniques and the visualisation of their results.

Overall, the existing mapping tools focus on discovering 1:1 alignments, while many of them either provide a simple interface for specifying 1:1 correspondences or are not equipped with a graphical user interface at all [15]. Hence, interaction with end users for supervising and/or contributing to correspondence detection is difficult. Especially when an n:m correspondence among entities needs to be specified, the majority, if not all, of the tools fail to efficiently handle such cases through an interactive, user friendly and supportive web interface.

Another aspect that needs to be considered is the mapping language being used in the background for the formal expression of correspondences. In fact, the underlying mapping language may pose significant limitations to the ontologies alignment tool concerning both the suggestion of mapping rules as well as their manual specification. Hence, it should be expressive enough in order to precisely define correspondence among elements taking into account the various mismatches which may be encountered.

Correspondence Patterns (CPs) [11] and Ontology Patterns (OPs) [12] provide a quite good approach for ontologies alignment. They satisfy a series of requirements [10], including expressiveness, support of conditional mappings, combination of the declaration and procedural part, while correspondences can be formally expressed through a mapping language such as Expressive Declarative Ontologies Alignment Language (EDOAL) [7]. EDOAL extends the Ontologies Alignment API [8], overcomes its limitations and provides a more flexible language to the end user for coping with the mismatches identified [9] that cannot be dealt with, for example, using OWL constructors [5] and Context OWL [6]. Apart from CPs/OPs, the use of *query*-driven methodologies for ontologies alignment purposes has been also presented in literature [29]. The latter enable users to specify 1:1 correspondences as well as more complicated ones using Global-As-View (GAV) and Local as view (LAV) approaches or a combination of them (GLAV) [18].

Concerning the process being followed for detecting candidate mapping rules, a variety of algorithms and techniques exists, as mentioned above. More precisely, for detecting the similarity among entities we can use both string based (e.g., edit

distance) and language based techniques (e.g., stemming), take into account the ontology structure (e.g., properties of each class) as well as the axioms specified (e.g., classes hierarchies, properties cardinality restrictions), possibly use external knowledge (e.g., a dictionary with terms synonyms) or even unstructured data available in literature [16]. In the majority of ontologies alignment tools the aforementioned techniques or a combination of them is being used for detecting *similarity* among entities. However, an interesting issue is how we can use such techniques for producing more complicated correspondences.

### 3 Mapping Rules

Ontologies alignment involves the specification of one or more mapping rules (aka correspondences or alignments). A *mapping rule* intends to precisely determine the correspondence among ontological elements so that we can “move”, for instance, from one ontology to the other one, ideally without information change or loss. The mapping rules specified, especially the complex ones which involve more than one element from each side, are closely related with the *purpose* they serve. This is perfectly rational taking into account that the outcome of the ontologies alignment process (being expressed in a mapping file) is not the ultimate goal, but it constitutes an intermediate, still necessary, step for supporting relevant tasks such as query and results translation.

When specifying a mapping rule, a series of parameters need to be determined, including the participating entities along with the relation among them. Concerning the formal description of the entities presented in each side of a mapping rule, we have been based on Ontology Patterns. An Ontology Pattern (OP) precisely determines an entity of a mapping rule and it may refer to an existing element (e.g., an existing class), a “new” one implied by the restriction of meaning and/or usage of an existing element (e.g., restricting the domain of a property within a specific class) or generally any combination of ontological elements (e.g., union of classes). In Table 1 we present the available OPs for specifying a Relation. We prompt readers to have a look at the OAT site [28] for an extended list of available OPs.

**Table 1.** Available Ontological Patterns for specifying a Relation based on existing elements

Ontological Pattern	Abbrev.	Description
Simple Relation Pattern	SR-P	An existing Relations (Object Property) described by a URI.
Inverse Relation Pattern	IR-P	The inverse relation implied by the Relation specified.
Relation Domain Restriction Pattern	RDR-P	A Relation which can be used with those instances that belong to a specific Class.
Relation Range Restriction Pattern	RRR-P	A Relation which can point to those instances that belong to a specific Class.
Relation Path Pattern	RP-P	A Relation implied by the path formed by two or more Properties

It should be noted that the OPs are defined in such a way and have innate nested capabilities so that they can be combined to formulate more complex ones (see Fig. 5), increasing the mapping capabilities of the language offered to the end users for ontologies alignment. For instance, the Properties Collection Pattern (PCP) allows "grouping" of properties of the same class, the values of which are strongly interdependent. An example is the hematological examination result, which is being defined by a value followed by a unit or a units' expression. However, both value and unit can be captured by an Object or Datatype Properties (also see Fig. 4) depending on the modelling decision made during the design of the ontology. The nested capabilities of OPs allow for the detailed specification of the patterns' elements (in this case, the value and unit properties in the PCP).

When mapping properties, it may also be necessary to apply some changes to their values, when, for example, the mapping rules are being used for translating data from one ontology to the other one. For instance, in both ontologies we may find a property used for capturing the monthly income of a person, which may also have the same type/range (e.g. integer, float). However, their values may be expressed in a different currency (e.g. Euro in one ontology and Dollar or Pound in the other one). In such cases, in order to precisely determine the relation among them so that we can answer not only questions about whether a person has a monthly income or not, but also whether, for example, this is above a specific value, we should also determine a *data-transformation* that should take place in the properties' values when mappings are used to support relevant tasks. Except from the "direct" transformation, we may also need to specify the "inverse" one, but this is closely related with the *direction* for which a mapping rule is valid.

In Table 2, we have summarised the parameters which should be specified when determining the correspondence among the entities of two ontologies along with a brief description for each one of them and their labelling as *mandatory* or *optional*.

**Table 2.** The (M)andatory and (O)ptional parameters of a Mapping Rule

Parameter	M/O	Description
Entity 1 and 2	M	Specifies the elements participating in the left and right side of a mapping rule.
Transformation	O	Specifies the transformation that should take place in the value(s) of properties defined
Relation	M	Specifies the relation of Entity 1 towards Entity 2 (e.g., equivalent terms)
Direction	O	Specifies the direction for which the correspondence is valid. (e.g., from ontology 1 to ontology 2)
Origin	O	Indicates the way the mapping rule has been produced (e.g., manually specified by the end user through OAT)
Confidence Value	O	Indicates how confident the OAT was for the suggested mapping rule, when accepted by the end user
Comments	O	Provides a human readable description of the mapping rule specified

## 4 The Ontologies Alignment Tool

The Ontologies Alignment Tool (OAT) [28] aims at facilitating the user in the specification of the mapping between two ontologies. Hence, its main functionalities include: i. Loading of the two ontologies on the left and right panels of its screen (being specified either through the provision of their URLs or the uploading of their Files) and navigation through their elements, ii. Manual specification of the mapping among their terms, iii. Management (acceptance, rejection, adjustment) of automatically generated mapping suggestions and iv. Exporting of the correspondences for further use.

The OAT consists of a web interface (Fig. 1) which in the background utilises the services provided by the Requests Handler component deployed on the server side. The latter exploits the Ontologies Handler and the Mappings Handler Components. The Ontologies Handler is used for uploading source and target ontologies (returns Ontology IDs), retrieving ontological elements specified (based on Ontology ID) as well as their definition (based on Ontology ID and Element URI) and finding candidate mapping rules according to the definition of entities in each ontology. The Mappings Handler component is responsible for importing or exporting/saving the mapping rules specified in the appropriate format. During the correspondence detection process the Mappings Handler is being used in conjunction with the Ontologies Handler for finding for finding instances that have been already specified.

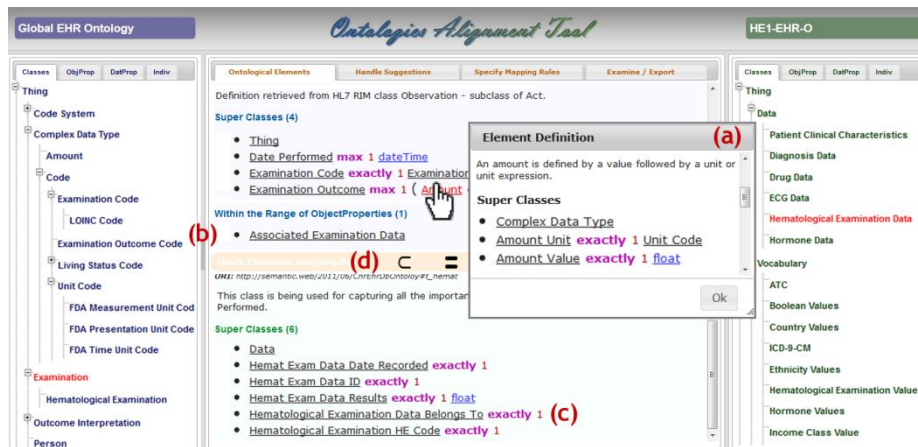


Fig. 1. Ontologies Alignment Tool after loading G-EHR-O and HE1-EHR-O.

In the Fig. 1, a snapshot from the web application after loading two - source and target - ontologies is presented. In this example, the Global EHR ontology (G-EHR-O) on the left represents healthcare patient data of interest to clinical research, while the Healthcare Entity 1 EHR ontology (HE1-EHR-O) on the right provides an ontological representation of the underline schema used for capturing patients' data within Healthcare Entity 1 (being a clinic, a hospital department, etc ).

## 4.1 Exploring Ontologies

In the left and right panels of the screen the ontological elements are presented in a tree-based view (like popular ontology editing tools with a broad user community, such as Protégé [24]) based on the axioms specified. In the middle panel (first tab), the user can further examine the entities selected, such as provided comments or any other available information (e.g. the Range of properties). In the latter case, the information about entities is presented in a pop-up window – Fig. 1a.

The information provided for each entity is not limited to what has been directly specified within the ontology, but it also includes inferred data. For instance, in case that the element being selected is an OWL class, we provide both “in-coming” (those relations that can possibly refer to an instance of this class) and “out-going” (those relations or attributes that can be applied to an instance of this class) properties. This information is very useful for better understanding ontologies and detecting possible alignments. For example, as we can see in Fig. 1b, there is a property (associated examination data) that “points” to Examinations (hence Hematological Examination) performed by a person. Moreover, in the definition of Hematological Examination Data class in HE1 (Fig. 1c) any instance of such a class has a property that indicates that such data belongs to a Patient. In other words, by simultaneously examining ontologies through our tool we can see that the aforementioned properties are related.

## 4.2 Manage Suggested Mapping Rules

In the second tab of the middle panel, the user can handle candidate mapping rules which have been automatically detected based on the definition (labels, axioms, etc.) of corresponding elements.

The screenshot shows a list of suggested mapping rules between two ontologies. The top rule is highlighted with a red arrow and a red circle labeled (a). The rule is: Hematological Examination (Equivalent) to Hematological Examination Data with a confidence score of 0.6667. The explanation for this rule is based on the Hungarian Algorithm for phrase similarity. The formula used is:  $\text{Similarity} = (\text{Best-Token-Matching-Similarity-Sum} / \text{Max-Tokens-Number})$ . The similarity calculation is:  $(1 + 1) / 3 = 0.6667$ . A table shows the token matching between 'hematological', 'examination', and 'data' in both ontologies. The 'hematological' token in the first ontology matches 'hematological' in the second (1), and 'examination' in the first matches 'data' in the second (1). The 'data' token in the first ontology has no match in the second (0). The similarity among tokens is also calculated using a formula:  $(\text{Stems-EditDistance-Similarity} + \text{Stems-N-3-Gram-Similarity}) / 2$ . The explanation lists the following token similarity calculations: 'hematological', 'hematological': 1 (since tokens are the same); 'hematological', 'examination':  $(0.2222 + 0) / 2 = 0.1111$ ; 'hematological', 'data':  $(0.2222 + 0) / 2 = 0.1111$ ; 'examination', 'hematological':  $(0.2222 + 0) / 2 = 0.1111$ ; 'examination', 'examination': 1 (since tokens are the same); 'examination', 'data':  $(0.1667 + 0) / 2 = 0.0833$ .

	hematological	examination	data
hematological	1	0.1111	0.1111
examination	0.1111	1	0.0833

Fig. 2. Suggested mapping rules when both G-EHR-O and HE1-EHR-O loaded

In Fig. 2 the candidate mapping rules after loading G-EHR-O and HE1-EHR-O are presented. The candidate mapping rules are sorted based on their confidence value (i.e., a value between 0 and 1 indicating the OAT certainty for the suggested correspondence) while an explanation for each one is also provided (Fig. 2a). The user can easily examine the definition (appearing in a pop-up window) of entities

participating at each side and *accept* or *reject* each mapping rule using the buttons presented at their right side, or massively, based on their confidence value.

When producing candidate mapping rules, the tool takes into account not only information specified in the ontologies but also the user-defined mapping rules along with the suggested ones which were rejected. In Fig. 3 we can see the candidate mapping rules proposed by the system when we have already defined two mapping rules: i. hematological examination classes are equivalent (by accepting the suggested mapping rule) and ii. person is a broader term/class than patient (manually specified by the end user – Section 4.3). As we can easily notice, the confidence value of the mapping rule between Person and Patient Unique ID properties (already suggested but not accepted yet) is being increased from 0.6296 (Fig. 2) to 0.8624 (Fig. 3a) since we have already defined the correspondence among their domain classes. We should also notice that the tool correctly detects not only 1:1 correspondence but also more complicated ones such as the correspondence among properties used to “link” person/patient with its examination data (Fig. 3b). More precisely, when the two aforementioned mapping rules have been specified, the tool identifies that the property which is used to link a person with their examination, on condition that it refers to a hematological examination, is equivalent with the inverse one used on the other side to link hematological examination data with the patient they belong to.

Person Unique ID	Equivalent ⓘ <b>0.8624 (a)</b>	Patient Unique ID	 
<b>RelationRangeRestrictionPattern</b> Relation:  Associated Examination Data RangeClass:  Hematological Examination	Equivalent ⓘ <b>0.7604 (b)</b>	<b>InverseRelationPattern</b> Relation:  Hematological Examination Data Belongs To	 
<b>RelationPropertyPathPattern</b> Relation:  Examination Code Property:  Code Value	Equivalent ⓘ <b>0.7083</b>	Hematological Examination HE Code Value	 

**Fig. 3.** Suggested mapping rules when we have already defined the correspondence among hematological examination and person/patient classes.

#### 4.2.1 The underlying mechanisms

The correspondence detection algorithm (used in the background for detecting candidate mapping rules) is based on the similarity among OPs rather than similarity of the elements defined. More precisely, for each one of the two ontologies, any possible OP based on their elements is produced and, accordingly, the similarity among them is measured. For example, for each Object Property an instance of SR-P, an instance of IR-P, any possible instance of RDR-P and RRR-P and finally any possible instance of RP-P (also see Table 1) is produced taking into account the domain and range of properties (if specified) along with the classification of terms. Accordingly, the similarity among ontological patterns from the two ontologies is measured taking into account the specific type of OPs being examined. In particular, the similarity among relations is calculated using the expression (1) with parameter k



(part of system configuration) being greater than one. It should be noted that the three key features of each relation (Domain/Range Classes and Relation Label) depend on the specific Relation OP presented. For instance, in case of RDR-P the “domain” class being used when calculating expression (1) is the restricted one rather than the initial one specified in the definition of property.

$$( \text{Domain-Class-Sim} + k * \text{Property-Label-Sim} + \text{Range-Class-Sim} ) / ( 2 + k ) \quad (1)$$

The candidate mapping rules consists of those OPs for which the similarity (expressed through the confidence value) calculated is above a predefined, yet customisable, *threshold*. Given that this threshold is expected to vary depending on the ontologies being aligned as well as their domain, it has been set as part of tool configuration. It should be noted that in case that more than one mapping options are available for an existing ontological element (or group of them) as a result of different OPs produced for each one, we select those ones with the highest confidence value.

A key characteristic of the algorithm is the fact that we take into account any mapping rule specified or even those rejected. For example, in case two classes are equivalent, their similarity is 1 whereas in the opposite scenario where their mapping has been rejected, their similarity is set to 0. Consequently, the confidence value among relevant relations or properties is being improved, when corresponding domain and/or range classes are mapped (either manually or by accepting a suggested mapping rule).

The similarity among Labels has a distinctive role in the above process and it is being calculated using a variety of techniques. Initially, we retrieve the “main” token from each Label ignoring punctuation characters and stop words [19] which do not actually add meaning in the phrases. Accordingly, we produce a matrix with similarity among tokens. The latter is based on the Porter stemming algorithm [21] for retrieving their stems (hence, cope with words’ variations) as well as a combination of Levenshtein Distance [20] and N-3-Gram [23] for the similarity of the remaining sequence of characters. Finally, we use the Hungarian Algorithm [22] for finding the best “matching” among tokens so that we maximise overall similarity, while ignoring the order of tokens presented (Fig. 2).

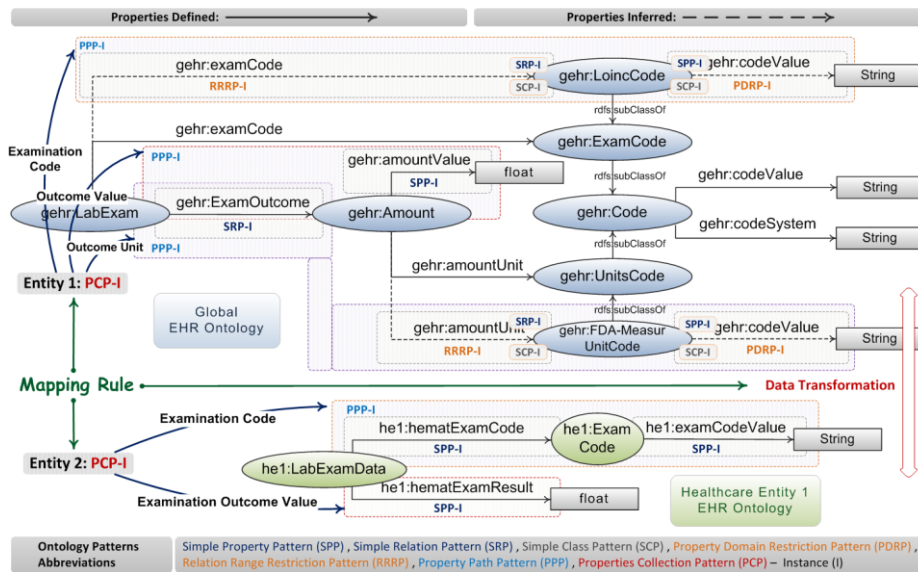
### 4.3 User Defined Mapping Rules

In order to allow for the proper and complete alignment of ontologies, the inference of which might be quite difficult or may require data which are not available to the tool for automatic suggestion, the OAT allows the user to also introduce their own mapping rules. For example, the correspondence between the two classes Person (from G-EHR-O) and Patients Clinical Characteristics (from HE1-EHR-O) has not been detected, despite the fact both are being used for the same purpose (capture basic patients data, such as ID, and demographics). For this purpose, we will manually define such 1:1 correspondence using buttons presented in the first tab (Fig. 1d).

Also, if we carefully examine the definition of Hematological Examination class in G-EHR-O along with the corresponding one in HE1-EHR-O we will notice that they are quite similar; they both capture examination code, date performed and its

outcome. However, the definition of corresponding elements especially for examination code and its outcome differ a lot. First of all, in the G-EHR-O LOINC [33] terms are being used for hematological examinations whereas in the HE1-EHR-O “locally” defined terms/codes. So, when moving from one side to the other one, we should find the corresponding code(s) used in HE1 for the hematological examination(s) specified. Also, the outcome in the G-EHR-O is an “Amount” – Fig. 1a – which in turn has a value and a unit of measurement, whereas in the HE1-EHR-O ontology the same information is being captured by only one property (Examination Results) the value of which is a “float” number. In the HE1, the units of measurement is predefined for each examination and stored in a separate XML File, as mentioned in the comments existing in the data source (also included in the ontology). Consequently, the units of measurements specified for an examination outcome based on the terms of G-EHR-O may be different than the ones used in HE1.

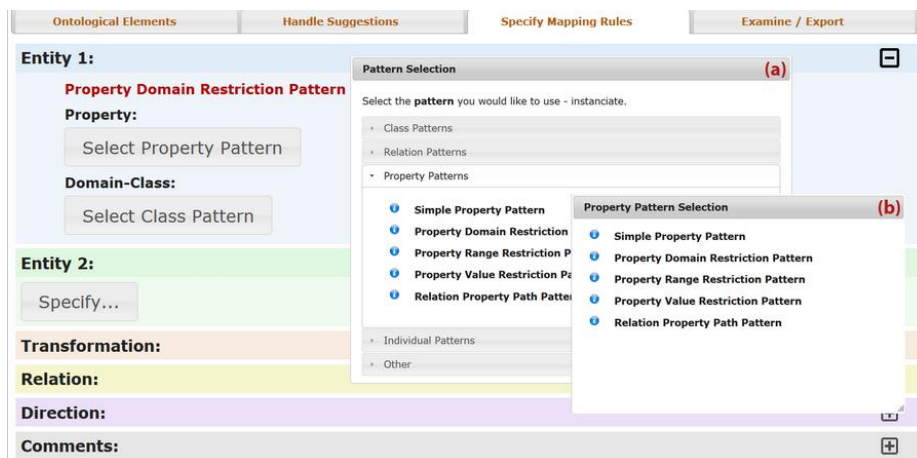
In Fig. 4 the correspondence among the aforementioned ontological elements is being presented as well as the OPs that we should use/instantiate for specifying such a correspondence. It should be noted that the complex mismatch presented in this example stems from the fact that while the G-EHR-O provides a conceptualisation of a domain (i.e., patient data for study recruitment purposes), the HE1-EHR-O is quite close to the structure being used for recording such data. Also, in the G-EHR-O international classification systems are being used, whereas in the HE1 “local” terms/codes.



**Fig. 4.** Mapping among elements used in G-EHR-O and HE1-EHR-O for a hematological examination specified as well as its outcome.

For specifying the correspondence among aforementioned entities, we should use/instantiate a variety of OPs (Fig. 4). For this purpose, the OAT provides an interactive interface in the third tab that can be used for instantiating and combining

the appropriate OPs as well as determining all parameters of a mapping rule (Fig 5). Initially, in both entities, we should use the Properties Collection Pattern (PCP) and, accordingly, determine the properties referring to the examination and its outcome (i.e., value and unit). For specifying the aforementioned parameters, especially in the first entity, we use the Properties Path Pattern (PPP) – a property composed by one or more relations followed by a property – and accordingly, we specify the internal OPs. The OAT facilitates the instantiation of OPs along with the specification of corresponding elements for both entities 1 and 2, while offering *auto-complete* as well as *copy/paste* functionality.



**Fig. 5.** The GUI for manually specifying a Mapping Rule, especially a complicated one. The pop-up window (a) appears when specifying an entity whereas (b) for the internal OPs.

In the aforementioned example, a “direct” data transformation is necessary when translating data or queries from G-EHR-O to HE-1-EHR-O, which will be responsible to find the corresponding hematological examination code(s) in HE1 as well as make the necessary changes in the value(s) provided taking into account the units of measurement specified and the ones (based on hematological examination) by which examination outcome values are expressed in the HE1.

#### 4.4 Examine and Export Mapping Rules

In the last tab of the middle screen the correspondences which have already been specified are presented. In this screen, the user can further examine the elements aligned (with their definition being presented in a pop-up window) and delete a mapping rule in case they detect an error. Furthermore, the user can export the correspondences specified in the following formats: i. JSON, ii. XML–EDOAL, iii. HTML and iv. OWL. The latter enable users to merge the mapping rules specified with ontologies and in combination with an OWL reasoner can be used for detecting inconsistencies.

Through this tab, the user can also reload existing mapping files and update them. It should be noted that this functionality is offered provided that the mapping file is in JSON format. The reason is that, unlike HTML, a JSON file is well structured. Moreover, it contains all parameters specified for each mapping rule, whereas the EDOAL XML format involves some information loss (e.g., the human readable description of mapping rule specified is stored using XML comments and hence its retrieval is hard). However, we are planning to also support loading mapping rules from EDOAL XML and OWL Files (ignoring some data such as comments) in the next versions of OAT.

## 5 Evaluation and Discussion

The evaluation of a GUI-equipped mapping tool differs from the respective process followed regarding ontologies alignment algorithms with automatic detection of candidate correspondences, which are typically based on precision and recall or a combination of them (f-measure). For this reason, a framework has been proposed by Paulheim et al. [17] taking into account the quality of the mappings produced along with human interventions required. However, the whole approach is based on the *cost* of the users' actions which is difficult to be accurately measured.

### 5.1 Automatically Detected Mapping Rules and User Interventions Required

For evaluation purposes, five biomedical domain experts used the OAT for specifying mappings between twenty biomedical ontologies including reference models, such as the G-EHR-O presented, and ontologies produced in a semi-automatic way based on the schema of relational databases, such as the HE1-EHR-O. The outcome of this process showed that more than half of the total mapping rules specified in each case have been correctly detected by the tool. It should be noted that among these correspondences, there were not only 1:1 but also more complex ones, such as the one presented in section 4.3, which popular mapping tools such as SAMBO, Falcon-AO and Optima could not detect. Nevertheless, as the main focus of this paper lies in the overall Ontology Alignment tool, a thorough presentation and evaluation of the automatic correspondence detection process will be part of our future work.

Another important parameter is the human *effort* required for detecting and specifying mapping rules. As presented in section 4.3, the user can determine the correspondences while examining the definition (e.g., description and axioms specified) of corresponding elements (Fig. 1d). Based on the domain experts' feedback, they were presented with *enough* and *clear* information allowing them to easily *detect* and *express 1:1 - but most importantly n:m correspondences*. In fact, the interactive interface of OAT allowed the end users to efficiently instantiate the appropriate OPs within each entity, as well as to combine them for expressing more complicated correspondences (Fig. 5). Moreover, the users could easily specify the corresponding ontological elements within each entity through the auto-complete and copy-paste functionalities offered. However, providing an estimation about the "cost"

of the users' interventions, especially for the complicated mapping rules, is rather complicated and is considered to be out of scope for this work. An option would be to measure the average number of entities examined before forming a mapping rule or the total time needed for specifying each one, although this strongly depends on the user's level of familiarity with the tool, the complexity of the mapping rule and the available information for the ontology elements, among others.

## 5.2 User Driven Requirements Covered

The Ontologies Alignment Tool (OAT) has been primarily designed so that it can be used by domain experts rather than software agents. For evaluating the extent to which the tool covers our needs, as well as, its user satisfaction and friendliness, we have asked from five domain experts to use it for mapping two ontologies of their interest and, accordingly, answer to which extent (from 0 to 10) each one of ten Nielsen Heuristics [25][26] covered, along with justifying their answers. All users agreed on the fact that OAT adequately covers their needs, especially when dealing with n:m correspondences. Also, the end users reported that they were able to precisely define *every* mapping rule required, including complex ones, which they could not specify with any other mapping tool available.

In order to verify that all desired functionalities are provided by OAT we have also compared them against the User Driven Requirements specified by Granitzer et al. [16] for a semi-automatic ontologies alignment tool. The evaluation showed that the majority of these requirements are already covered in OAT, whereas we should also provide an *overview* of suggested correspondences (a need already mentioned by the end users). More precisely, *classification* of both suggested and manually specified mapping rules based on their features (e.g. type of elements involved) is necessary, especially when mapping large ontologies with thousands of terms. Also a *visualisation* of the specified mapping rules can facilitate end users to better understand the proposed correspondences (especially complex ones) and more easily detect the ones missing.

Another interesting requirement not currently covered is *collaborative ontologies alignment*. This functionality is considered useful especially in cases that the size of the ontologies is rather large and, hence, the mapping process requires collaborative effort by several experts. However, this functionality is considered to be of lower priority, compared to the other ones, and, hence, is to be included in future releases of the tool.

## 6 Conclusion and Next Steps

In this paper, we have presented a flexible web application for specifying complex correspondences among the terms of ontologies. The tool presented also suggests possible correspondences for accelerating the ontologies alignment process, while the user can easily manage them or define new ones. Moreover, the specified mappings are presented in a user friendly manner, while they can be easily exported in a variety of desired formats. The evaluation of the tool indicated that it can adequately cover

the ontologies alignment process, by equipping users with the ability to be automatically presented with or to detect and express by themselves both simple and complex correspondences. Still, there are open issues to be considered in our future work.

Another important part of ontologies alignment is how we can use correspondences specified for supporting relevant tasks, such as query and results rewriting. In our paper [3] we have already presented an algorithm for SPARQL query rewriting, based on CPs specified within an EDOAL XML mapping file. The introduction of a more flexible mapping language based on OPs (described in section 3) requires much more complicated query and results rewriting mechanisms, which will be part of our future work and which will be responsible to dynamically detect the *complexity of the mapping rule* and make the necessary interventions in the SPARQL query provided as well as the results retrieved from the data source.

**Acknowledgements.** This work is being supported by the OpenScienceLink project [27] and has been partially funded by the European Commission's CIP-PSP under contract number 325101. This paper expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this paper.

## References

1. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition. 5, 199--220 (1993)
2. Guarino, N.: Understanding, building and using ontologies. Int. J. Hum.-Comput. Stud. 46, 293--310 (1997)
3. Chondrogiannis, E., Andronikou, V., Mourtzoukos, K., Tagaris, A., Varvarigou, T.: A novel query rewriting mechanism for semantically interlinking clinical research with electronic health records. In: 2nd International Conference on Web Intelligence, Mining and Semantics, 48:1--48:12. ACM, New York, USA (2012)
4. Klein, M.: Combining and relating ontologies: an analysis of problems and solutions. In: IJCAI-01 Workshop on Ontologies and Information Sharing, pp. 53--62. Seattle, USA (2001)
5. Web Ontology Language, <http://www.w3.org/TR/owl-features/>
6. Bouquet, P., Giunchiglia, F., Harmelen, F.V., Seraf, L., Stuckenschmidt, H.: C-OWL: Contextualizing ontologies. In: International Semantic Web Conference. pp. 164--179. Springer, Berlin (2003)
7. Expressive and Declarative Ontology Alignment Language, <http://alignapi.gforge.inria.fr/edoal.html>
8. Alignment API and Alignment Server, <http://alignapi.gforge.inria.fr/>
9. David, J., Euzenat, J., Scharffe, F., Trojahn dos Santos, C.: The Alignment API 4.0. Semantic Web, 2, 3--10 (2011)
10. Scharffe, F., de Bruijn, J.: A Language to Specify Mappings between Ontologies. In: 1st International Conference on Signal-Image Technology and Internet-Based Systems, pp. 267--271. Dicolor Press, Cameroon (2005)
11. Scharffe, F., Fensel, D.: Correspondence Patterns for Ontology Alignment. In: 16th international conference on Knowledge Engineering: Practice and Patterns, pp. 83--92. Springer Berlin Heidelberg, New York (2008)

12. Šváb-Zamazal, O., Svátek, V., Scharffe, F., David, J.: Detection and Transformation of Ontology Patterns. In: Fred, A., Dietz, J.L.G., Liu, K., Filipe, J. (eds.) IC3K 2009. CCIS, vol. 128, pp. 210–223. Springer, Heidelberg (2011)
13. Lambrix, P., Tan, H.: SAMBO-A System for Aligning and Merging Biomedical Ontologies. *Web Semant.* 4, 196–206 (2006)
14. Ningsheng, J., Wei, H., Gong, C., Yuzhong, Q.: Falcon-AO: Aligning Ontologies with Falcon. In: K-Cap 2005 Workshop on Integrating Ontologies. pp. 87–93. Banff, Alberta, Canada (2005)
15. Shvaiko, P., Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. *IEEE Trans. Kn. Dat. Eng.* 25, 158–176 (2013)
16. Granitzer, M., Sabol, V., Onn, K.W., Lukose, D., Tochtermann, K.: Ontology Alignment—A Survey with Focus on Visually Supported Semi-Automatic Techniques. *Future Internet.* 2, 238–258 (2010)
17. Paulheim, H., Hertling, S., Ritze, D.: Towards evaluating interactive ontology matching tools. In: 10th European Semantic Web Conference, pp. 31–45. Springer Berlin Heidelberg, Montpellier, France (2013)
18. Lenzerini, M.: Data Integration: A Theoretical Perspective. In: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 233–246. ACM, New York, USA (2002)
19. Common English Stop Words, <http://www.textfixer.com/resources/common-english-words.php>
20. Pieterse, v., Black, E.P.: Levenshtein distance. In: Dictionary of Algorithms and Data Structures, CRC Press LLC (1999)
21. Willett, P.: The Porter stemming algorithm: then and now. *Program: electronic library and information systems.* 40, 219–223 (2006)
22. Kuhn, H.W.: The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly.* 2, 83–97 (1955)
23. Brown, P.F., deSouza, P.V., Mercer, R.L., Della Pietra, V.J., Lai, J.C.: Class-based N-gram Models of Natural Language. *Comput. Linguist.* 18, 467–479 (1992)
24. Protégé, <http://protege.stanford.edu/>
25. Nielsen, J., Molich, R.: Heuristic Evaluation of User Interfaces. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 249–256. ACM, Seattle, Washington, USA (1990)
26. Molich, R., Nielsen, J.: Improving a Human-computer Dialogue, *Commun. ACM*, 33, 338–348 (1990)
27. Karanastasis, E., Andronikou, V., Chondrogiannis, E., Tsatsaronis, G., Eisinger, D., Petrova, A.: The OpenScienceLink architecture for novel services exploiting open access data in the biomedical domain. In: Proceedings of the 18th Panhellenic Conference on Informatics (PCI '14), ACM, New York, NY, USA (2014)
28. Ontologies Alignment Tool, <http://ponte.grid.ece.ntua.gr:8080/OntoMapping>
29. Zhang, H.: A Query Driven Method of Mapping from Global Ontology to Local Ontology in Ontology-based Data Integration. *Journal of Software.* 9, 738–742 (2014)
30. Kolli, R., Doshi, P.: OPTIMA: Tool for Ontology Alignment with Application to Semantic Reconciliation of Sensor Metadata for Publication in SensorMap. In: Proceedings of the second IEEE International Conference on Semantic Computing. 4–7, 484–485 (2008)
31. Cruz, I.F., F.P. Antonelli, and C. Stroe: AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *PVLDB* 2(2), demo paper, 1586–1589 (2009)
32. Massmann, S., Raunich, S., Aumuller, D., Arnold, P., Rahm, E.: Evolution of the COMA Match System. In: The Sixth International Workshop on Ontology Matching (2011)
33. Logical Observation Identifiers Names and Codes, <http://loinc.org/>
34. Unified Medical Language System, <http://www.nlm.nih.gov/research/umls/>