Applying an analytic method for matching approach selection

Malgorzata Mochol, Anja Jentzsch, Jérôme Euzenat

Applying an Analytic Method for Matching Approach Selection

Malgorzata Mochol[1], Anja Jentzsch[1] and Jérôme Euzenat[2]

[1] Freie Universität Berlin, Institut für Informatik
Takustr. 9, D-14195 Berlin, Germany
mochol@inf.fu-berlin.de, anja@anjeve.de
[2] INRIA Rhône-Alpes
655 avenue de l'Europe, 38330 Montbonnot Saint-Martin, France
Jerome.Euzenat@inrialpes.fr

Abstract. One of the main open issues in the ontology matching field is the selection of a current relevant and suitable matcher. The suitability of the given approaches is determined w.r.t the requirements of the application and with careful consideration of a number of factors. This work proposes a multilevel characteristic for matching approaches, which provides a basis for the comparison of different matchers and is used in the decision making process for selection the most appropriate algorithm.

1 Introduction

Many methods and tools are under development to solve specific problems in the Semantic Web; however none of these solutions can be deployed for all problems in this area. This statement is also true in the ontology matching field, in which there is no and will never be an overarching matching algorithm for ontologies that is capable of serving all (heterogeneous) ontological sources. Most of the research in this area proposes new approaches based on different principles and relies on various features. These new approaches only solve small parts of "global" problems in the matching field or fill some open matching gaps[12]. Therefore in general, when implementing an application using a matching approach, the corresponding algorithm is typically built from scratch and no attempt to reuse existing methods is made. Despite an impressive number of research initiatives in the matching field, containing valuable ideas and techniques, current matching approaches still feature major limitations when applied to the emerging Semantic Web. For example, the majority of existing approaches to ontology matching are (implicitly) restricted to processing particular classes of ontologies and thus they are unable to guarantee a predictable quality of results on arbitrary inputs. What is required are appropriate ontology matching techniques capable of coping with different levels of detail in concept descriptions[3]. Aside from the problems mentioned above, there are many other open issues, of a global nature, which need to be solved in the future. Firstly there is the question of what should be matched based upon what needs to be found. Also it is important to avoid performing relatively blind matching, while being aware of when to stop the matching process. Furthermore, the selection of a currently relevant matching algorithm that is suitable w.r.t the given specification and the definition of the appropriate criteria for this decision making process needs to be taken into account. Regarding the latter, one of the first steps on the way to solve this issue can be an infrastructure for taking advantage of existing ontology alignments. To tackle the issues of the heterogeneity of existing ontology matchers as well as to limit the disadvantages of the singular approach a *reuse strategy* for matching approaches based on the examination of their characteristics is needed. The first goal within such a methodology is the detection of potentially suitable approaches from the huge number of existing methods. After the analysis of existing approaches, evaluation of their usage context and conducting interviews with various domain (matcher) experts some factors were identified that are relevant for the selection of a suitable matcher(s) w.r.t the requirements of the given application. The objective of the work is to develop a framework that takes into account the characteristics of matching algorithms and offers methods and tools to support the process of selecting an applicable matcher. The rest of this paper is organized as follows: Section 2 gives an overview of the relevant criteria to describe and compare matching approaches. This is followed by a description of one of the methods for multi-criteria decision making called Analytic Hierarchy Process (AHP) in Section 3 and its application into a matching selection process in Section 4. The conclusions, along with future work, are discussed in Section 5.

2 How to Characterize Matching Approaches?

One of the main open issues in the ontology matching field is that of choosing a *current relevant* and *suitable* matching algorithm. Since there is no such thing as "general" matching problem, there is thus no "general" way to solve the matching issues by only posing the query "find a matching algorithms for two ontologies and deliver a set of relations". This query covers indeed ever type of ontology and matching algorithm it also gives same basics information about the alignments however it does not address the specific requirements of a particular application. The matching algorithm should not only be chosen with respect to the given data but should also be adapted to the system, taking into consideration the problem to be solved by the approach, for example merge ontologies to create new one, match ontologies to compare profiles, match data etc.
The matching problem should be seen as a collection of small particular subproblems, which are dependent on various criteria and circumstances. Following this idea, for a given (characterized) pair of ontologies to be matched, having a definition of the problem to be solved along with particular requirements regarding the final application, one must decide which matching algorithms are to be applied to satisfy these specification and to obtain the desired output. Possible attributes, that could have an impact on the selection of an adequate matching approach, must be defined in order to find a suitable solution to this issue. Accounting for the empirical findings of different case studies in ontology engineering[23–25], and regarding the requirements collected during the devel-

opment of different Semantic Web application scenarios[2, 13][3], as well as during the intensive collaborations with ontology and software engineers, six groups of factors (dimensions) has been defined as relevant for the matching selection process. These dimensions are the main aspects that must be taken into account during the examination of the suitability of a single matching approach for the solving of a given problem: (i)*input characteristic* that takes into account the ontologies to be matched; (ii)*approach characteristic* describes the matching algorithms themselves; (iii)*output characteristic* defines the desired result of the matching execution; (iv)*usage characteristic* takes into account the different situations where the approaches have been used; (v)*documentation characteristic* points out the existence and type of the documentation; and (vi)*cost characteristics* addresses the costs which have to be paid for the usage of the algorithm. The dimensions form the superficial collection for matcher attributes and build the first level of the so called *multilevel characteristic for matching approaches*. The multilevel characteristic is organized in the form of a taxonomy where dimensions are defined by sets of factors and these are described by the attributes. These characteristics can be illustrated as a hierarchical tree (cf. Fig 1) where the child nodes describe and represent the parent nodes' properties[20].

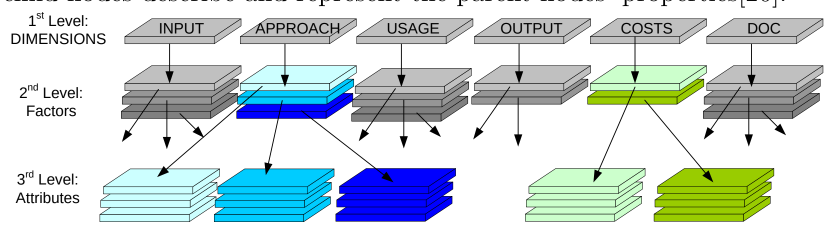

Fig. 1. Multilevel characteristic with dimensions, factors and attributes

In the following sections we briefly describe some of the factors of each dimension and state others in the form of tables (cf. Tab. 1,2,3,4,5,6) since the exact specification of all criteria would go beyond the scope of this paper.

2.1 Input Characteristic

The first step towards the analysis of the matching characteristics is the examination of the matching input. In our opinion, the attributes that describe the input are the most important and relevant criteria that play a crucial role in the selection of the appropriate algorithm. Despite the relatively large number of promising matching approaches their limitations w.r.t. certain ontology characteristics have often been emphasized in recent literature[14, 21, 22, 29, 30]. The dimension *input characteristic* describes not only the heterogeneity of the sources that are to be matched, e.g. *size* (some matchers perform well on relatively small inputs), *natural language* used for the definition of concepts (some algorithms require certain nat. language) and *input structure* (some matchers do

[3] Projects:(i)Wissensnetze,`http://wissensnetze.ag.nbi.de`,(ii)Reisewissen, `http://reisewissen.ag.nbi.de`, (iii)SWPatho, `http://swpatho.ag.nbi.de`,(iv)Knowledge Web `http://knowledgeweb.semanticweb.org/`

not perform well on heterog. structures[14]), but also takes into account *external sources*, which a matching algorithm can use for its execution (cf. Tab. 1).

DIMENSION: INPUT CHARACTERISTIC	
Factor	**Description**
Input Size (algorithm is able to handle:)	
number of ontologies	number of different ontologies to be matched (two or more)
size of input	number of ontological primitives (concepts,properties, axioms, instances) to be matched: *small* (up to 100 primitives), *middle* (100-1000 primitives) *big* (over 1000 primitives)
size of instances	number of instances to be matched: *no instances*, *small* (up to 100 primitives), *middle* (100-1000 primitives), *big* (over 1000 primitives)
number of concepts	number of concepts to be matched: *small* (up to 100 primitives), *middle* (100-1000 primitives), *big* (over 1000 primitives)
number of relations	number of relations to be matched: *small* (up to 100 primitives), *middle* (100-1000 primitives), *big* (over 1000 primitives)
number of axioms	number of axioms to be matched: *no instances*, *small* (up to 100 primitives), *middle* (100-1000 primitives), *big* (over 1000 primitives)
Input category (algorithm is able to handle:)	
glossary	a list of terms with the definitions for those terms
thesaurus	a list of important terms (single-word or multi-word) in a given domain and a set of related terms for each term in the list
taxonomy	indicates only class/subclass relationship (hierarchy)[9]
DBschema	often does not provide explicit semantics for their data
ontology	an explicit specification of a conceptual.[16]; describes a domain completely[9]
Input formality level[32, 33] (algorithm is able to handle:)	
(highly/semi) informal ontology	expressed loosely in natural language or in a restricted and structured form of natural language
semi-formal ontology	expressed in an artificial formally defined language
(rigorously) formal ontology	meticulously defined terms with formal semantics, theorems and proofs of such properties as soundness and completeness
Input model type (algorithm is able to handle:)	
task ontology	model build for a specific task
application ontology	model build for a specific application
domain ontology	model of a specific domain or part of the world
upper-level ontology	model of the common objects that are generally applicable across a wide range of domain ontologies; it describes very general concepts
Input type (algorithm is able to handle:)	
scheme	schema-based matcher
instance	instance/contents-based matchers
External sources (algorithm is able to handle /to provide:)	
additional user input	most matchers rely not only on the input to be matched (like schemas or instances) but also on auxiliary information
previous matching decision	
training matches	
domain constrains	
list of valid domain values	
dictionary	
miss-match information	
matching rules	
global schemas	
Input natural language (NL) (algorithm is:)	
NL-specific (one language)	the approach is dependent on one natural language
NL-specific (many languages)	the approach is dependent on more then one natural languages
NL-independent	the approach is language independent
Input representation language (RL)[33] (algorithm is:)	
RL-specific (one language)	the approach is dependent on one rep. language
RL-specific (many languages)	the approach is dependent on more then one rep. languages
RL-independent	the approach is independent on rep. language
Input structure (algorithm is able to handle:)	
tree structure	the approach can handle only tree-structers
graph structure	the approach can handle (heterogenous) graph structers
is-a relations	the approach can handle is-a relations
heterogeneous relations	the approach can perform not also on heterogeneous relations

Table 1. Input characteristic

2.2 Approach Characteristic

The second crucial dimension characterizes the matching approaches themselves. The corresponding factors and attributes compile a list of matcher features that are empirically proved to have an impact on the quality of matching tasks. They consider e.g. the common classification of the approaches[5, 26, 29] and distinguish between *individual algorithms*[14, 31] and combinations of the individual algorithms: *hybrid* and *composite solutions*. A hybrid approach[21] follows a *black box paradigm*, in which various individual matchers are synthesized into a new algorithm, while the composite matchers allow an increased user interaction[6, 8].

The *approach characteristic* also takes into account issues like processing type, matching ground and execution parameter (cf. Tab. 2).

DIMENSION: APPROACH CHARACTERISTIC	
Factor	**Description**
Matcher Type (algorithm is a(n):	
individual matcher	computes a mapping based on a single matching criteria
combined matcher	uses multiple individual matchers
Processing (algorithm supports:)	
manual execution	manual execution
white box paradigm	semi-automatic execution where the human intervention is possible
black box paradigm	automatic execution without human intervention
manual preprocessing allowed / required	human intervention before the execution is allowed or even required
manual postprocessing allowed /required	human intervention after the execution is allowed or even required
simultaneous execution	the single matching algorithms (within a composite matcher) can be executed simultaneously
sequential execution	the single matching algorithms (within a composite matcher) can be executed sequentially
Kind of Similarity Relation (algorithm performs:)	
syntactic matching	similarity based on syntax driven techniques and syntactic similarity measures; relation computed between labels at nodes[29]
semantic matching	relation computed between concepts at nodes[29]
Matcher Level (algorithm can perform on:)	
element level	match performed for individual schema elements
structure level	match performed for complex schema structures
atomic level	elements at the finest level of granularity are considered e.g. attributes in an XML schema[26]
non-atomic (higher) level	e.g. XML elements
Matching Ground	
heuristic	"guessing" relations between similar labels or graph structures[28]
formal	uses formal techniques (e.g. can have model-theoretic semantics which is used to justify the results)[28]
Semantic Codification Type(algorithm uses:)	
implicit techniques	syntax driven techniques[28](e.g. considers labels as strings)
explicit techniques	exploit the semantics of labels[28]; uses an external sources for assessing the meaning of labels
Execution Parameter (algorithm needs:)	
max time of execution	describes the maximal needed time of execution
max disc space for execution	describes the maximal needed disc space
precision	expresses the proportion of retrieved matches which are relevant[34]
recall	expresses the proportion of relevant documents retrieved[34]

Table 2. Approach characteristic

2.3 Usage Characteristic

One of the fundamental requirements for the realization of the vision of the fully developed Semantic Web are "tried and tested" ontology matching algorithms. Though containing valuable ideas and techniques some of the current matching approaches lack exhaustive testing in real world scenarios. Considering this problem and additionally making allowance for the fact that some of the algorithms cannot be applied across various domains to the same effect[14], it is important to know, if a particular approach has already been successfully adapted for different *domains*, *applications* and *tasks*. Additionally, the *usage characteristic* dimension also considers *different types of users*: ontology engineers who e.g. look for means to compare sources for building a new ontology or Web Services seeking automatized methods to generate mediation ontologies (cf. Tab. 3).

2.4 Output Characteristic

In addition to the input, approach and usage dimensions, the *output characteristic* (cf. Tab. 4) plays a decisive role in the process of selecting the suitable matching algorithm. Depending on the given requirements, an application can for example need a matcher that considers only some of elements of the schemes, while other systems might lack a match for all elements. One of the key factors in this dimension is the *cardinality* (global vs. local cardinality) which specifies whether a matcher compares one or more elements of one scheme with one or more elements of another scheme (in some cases the results are based on a one-to-one mapping between taxonomies[7] and in others on one-to-n).

DIMENSION: USAGE CHARACTERISTIC	
Factor	**Description**
Usage goal (algorithm is build for:)	
local use	approach developed for local use
network use	approach developed for network use
internet-based use	approach developed for internet-based use
Application Area (algorithm is build for:)	
reuse of sources	the matching approach is applied to ontology reuse which may be defined as a process in which available knowledge is used as input to generate new ontologies
usage of sources	the matching approach is applied to use the ontologies (within an application) e.g. to compare profiles
integration	reusing available source ontologies within a range to build a new ontology which serves at a higher level in the application than that of various ontologies in ontology libraries[19]
translation	ontology translation is required when translating data sets, generating ontology extensions, and querying through different ontologies[10]
Usage type (algorithm is:)	
applicable by human	approach can be used only by humans (human interaction indispensable)
applicable by machine	approach can be used by machine as a service
Adaption parameter (algorithm has been applied for:)	
number of domains	number of different domains the matching approach was applied for
number of applications	number of different applications the matching approach was applied for
number of tasks	number of different tasks the matching approach was applied for
reference of usage	has the approach been utilized by other users

Table 3. Usage characteristic

DIMENSION: OUTPUT CHARACTERISTIC	
Factor	**Description**
Output type	
deliver relations	the output of most matching systems is a set of the correspondences between attributes of schemas
deliver value	e.g. matcher used to determine the semantic similarity between concepts
deliver understandable	matcher delivers some explanations of the results
(for humans) results	
Matching Cardinality	
global 1:1	relationship cardinalities between matching elements w.r.t different mapping elements[26]
global n:1	
global 1:m	
global n:m	
local 1:1	relationship cardinalities between matching elements w.r.t an individual mapping element[26]
local n:1	
local 1:m	
local n:m	
Execution Completeness	
full match	considers all elements of the schemes
partial match	considers only some elements of the schemes
injective match	all elements of the domain are mapped to elements of the range
surjective match	all elements of the range are mapped to elements of the domain

Table 4. Output characteristic

2.5 Documentation Characteristic

Due to the fact that documentation is an essential part of every software product and in many ways it is even more important than the program code[18] the information about its quality and clarity can be significant for the selection of an approach. Furthermore, since one of the goals of documentation is to provide sufficient information so that an architecture can be analyzed for suitability to the purpose[4], it could be a determining coefficient for the selection of a particular algorithm, especially if the algorithm is to be reused in a different context from the domain or application it was originally developed for (cf. Tab. 5).

2.6 Cost Characteristic

The last dimension, *cost characteristic*, describes the financial factors regarding the (commercial) usage of a single matching approach like the matcher licence or the access to the appropriate matcher interface (cf. Tab. 6).

DIMENSION: DOCUMENTATION CHARACTERISTIC	
Factor	**Description**
quality of documentation	quality of the available documentation
clarity of documentation	clarity of the available documentation
clarity of maturity description	clarity of the description of the approach's maturity
availability of examples	are examples of the approach available

Table 5. Documentation characteristic

DIMENSION: COST CHARACTERISTIC	
Factor	**Description**
costs of matcher licence	the costs that have to be paid for the matcher licence
costs of matcher tool licence	the costs that have to be paid for the using of the tools matcher have been developed with
costs of access matcher interface	the costs that have to be paid for the using of interface

Table 6. Cost characteristic

3 A Method to Detect Suitable Matching Approaches

In the previous section we introduced the multilevel characteristic for matching approaches that provides a framework for matcher description. It can be used as a basic principle in the process of comparing different algorithms to determine an appropriate approach w.r.t the given circumstances.
As long as decisions rely on single criterion that serves as the basis for comparison of alternatives or the scales of the different criteria are consistent and numeric measures accurately capture expected performance, summary statistics or, in some cases, just acting on the human instinct may be sufficient for the decision making process. However, when the decision depends on multiple criteria and scales are not consistent the process becomes very complex and difficult, and the involvement of qualitative as well as quantitative methodologies or tools is indispensable. Consequently, in such cases a multi criteria decision making process is required, otherwise known as a *Multi Criteria Decision Analysis (MCDA)*, which is a procedure that aims to support decision makers whose problems are concerned with numerous and conflicting criteria. Such methods developed for better model decision scenarios vary in their mathematical rigor, validity, and design[15]. One of such method, a methodology for supporting a decision making process called *Analytic Hierarchy Process (AHP)* takes into account the considerations of Hahn[17] regarding the need for a structured results-based approach for decision making that allows trade-offs into the systematic method, including all perspectives and considerations. The AHP is a systematic approach developed to structure the expectance, intuition, and heuristics based decision making into a well-defined methodology on the basis of sound mathematical principles[1]. It helps to set priorities and to make the best decision when both qualitative and quantitative aspects of a decision need to be considered[27], i.e. AHP provides a mathematically rigorous application and proven process for prioritization and decision-making. By reducing complex decisions to a series of pair-wise comparisons and then synthesizing the results, decision-makers arrive at the best decision based on a clear rationale. It is generally accepted, that AHP constitutes one of the best options to aid multi-criteria decision making since it does not use the normalized groups of separate numbers which destroy the lineal relationship among them[11]. Instead it compares the relative importance that each

criterion has with respect to the others, while enabling the relative weight of the criteria to be calculated. Finally it normalizes the weights in order to obtain the measures for the existing alternatives. The AHP-method consists of:
STEP 1 - define the problem or the project objectives: e.g. buying a car;
STEP 2 - build a hierarchy of decision: AHP provides a means to break down the problem into a hierarchy of subproblems (hierarchy of goal, criteria, sub-criteria and alternatives) which can more easily be comprehended and subjectively evaluated[1]. At the root of the hierarchy is the goal (e.g. suitable car) or objectives of the problem in question, the leaf nodes are the alternatives (e.g. Mercedes, VW) which are to be compared and between these two levels are various criteria (c) and sub-criteria (sc) (e.g. c-car comfort: sc-air condition, sc-leather seat; c-car security: sc-ABS, sc-airbag and c-car body design)
STEP 3 - data collection; data is collected from domain experts corresponding to the hierarchical structure in the pairwise comparison of the alternatives on a qualitative scale. This step assesses the characteristics of each alternative (e.g. Alternative 1 (Mercedes) is much better then Alternative 2 (VW) w.r.t leather seats, airbag and car body design but Alternative 2 (VW) is better then Alternative 1 (Mercedes) considering ABS and air condition).
STEP 4 - build a pairwise comparison: for each level of criteria (sub-criteria and criteria) a pairwise comparison between the sibling nodes is to be built and organized into square matrix[4] (e.g. car security is much more important than car body design and more important than car comfort while car comfort is only a little bit more important than car body design).
STEP 5 - calculate the final result: the ratings of each alternative (cf. step 3) is multiplied by the weight of the sub-criteria (cf. step 4) and aggregated to get local ratings with respect to each criterion. The local ratings are then multiplied by the weights of the criteria (cf. step 4) and aggregated to the global ratings. The final value is used to make a decision about the problem defined in the step 1.

4 Applying AHP for the Matcher Selection

To allow a selection of matching approaches based on a mathematically rigorous method that provide a proven process for prioritization and decision-making the abovementioned process AHP is to be applied. By reducing complex decisions, i.e. which matching is suitable for a given set of requirements, to a series of pair-wise comparisons (dimensions, factors and attributes) and synthesizing the results (list of possible algorithms) decision-makers arrive at the best decision (the best matching approach) based on a clear rationale[27]. In the following we give a brief overview of how the AHP steps described in the Section 3 can be applied to the process of matcher selection taking into account some tool support for the data collection and calculation of the best alternative.
STEP 1: The problem to be solved: "Which matching approach is currently relevant w.r.t the given application requirements?"

[4] For details see[27]

STEP 2: The hierarchy of decision is built using the hierarchical tree described in Section 2 whereby the goal is to "find a suitable approach" (level 0) which is connected though three levels of criteria: 1^{st}level - dimensions, 2^{nd} level - factors and 3^{rd} level - attributes with the alternative matching approaches (cf. Fig. 2)

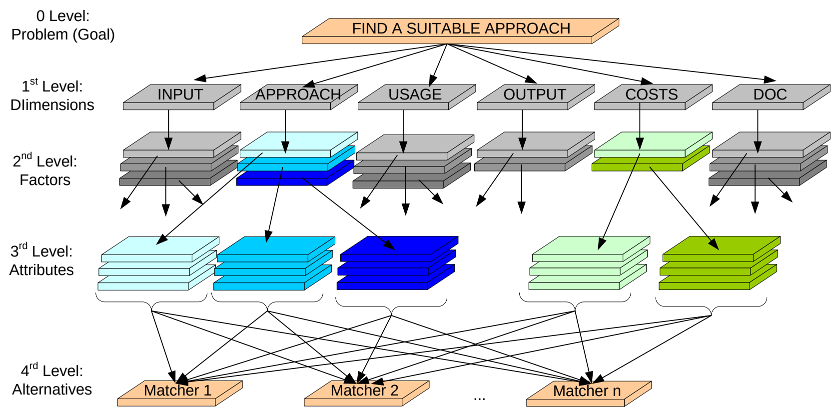

Fig. 2. AHP hierarchy structure (Detection of the suitable matching approach)

STEP 3: In order to collect data about the different alternatives of matching approaches and to be able to conduct the pairwise comparisons we firstly need the relevant information about the particular alternatives. For this reason we have developed (following the hierarchical structure of the matching characteristic) an online questionnaire (to be fill out by the domain and matching experts) that allows the addition and rating (by usage of a predefined scale from 0 to 8) of new matching alternatives. When a new matcher is added via the questionnaire into the collection of the alternatives, all available alternatives in the system are automatically weighted against the new approach. Given two matcher alternatives m_1, m_2 and criteria c as well as the user defined weighings for the single approach $w(c)_{m_1}$ and $w(c)_{m_2}$ the weighings for the pairwise comparisons (between alternatives m_1, m_2) $w(c)_{m_1,m_2}$ and $w(c)_{m_2,m_1}$ are calculated as follows: (i)$w(c)_{m_1,m_2} = w(c)_{m_1} - w(c)_{m_2}$; (ii)$w(c)_{m_2,m_1} = w(c)_{m_2} - w(c)_{m_1}$. PHPSurveyor[5] is used as the tool for providing the online questionnaire. The collected data regarding the matcher alternatives from the questionnaire is stored in a questionnaire database (MySQL) while an additional database (AHP database) stores the weighting results of the pairwise comparisons (cf. Fig. 3).

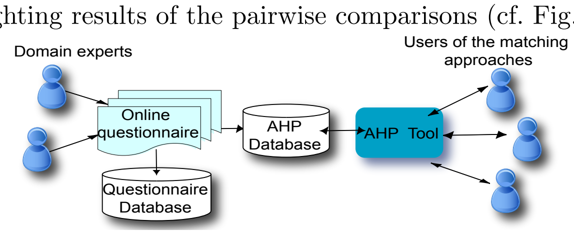

Fig. 3. AHP Tool with online questionnaire

[5] http://www.phpsurveyor.org

STEP 4: To enable a user-friendly pairwise comparison of the criteria from the multilevel hierarchy matcher characteristic we developed a tool which supports the processing of the AHP method[6]. Since the users of the AHP-tool have defined the requirements of their application w.r.t the suitable matching approach, they are able to weight the criteria (dimensions, factors and attributes) in the pairwise comparison on the scale from 0 - equal (two criteria have the same importance) to 8 - extremely important (one criteria is much more relevant than the other) concerning their system specification. This means, that for each level of criteria the users build a pairwise comparison between the sibling nodes: they weight the attributes against attributes, factors against factors and dimensions against dimensions (e.g. within the factor *formality level* the attribute *formal* (ontologies) is more important than *informal*, cf. Fig. 4).

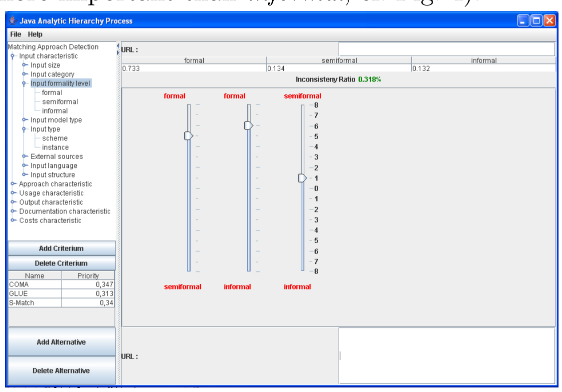

Fig. 4. AHP tool: weighed attributes

STEP 5: The decision regarding the determination of the suitable matching approach defined in the step 1 is based on the ranking $r(goal)$ of a matcher alternative m. The ranking reflects the global importance of the approach according to the alternative weightings performed in step 3 as well as criteria weightings from step 4 and is calculated as followed:

$c_{crit} = \{n | n \text{ child of } crit\}$

$$r(crit) = \begin{cases} |\text{getWeight}(m, crit)|, & \text{if crit is at lowest hierarchy level} \\ \sum_{n \in c_{crit}} r(n) \cdot |\text{getWeight}(m, n)|, & \text{otherwise} \end{cases}$$

The higher a matcher alterative m is weighted for various criteria, with each criteria weighted with respect to the users requirements, the higher the priority of the particular approach in the entire ranking. Following this weighting process the AHP tool supports the creation of a ranking of the alternatives in depending upon the multilevel hierarchy matcher characteristic, weightings of these characteristics as well as weightings of the alternatives that shows the priority of each alternative for the defined goal.

[6] AHP tool is a modification of the Java AHP tool JAHP; `http://www2.lifl.fr/~morge/software/JAHP.html`

5 Conclusion

In this paper we presented the adaption of the Analytic Hierarchy Process (AHP) to the process of detection of a suitable matching approach. The proposed strategy for the decision making based on *multilevel characteristic for matching approaches* and supported by the AHP tool enables e.g. domain experts with poor expertise in ontology matching field to find appropriate approach w.r.t their application requirements. The future work will be dedicated to the collection of further matcher alternatives (with help of the online questionnaire) and the application of the AHP tool into the various Semantic Web scenarios connected with the evaluation of the entire framework.
Acknowledgements: This work has been partially supported by the Knowledge Nets project, which is part of the InterVal- Berlin Research Centre for the Internet Economy, funded by the German Ministry of Research (BMBF) and by the EU Network of Excellence KnowledgeWeb (FP6-507482).

References

1. N. Bhushan and K. Rai, editors. *Strategic Decision Making: Applying the Analytic Hierarchy Process*. Springer, 2004.
2. C. Bizer, R. Heese, M. Mochol, R. Oldakowski, R. Tolksdorf, and R. Eckstein. The Impact of Semantic Web Technologies on Job Recruitment Processes. In *Proc. of the 7th Internationale Tagung Wirtschaftsinformatik 2005*, pages 1367–1383, 2005.
3. S. Castano, A. Ferrara, and S. Montanelli. Methods and Techniques for Ontology-based Semantic Interoperability in Networked Enterprise Contexts. In *Proc. of the 1st CAiSE INTEROP Workshop On Enterprise Modelling and Ontologies for Interoperability (EMOI - INTEROP 2004)*, pages 261–264, June 2004.
4. P. Clements, F. Bachman, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, and J. Stafford, editors. *Documenting Software Architectures: Views and Beyond*. Addison-Wesley Professional, 2002.
5. H. H. Do, S. Melnik, and E Rahm. Comparison of Schema Matching Evaluations. In *Proc. of GI-Workshop "Web and Databases"*, 2002.
6. H. H. Do and E. Rahm. COMA—a system for flexible combination of schema matching approaches. In *Proc. of the 28th VLDB Conference*, 2002.
7. A. Doan, P. Domingos, and A. Halevy. Reconciling Schemas of disparate Data sources: A Machine Learning Approach. In *Proc. of the the SIGMOD01*, 2001.
8. A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Ontology Matching: A Machine Learning Approach. *Handbook on Ontologies*, pages 385–516, 2004.
9. A. Dogac, G. Laleci, Y. Kabak, and I. Cingil. Exploitingweb service semantics: Taxonomies vs. ontologies. *IEEE DATA ENGINEERING BULLETIN*, 4, 2002.
10. D. Dou, D. McDermott, and P. Qi. Ontology Translation on the Semantic Web. In *Proc. of the AInt'l Conf. on Ontologies, Databases and Applications of Semantics (ODBASE2003)*, pages 952–969, 2003.
11. N. Fenton and L. Pfleeger, editors. *Software Metrics, A Rigorous & Practical Approach*. International Thomson Cmputer Press, 1996.
12. F. Frst and F. Trichet. Axiom-based ontology matching. In *Proc. of the 3rd international conference on Knowledge capture (K-CAP'05)*, pages 195–196, New York, NY, USA, 2005. ACM Press.

13. J. Garbers, M. Niemann, and M. Mochol. A personalized hotel selection engine. In *Proc. of the Poster Session of 3rd ESWC 2006*, 2006.
14. F. Giuchiglia and P. Shvaiko. Semantic Matching. *Knowledge Web Review Journal*, pages 265–280, 2004.
15. J. R. Grandzol. Improving the faculty selection process in highe education: A case for the analytic hierarchy process. *Using Advenced Tools, Tehciques, and Methodlogies. Association for Institutional Research*, 6, 2005.
16. T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6):907–928, 1995.
17. E. D. Hahn. Better decisions come from a results-based approach. *Marketing News*, 36(36):22–24, 2002.
18. W. S. Humphrey, editor. *Introduction to the Team Software Process*. Addison-Wesley Professional, 1999.
19. L. Li, B. Wu, and Y. Yang. Agent-based Ontology Integration for Ontology-based Applications. In *Proc. of the Australasian Ontology Workshop (AOW 2005)*, volume 58, pages 53–59, 2005.
20. A. Lozano-Tello and A. Gomez-Perez. ONTOMETRIC: A Method to Choose the Appropriate Ontology. *Journal of Database Management*, 15:1–18, 2004.
21. J. Madhavan, P. A. Bernstein, and E. Rham. Generic Schema Matching with Cupid. In *PROC. of the 27th VLDB Conference*, 2001.
22. S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. In *Proc. of the 18th International Conference on Data Engineering (ICDE02)*, 2002.
23. M. Mochol and E. Paslaru B. Simperl. Practical guidelines for building semantic erecruitment applications. In *Proc. of the International Conference on Knowledge Management (iKnow'06), Special Track: Advanced Semantic Technologies*, 2006.
24. E. Paslaru Bontas and M. Mochol. Towards a reuse-oriented methodology for ontology engineering. In *Proc. of 7th International Conference on Terminology and Knowledge Engineering (TKE 2005)*, 2005.
25. E. Paslaru Bontas, M. Mochol, and R. Tolksdorf. Case Studies on Ontology Reuse. In *Proc. of the 5th International Conference on Knowledge Management*, 2005.
26. E. Rham and P. A. Bernstein. A survey of approaches to automatic schema matching. *Journal of Very Large Data Bases*, 2001.
27. T. L. Saatly. How to Make a Decision: The Analytic Hierarchy Process. *European Journal of Operational Research*, (48):9–26, 1990.
28. P. Shvaiko. A Classification of Schema-Based Matching Approaches. Technical Report DIT-04-09, University of Trento, `http://eprints.biblio.unitn.it/archive/00000654/01/093.pdf`, December 2004.
29. P. Shvaiko. Iterative schema-besed semantic matching. Technical Report DIT-04-020, University of Trento, `http://eprints.biblio.unitn.it/archive/00000550/01/020.pdf`, June 2004.
30. P. Shvaiko and J. Euzenat. A Survey of Schema-Based Matching Approaches. *Journal on Data Semantics*, 4:146–171, 2005.
31. G. Stumme and M. Alexander. FCA-MERGE: Bottom-up merging of ontologies. In *Proc. of the 17th IJCAI 2001*, pages 225–230, 2001.
32. M. Uschold and M. Grninger. ONTOLOGIES: Principles, Methods and Applications. *Knowledge Engineering Review*, 11(2), 1996.
33. M. Uschold and R. Jasper. A Framework for Understanding and Classifying Ontology Applications, 1999.
34. C.J. van Rijsbergen, editor. *Information retrieval, 2nd edition*. ButterworthsLondon, 1979.