Journal of Web Engineering, Vol. 12, No.5 (2013) 422-438

A FRAMEWORK FOR DETECTING AND REMOVING KNOWLEDGE OVERLAPS IN A COLLABORATIVE ENVIRONMENT: CASE OF STUDY A COMPUTER CONFIGURATION PROBLEM

MARIA VARGAS-VERA

Facultad de Ingenieria y Ciencias, Centro de Investigaciones en Informatica

Universidad Adolfo Ibanez, Vinia del Mar, Chile
maria.vargas-vera@uai.cl

MIKLOS NAGY

Knowledge Media Institute, Open University

Walton Hall, Milton Keynes, MK7 6AA, UK
M.Nagy@open.ac.uk

PATRICIA ORDONEZ DE PABLOS

Faculty of Economics, Oviedo University, Spain
patriop@uniovi.es

Received February 20, 2013
Revised March 25, 2013

This paper presents a framework for knowledge integration based on mappings between similar concepts in constraint graphs associated to a configuration problem. In particular, the paper deals with one of the problems which could arise when performing collaborative knowledge integration, namely detecting knowledge overlaps. The solution to the overlapping problem relies on the use of matching algorithms embedded in DSSim (short for Dempster-Shafer Similarity). To illustrate the approach, a case study of a computer configuration problem is presented. The solution to the knowledge overlap problem is important as it has the promise to become an alternative approach for the current knowledge integration solutions. Through our approach the real cost of integration can be reduced as it is not necessary to invest a great amount of resources beforehand a truly integrated system can be operational.

Key words: Knowledge-based systems, Knowledge integration, Configuration

1 Introduction

The term "Knowledge Integration" has different meanings. The definitions found in the fields of Artificial Intelligence, Ontologies, Databases and Knowledge Management vary strongly. For example, in the Artificial Intelligence community, Knowledge Integration is seen as the process of integrating knowledge into an existent body of knowledge [20]. Knowledge integration refers to the identification of how new and prior knowledge interacts while incorporating new information into a knowledge base. In contrast, the view taken from Knowledge Management is that knowledge integration is a fundamental management practice. According to Grant [11], the organization's primary concern is the integration of its dispersed knowledge resources in order to apply them to a "production of a new artifact'' as a mean of creating new knowledge out of novel combinations of existing knowledge However, in our opinion knowledge integration is the process of creating a unified knowledge model by means of integrating individual models made by different knowledge engineers. This integration is basically a reconciliation of the terms and relations used by each knowledge engineer while building their own model.

The motivating scenario of our work is the assumption that large knowledge bases are typically constructed by different knowledge engineers or domain experts in an incremental and collaborative way. The development of a knowledge base for a product configuration system [25] is a typical example as different organizational units contribute technical and process or marketing-related constraints on legal product constellations. The problem is even harder, when the configurable product is delivered by multiple providers in a supply-chain [1], and requires the cross-company integration of knowledge bases and interfaces.

The main contribution of this paper is to propose as a solution to the overlapping problem based on matching algorithms which use Dempster-Shafer and Fuzzy Voting Model. A scenario to illustrate the knowledge integration using DSSim best methods is outlined in our case of study.

The rest of the paper is organized as follows. Section 2 provides an overview of related work. Section 3 presents a case scenario that illustrates the overlapping problem when performing knowledge integration. Section 4 presents an evaluation of our methodology to knowledge integration. Finally, in Section 5 we present our conclusions and describe our future work.

2 Related Work

Information Integration has been investigated by several research communities in Computer Science. After an analysis of the literature, four perspectives have been identified. These perspectives are Knowledge Based Systems, Ontologies, Databases and Knowledge Management. The first perspective deals with problems in knowledge modeling in particular in expert systems. The second perspective is the work in the Ontologies field ranging from ontology merging to alignment. The third perspective, Databases, is more related to data integration which consists of providing a unified view on the data stored in different databases with different models. Finally, the fourth perspective Knowledge Management is not explored in too much detail as is not the main focus of this paper.

2.1 Expert Systems view

Murray [19] presents an approach to knowledge integration as a machine learning task. He implemented a system called REACT which is a computational model that identifies three activities. (1) ``Elaboration": (2) ``Recognition", and (3) `Adaptation". In particular "Adaptation'' exploits the learning opportunities by modifying the new and prior knowledge. A learning opportunity occurs when a property of a particular object in the learning context can be generalized into a property for every instance of a class of objects. Empirical evidence indicates that indeed, knowledge integration helps knowledge engineers to integrate new information into a large knowledge base.

Knowledge Integration has become an essential element in the Semantic Web Community. For example, knowledge integrations allows to access services which offer knowledge contained in various distributed databases associated with semantically described web portals. In this context, Zygmunt et al., propose a framework for knowledge integration supported by using an agent-based architecture [30]. The approach relies very much on the integration of ontologies by the gradeAgent which estimates the similarity between classes and properties in the ontology. The approach uses algorithms of lexical and structural comparison. The checking of similarity between larger parts of a graph is performed with the use of Similarity Flooding algorithm. The approach also applied additional techniques based on a thesaurus when looking for synonyms and on the use of high level ontology to adjust concepts from the ontology to a given set of concepts which identify important notions. The framework does not handle uncertainty in the similarity metrics. In principle, it seemed as a good solution but in real scenarios the notion of uncertainty limited to a crisp mappings made a strong limitation in a proper identification of matching concepts and properties.

2.2 Ontologies View

The knowledge engineering community uses ontologies as the main approach for resolving semantic differences in heterogeneous data sources. Based on this approach several categories can be identified to Data Integration. One of them is to create a global ontology. In this manner, all the different sources share the same ontology in order to make the information integration possible. These solutions fit well when the number of sources is limited and a consensus can be achieved between partners. However, for real life scenarios, this solution is inflexible in nature and is not considered as a viable alternative in the context of knowledge integration.

Ontology merging aims to achieve semantic integration through merging different source ontologies into a consistent union of the source ontologies. Examples of merging systems are described as follows: FCAMERGE [10] offers a global structural approach to the merging process. It takes the source ontologies and extracts instances from a given set of domain-specific text documents by applying natural language processing techniques. Based on the extracted instances the system applies formal concept analysis techniques to derive a lattice of concepts as a structural result of merge process. The produced result is explored and transformed to the merged ontology by the ontology engineer. PROMPT [8] makes initial suggestions based on linguistic similarity between class names then performs automatic updates, finds new conflicts and makes new suggestions.

Ontology mapping aims to achieve semantic integration through the creation of mappings between concepts attributes etc. between two ontology entities. Based on database schema integration solutions a wide range of techniques has been proposed from manually defined rules to semi-automatic approaches that make use of machine learning, heuristics, natural language processing and graph matching algorithms. Some mapping systems are presented below.

MAFRA [17] supports an interactive, incremental and dynamic ontology mapping process in the Semantic Web context. The main contribution of this approach is that it creates a true distributed ontology mapping framework that is different from mediator based approach.

GLUE [7] evolved from a mediator based LSD [6] data source schema matching, applies machine learning techniques and similarity measures based on joint probabilistic distributions. In short, numerous ontology mapping systems have been proposed but only a handful of them have participated in the Ontology Alignment Initiative (OAEI) evaluation, which serves as a comparison benchmark for mapping systems.

Mapping systems which have participated in the OAEI evaluation are described as follows:

ASMOV) [15] carries out the mapping in two phases. In the first phase, different similarity measures are calculated and combined in order to establish preliminary mapping pairs. In the second phase the system carries out a semantic verification, in order to detect semantically inconsistent mappings and their causes.

RiMOM [29] uses the combination of different strategies. The strategies are selected based on the characteristics of the source ontologies and the pre-defined rules.

Anchor-Flood [26] has been developed in the context of International Patent Classification (IPC) in order to exploit the available taxonomy of related terms found in an abstract and aligns it with the taxonomy of IPC ontology. The mapping is done in two phases. The first phase is the ontology mapping, where the concepts and properties in the different ontologies are aligned. The second phase is the mapping of the instances of the ontologies.

TaxoMap [13] was designed to support information integration between different sources. The mapping process is oriented from ontologies that describe external resources (named source ontology) to the ontology (named target ontology) of different web portals. TaxoMap heavily relies on the labels it uses a moropho-syntactic analysis for tagging text with part-of-speech and lemma information and a similarity measure which compares the trigraph of the concept labels.

Lily [28] employs hybrid matching strategies to create the mappings for both normal and large scale ontologies.

2.3 Databases view

In the database community several solutions have been proposed. However, not all approaches [3] have been implemented in real life applications. The characteristics of these approaches are that they all have inputs and outputs, which are supplied or processed by a human designer. The inputs are usually the domain models including entity relationships, views and sometimes queries whereas the outputs are conceptual models, global schemas, mapping rules or conflicts. The majority of approaches based on mediator architecture that involve logical database schemas, which are used as shared mediated views over the queried schemas. A number of systems have been proposed e.g. TSIMMIS [9], Information Manifold [16], InfoSleuth [4], MOMIS [5], LSD [6] that shows the flexibility and the scalability of these approaches. In particular, MOMIS is focused a data integration from scientific data sources but it also been applied to other domains like building a tourism information provider [5]. The problem, however, is that these solutions rely on the initial idea of database schema integrations namely to create a global view, which will be used as a mediator between the different sources [12].

2.4 Knowledge Management view

Hung [14] presents an empirical study that investigates the patterns of knowledge integration in the collaborative development of system on a chip (SoC) by semiconductor firms. The study focused on the central interactive process for engineering applications and experimental practice to enhance knowledge integration and technology innovation for rapid product development. A process model for knowledge integration via experimental practice is presented; further explanation can be found in [14]. The process of knowledge integration is triggered by new requirements i.e. new product features or testing methods, which cannot be resolved based on the current knowledge. This integration process depends upon knowledge already existing in the organization as well as new external knowledge. The outcome of the process is a technological innovation and the fact that the knowledge of the organization is enhanced by means of knowledge integration. The Knowledge Management perspective which appears related to our work is the one based on the Distributed Knowledge Management (DKM) approach explored in the Knowledge Management community [18], in which subjective and social aspects of the real world are taken into account. However, this perspective is not going to be explored as is out of the scope of this paper.

3 Case Study: Computer Configuration

Our case of study is a restricted version of a computer configuration problem. The problem of configuration is defined as a CSP (Constraint Satisfaction Problem) problem where a model using Variables (Table 1), Domain for the variables (Table 2) and Constraints over the variables (Table 3) is defined. The main goal is to obtain an assignment (i.e. a value for all the variables).

Table 1 Variables

V1	OS
V2	Memory
V3	Hard_disk_size
V4	CPU
V5	Monitor
V6	Mouse
V7	Video_card
V8	Graphics_card
V9	Gaming_PC
V10	Keyboard
V11	Monitor_resolution

To illustrate the approach, the initial constraints graphs built by different engineers using different knowledge models have been selected. These original graphs hold by our individual departments are depicted in Figure 1 and Figure 2. These graphs use standard computer jargon although; they have discrepancies on the name of variables used. The term Video_card and Graphics_card (variable names) were used by different knowledge engineers to refer to the same concept. The latest problem suggested that in order to perform knowledge integration, one has to perform mappings between nodes in the constraints graphs. For the sake of clarity, only overlaps are presented in one node of the graph but this is not always the case. Figure 2 uses as variable name called Video_card whilst in Figure 1 the variable name is Graphics_card.

Table 2 Domain

OS	Vista, XP, MAC-OS, Windows 7,Linux
Memory	512 MB, 1024 MB, 2048 MB, 3072 MB
Hard_disk_size	160 GB, 180 GB, 320 GB
CPU	Pentium 4, Intel Centrino
Monitor	14 inches,18 inches,19 inches,20 inches
Mouse	Logitech, Magic mouse
Video_card	NVIDIA 600series, NVIDIA 700series, NVIDIA 800series
Graphics_card	GeForce 7600series, GeForce 7800series, GeForce 7900series
Gaming_PC	yes,no
Keyboard	Win keyboard,Mac Keyboard
Monitor_resolution	low,medium,high

Table 3 Constraints

C1	*IF OS* = "*XP*'' '*THEN Memory* ≤ 2048 *MB*
C2	*IF Monitor* = "20 *inches*" *THEN Graphics card* = "*GeForce* 7800 *series*"
C3	*IF OS* = "*XP*" *THEN CPU* = "*Pentium* 4"
C4	*IF OS* = "*Vista*" *THEN CPU* = "*Pentium* 4"
C5	*IF OS* = "*XP*" *THEN Hard disk size* ≥ "500 *MB*"
C6	*IF Gaming PC* = "*yes*" *THEN Graphics card* = "*NV IDIA* 8000*series*''
C7	*IF Gaming PC* = "*yes*" *THEN Memory* ≥ "2048"
C8	*IF Gaming PC* = "*yes*" *THEN Hard disk size* ≥ "160*GB*"
C9	*IF Monitor* ≥ 20*inches THEN Monitor_resolution* = "*high*"
C10	*IF OS* = "*MAC* − *OS*" *THEN Keyword* = "*Mac Keyboard*"

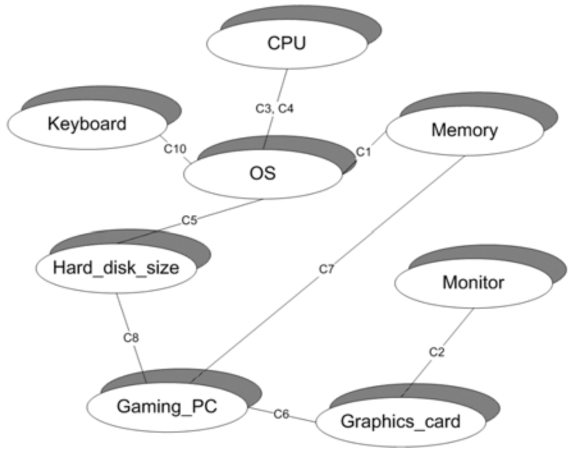

Figure 1 A Constraint graph for the computer configuration problem using variable Graphics card

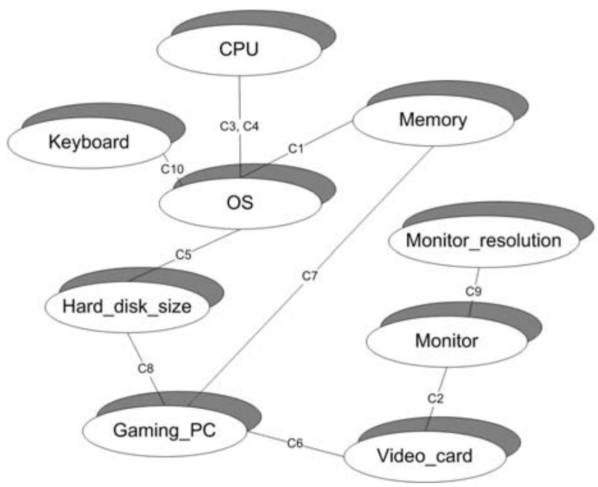

Figure 2 A Constraint graph using variable Video card

A unified view of two constraints graphs was produced (manually) by joining two initial constraints graphs. This unified view is depicted in Figure 3.

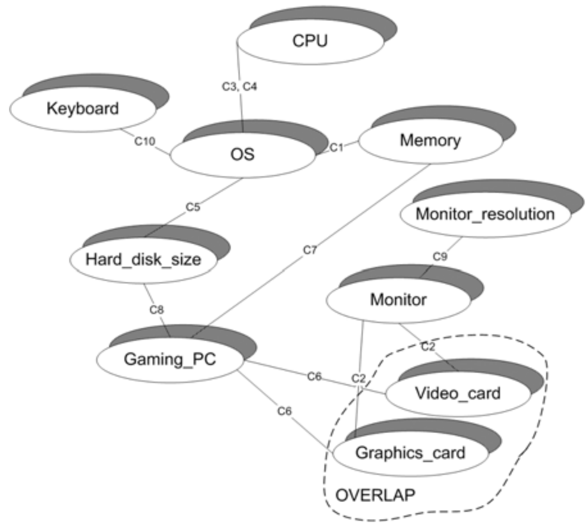

Figure 3 A Constraint graph for the computer configuration problem with 9 variables

4 Mapping Process

The objective of the ontology mapping is to use different similarity measures in order to establish the mappings. However, in practice one similarity measure or some technique can perform particularly well for one pair of concepts or properties and particularly badly for another pair of concepts or properties, which has to be considered in any mapping algorithm. In the proposed ontology-mapping approach different software agents are used where each agent carries only partial knowledge of the domain and can observe it from its own perspective where available prior knowledge is generally uncertain. Our main argument is that knowledge cannot be viewed as a simple conceptualization of the world, but it has to represent some degree of interpretation. Such interpretation depends on the context of the entities involved in the process. In order to represent these subjective probabilities in the proposed system the Dempster-Shafer theory of evidence [27] is used, which provides a mechanism for modeling and reasoning uncertain information in a numerical way, particularly when it is not possible to assign belief to a single element of a set of variables. Furthermore, our proposed solution involves consultation of background knowledge, assessment of similarities, resolving conflicts between the assessments and finally the selection of possible mappings i.e. items that are named differently but are the same in practice. As an example, consider that one needs to determine that the “Video_card’’ is equivalent to the “Graphics_card’’. For this example, hypothesis (H) is that these items are equivalent but one needs to find evidences that support or contradict the initial hypothesis. In this case, several hypotheses are created, comparing each element of the constraint graph to each other. For example, consider that the following three hypotheses selected from all available ones:

H1(equivalent) = {video_card} = {graphics_card}

H2(equivalent) = {video card} = {mouse}

Hn(equivalent) = {video card} = {term_n}

Furthermore, it is advisable that during the similarity assessment different similarity algorithms are used, i.e., different agents that are specialized in a particular similarity assessment. Since the hierarchy of the constraint graph cannot be exploited for similarity assessment the only way is to utilize the nodes in order to detect the mappings. As such consider that three agents using different string similarity measures are defined. The steps to produce the mappings are as follows:

Step 1 consult background knowledge: In this step, using general background knowledge, e.g., WordNet the system tries to determine the meaning of the terms. This case is specialized as the computer shop only sells electronics therefore, other meanings e.g. art context of graphics can be excluded from the process. After consulting background knowledge one can extend the initial terms using sister terms and direct hypernyms with the following computer science related terms:

Video card = {videodisplay; graphics; picture; graph}

Graphics card = {picture; movie; video; image; visual representation}

Mouse = {trackball; rotatableball; cursor control device}

Step 2 similarity assessments: Using different string similarities e.g. Jaccard, Jaro-Winkler, Monge-Elkan mapping agents have established that

Agent1 : H1(mapping) = 0.80;H2(mapping) = 0.3

Agent2 : H1(mapping) = 0.72;H2(mapping) = 0.2

Agent3 : H1(mapping) = 0.64;H2(mapping) = 0.2

After belief assessments have been performed, mapping agents can establish that H1 is the preferred choice between the available hypotheses and that H2 does not contain contradictory beliefs. However, H1 contains contradictions because Agent 2 belief does not support sufficiently that H1 can be selected. The different strategies for selecting the contradicting belief is out of the scope of this paper but for the presented scenario the rule of thumb is that in an ordered list of beliefs at least 2 agents should have the same belief otherwise there is a contradiction.

In our framework all the numerical values represent the belief mass function that each agent can deduce from the similarity calculations. The represented beliefs are the interpretation of each agent and such they are subjective. Once the beliefs in similarities have been established agents need to select the hypothesis with the highest belief. In our example, this corresponds to the H1 namely that the ``Video_card'' and ``Graphics_ card'' could be similar. Before the mapping is selected the system needs to verify that the original beliefs are not contradicting.

Step 3 verification and resolution of contradictions: it is important to point out that our proposed approach does not utilize thresholds for defining what is contradicting or not. Our solution makes use of comparisons between each agent's belief and eliminates the one that can be contradictory with the majority of the beliefs. The strategies for selecting, which agent should start evaluating trust is a complex issue and is out of the scope of this paper. However, in the presented scenario a basic rule was that the system tries to establish similar beliefs of at least two agents. Therefore, the beliefs in similarities need to be ordered and the agent whose belief function value is the smallest (smaller than the highest and greater than the smaller) will start to the trust evaluation process. In our example, Agent 2 is in the position of detecting such contradiction as both Agent 1 and Agent 3 has different belief on the similarity. The question in this case is to trust Agent 1 and support that ``Video_card" and ``Graphics_card" is equivalent or trust Agent 3 whose belief is lower and probably discharge the mapping.

In order to resolve the contradiction a fuzzy voting model [2] is used. This is because the different beliefs in similarity can be resolved if the mapping algorithm can produce an agreed solution, even though, the individual opinions about the available alternatives may vary. A solution for reaching this agreement is proposed by evaluating trust between established beliefs through voting, which is a general method of reconciling differences. Voting is a mechanism where the opinions from a set of votes are evaluated in order to select the alternatives that best represent the collective preferences. Unfortunately, deriving binary trust like trustful or not trustful from the difference of belief functions is not so straightforward since the different voters express their opinion as subjective probability over the similarities. For a particular mapping this always involves a certain degree of vagueness hence the threshold between the trust and distrust cannot be set definitely for all cases that can occur during the process. Additionally, there is no clear transition between characterizing a particular belief highly or less trustful. Therefore, our argument is that the trust membership or belief difference values, which are expressed by different voters, can be modeled properly by using fuzzy representation. Before each agent evaluates the trust in other agent's belief over the correctness of the mapping it calculates the difference between its own and the other agent's belief. Depending on the difference it can choose the available trust levels e.g. if the difference in beliefs is 0.08 (belief of Agent 2 - Agent 3 and belief of Agent 3 - Agent 2) then the available trust level can be high and medium.

4 Evaluation

In order to evaluate the proosed approach, experiments have been carried out (process is depicted in Figure 4) using two ontologies that were created from two on line PC (Personal Computers) store. Both ontologies contain categories and instances of items that are sold in the on-line shop. The main objective of our experiment was to evaluate how accurate our knowledge integration approach is. During the experiments 100 random configurations have been generated that simulate a customer choice and the correctness of the configurations have been evaluated after the two ontologies were mapped into each other.

The main idea of our experiment was to show the integration of our sample ontologies and then to use the integrated model for solving a computers configuration problem. The evaluation was performed in two parts. In the first part, two knowledge models (i.e. ontologies) were integrated from two online PC shops. This evaluation comprises to perform mapping between classes and properties of the two ontologies. In this task the DSSim system [22] has been used, which is a mapping system based on Dempster-Shafer Theory described in detail in [21] [22] [23] [24].

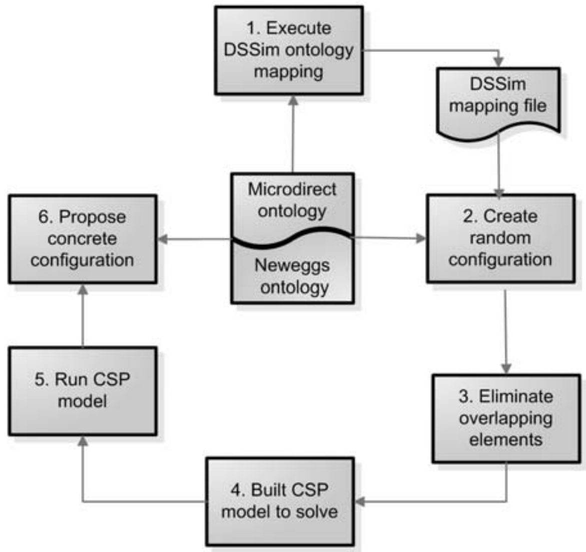

Figure 4 Experimental process

The second part of the experiment is the solution of the CSP problem using the mappings generated in the first phase. To illustrate a solution, PC configuration (basic configuration) has been used. The other two configurations (medium and expensive) were solved in a similar fashion. Our solution used a constraint solver called choco which is widely used in the CSP community. The notion of basic, medium and expensive configurations has been represented with the number of components assuming that the more expensive a configuration is the more components the configuration will contain. In our experiment the basic configuration has 30, the medium has 50 and the expensive has 70 components.

Summing up, experiments that have been carried out were based on the computer configuration problem as described in the section 3. In order to make it as close to real situation as possible two ontologies based on two online computer shops have been created that sell a wide variety of PC Components and Accessories. One shop is the Micro Direct Ltd from the UK and the second is Newegg from the US. For the experiments, ontologies that contain only partial component list from both sites have been created. The number of classes, properties and instances included in the ontologies are described on Table 4.

Table 4. Example ontology complexities

	Microdirect.co.uk	Newegg.com
Classes	102	121
Properties	47	46
Individuals	197	242
Subclass axioms	96	118
Equivalent classes axioms	19	5

4.1. Mapping quality

The first step of or experiment is to create a mapping file using DSSim in order to detect overlapping elements from the two ontologies. The idea behind our scenario and experiments is to integrate two data sources through ontology mapping. In practice, this means that our solution should make it possible to create configurations from two different shops without physically integrating the databases. The mapping file generated by our algorithm contains 93 mappings. These mappings range from the very obvious to hidden correspondences between concepts and properties e.g. Memory - Memory, ATi_Graphics_Card, Video_card. In addition, manually a mapping file has been created between the ontologies (Gold standard) in order to compare with the one that is generated by the system. This evaluation was measured using recall and precision, which are standard measurements from the Information Retrieval community.

Table 5. Mapping quality

	Value
Precision	0.66
Recall	1.0

Based on the result (depicted in Table 5) we can conclude that the recall rate is 100% .Therefore, all the possible mappings have been found by the system. However, the precision is 66%, which indicates that some additional mappings were found and they are incorrect. The precision rate is high and indeed, the manual mapping has resulted in the mapping file that contains only the equivalence relationships e.g. CPU - CPU between items. Our algorithm also identified not equivalence relations e.g. Motherboards - Server_Motherboard and this decreases the precision of the system.

4.2. Configuration quality

In the second experiment random configurations were created using components from both shops i.e. ontologies. For example, the memory from Microdirect and the Monitor from Neweggs are selected. The number of components can range between 30 and 70 depending on the configuration type.

In the following description, we made use of the numbering in the boxes in Figure 4, when referring to steps.

In step 3 using the mapping file (created in step 1) the overlapping components are eliminated from the configuration. For example, if Video_card was selected from ontology 1 and Graphics_card was also selected to the configuration the system leaves only one of them in the configuration.

During step 4 the available prices are taken for each component in the configuration. In practice, the system takes all instances of each component and add them as variables for the CSP problem. For example, for the Video card the system takes Sapphire_Radeon_HD_5850_1GB or

XFX_ATI_RADEON_4650. All these variables will feed into the CSP solver engine as textual variables.

In step 5 the CSP solver is executed in order to get what are the amounts that one can spend on each component in order to produce the suggested configuration. Given the fact that there is no guarantee that the CSP problem can be resolved in a timely manner 10 second constraint has been set on the choco solver in order to limit the available time for each experiment. In case the solver cannot find an optimal solution the random configuration will be returned.

In Step 6 the concrete components are selected that fit into the maximum amount that one can spend on each component.

The process from step 2 to 6 is repeated 100 times in order to obtain reasonable amount of data that can be analyzed. We are interested in measuring how well our proposed approach can perform in order to integrate knowledge from different sources. We measure how often overlapping elements are removed from random configurations and how often overlapping items have to be evaluated from the random configurations.

The experimental results are depicted in Figure 5.

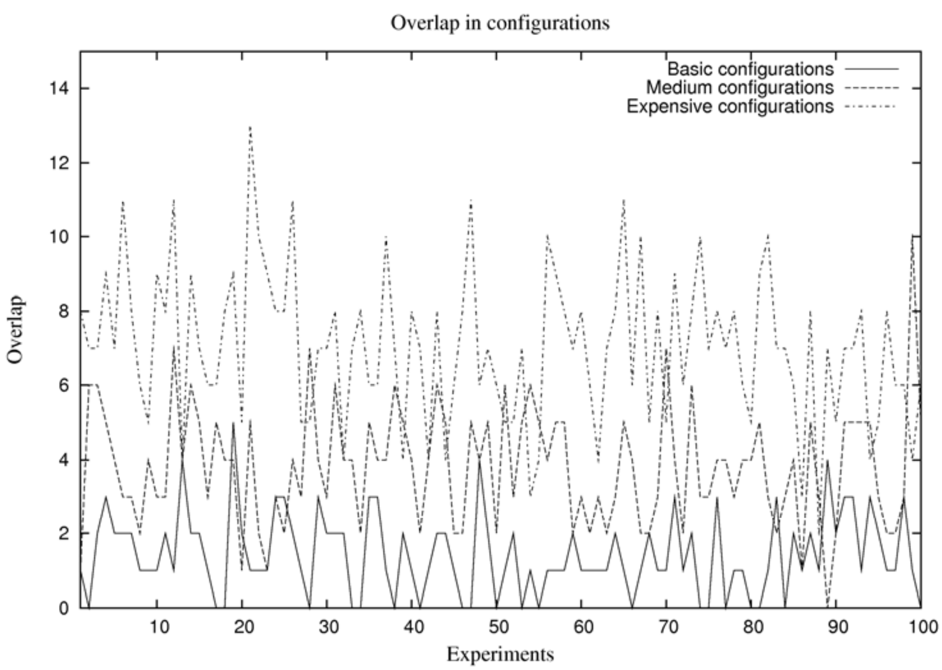

Figure 5 Overlapping components

As depicted in Figure 5 and Table 6 the number of overlapping elements per configuration varies from 0-13. According to the experiments 79% of the basic configuration, 99% of the medium configuration 100% for the expensive configuration represents cases where the overlapping elements have to be removed. In practice it means the knowledge integration occurs between 79-100 % percent of the cases. This is remarkable and in operational systems where users are involved this represents a considerable percentage. Based on our experiments, it can be established that knowledge integration can improve the PC configuration precision in the majority of the cases.

Table 6. Overlapping component statistics

	Min overlap	Max overlap	Average overlap	No overlap
Basic configuration	0	5	1.45	21
Medium configuration	0	10	3.83	1
Expensive configuration	2	13	7	0

Our experiments have showed that the result of constraint satisfaction problem for the PC configuration improves if the number of components for the configuration increases. This can be explained with the fact that with the more complex a configuration is the more overlapping in the CSP graph can occur. This is encouraging as our main objective is to establish a solution for the knowledge integration problem.

5 Conclusions and future work

This paper presented an approach to knowledge integration of several knowledge models. These knowledge models were created by different stakeholders. As a case of study to demonstrate the approach a restricted version of the computer configuration problem was introduced. The presented case of study was modelled as a Constraint Satisfaction Problem and the constraints graphs were produced. The detection of overlapping pieces of knowledge and its solution was performed by means of DSSim, an agent-based system which uses similarity algorithms coupled with a fuzzy voting model. The experiment shown was performed using two knowledge models and it was divided in two phases. The first phase was detection of overlapping knowledge and correction using our DSSim system. The second phase is the Constraint Satisfaction Problem using choco. Our preliminary findings are encouraging and they are the baseline for assessing the usefulness of our Knowledge Integration. Of course, more work needs to be done in order to fulfil our expectations of a generic framework for Knowledge Integration. Future work comprises to carry out experiments using more knowledge models.

A set of initial experiments and measures have been established that combine ontology mapping and constraint satisfaction in a real word scenario. Our pro- posed experimental context for knowledge integration is the logical federation of two on-line PC stores, without physically creating a unified database. The federation is carried out only the overlapping elements of the two different data sources in order to being able to eliminate the number of equivalent components for the proposed configuration. Our ontologies used during the experiment contain only a fraction of the information that can be extracted from the two on line stores. Nevertheless, our results are encouraging since even these relatively small ontologies produce 79-100% of overlaps in the configurations. The more elements one includes in the ontologies, the higher overlapping components will emerge in these configurations. Therefore, based on the current experiments we can conclude that the knowledge integration can occur in the majority of the cases and such approach can improve the overall situation of the system. However, in the future we intend to investigate further what influences the number of overlapping elements that occur in random configurations. In terms of constraint satisfaction our experiments have showed that only the expensive configuration performs well as the medium and basic contains far too much configuration that do not match the users criteria. One explanation is the limited number of instances in the two ontologies. We expect that the more instances will be includes into the ontologies (i.e. more PC components) the better our constraint can be met for the basic and medium configuration. In general, the experiments have showed that the proposed approach is promising. However, it requires more experiments with larger ontologies in order to further assess the strengths and weaknesses of our approach.

References

1. Ardissono, L., Felfernig, A., Felfernig, E., Friedrich, G., Goy, A., Jannach, D., Petrone, G., Schafer, R., Zanker, M.: A framework for the development of personalized, distributed web-based configuration systems. AI Magazine 24, 93-108 (2003)
2. Baldwin, J.F.: Mass assignment fundamentals for computing with words. In: IJCAI'97: Selected and Invited Papers from the Workshopon Fuzzy Logic in Artificial Intelligence. pp. 22-44. Springer-Verlag, London, UK (1999)
3. Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. ACM Computing Surveys 18(4), 323-364 (1986)
4. Bayardo, Jr., R.J., Bohrer, W., Brice, R., Cichocki, A., Fowler, J., Helal, A., Kashyap, V., Ksiezyk, T., Martin, G., Nodine, M., Rashid, M., Rusinkiewicz, M.,Shea, R., Unnikrishnan, C., Unruh, A., Woelk, D.: Infosleuth: agent-based semantic integration of information in open and dynamic environments. In: SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data. pp. 195-206. ACM, New York, NY, USA (1997)
5. Beneventano, D., Bergamaschi, S., Guerra, F., Vincini, M.: The momis approach to information integration. In: ICEIS. pp. 194-198 (2001)
6. Doan, A., Domingos, P., Halevy, A.: Reconciling schemas of disparate data sources: A machine-learning approach. In: In SIGMOD Conference. pp. 509-520 (2001)
7. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to map between ontologies on the semantic web. In: WWW '02: Proceedings of the 11th international conference on World Wide Web. pp. 662-673. ACM, New York, NY, USA (2002)
8. Fridman, N.N., Musen, M.: Prompt: Algorithm and tool for automated ontology merging and alignment. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and

Twelfth Conference on Innovative Applications of Artificial Intelligence. pp. 450-455. AAAI Press The MIT Press (2000)
9. Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Sagiv, Y., Ullman, J., Vassalos, V., Widom, J.: The tsimmis approach to mediation: Data models and languages. Journal of Intelligent Information Systems 8, 117-132 (1997)
10. Gerd, S., Alexander, M.: Fca-merge: bottom-up merging of ontologies. In: IJCAI'01: Proceedings of the 17th international joint conference on Artificial intelligence. pp. 225-230. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)
11. Grant, R.M.: Toward a knowledge-based theory of the rm. Strategic Management Journal 17, 109-122 (1996)
12. Halevy, A., Rajaraman, A., Ordille, J.: Data integration: The teenage years. In: VLDB. pp. 9-16 (2006)
13. Hamdi, F., Niraula, B.S.N.B., Reynaud, C.: Taxomap in the oaei 2009 alignment contest. In: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009). CEUR Workshop Proceedings, vol. 551, pp. 230-237 (2009)
14. Hung, H.F., Kao, H.P., Chu, Y.Y.: An empirical study on knowledge integration, technology innovation and experimental practice. Expert Systems with Applications 35(1-2), 177-186 (2008)
15. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Asmov: Results for oaei 2009. In: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009). CEUR Workshop Proceedings, vol. 551, pp. 152-159 (2009)
16. Levy, A.Y.: The information manifold approach to data integration. IEEE Intelligent Systems 13, 12-16 (1998)
17. Maedche, A., Motik, B., Silva, N., Volz, R.: Mafra - a mapping framework for distributed ontologies. In: EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web. pp. 235-250. Springer-Verlag, London, UK (2002)
18. Matteo, B., Paolo, B., Paolo, T.: Enabling distributed knowledge management:Managerial and technological implications. Tech. rep., Ingegneria e Scienza dell'Informazione, University of Trento (2002)
19. Murray, K.S.: Learning as Knowledge Integration. Ph.D. thesis, The university of Texas, Austin (1995)
20. Murray, K.S.: Ki: A tool for knowledge integration. In: AAAI/IAAI, Vol. 1. pp.835-842 (1996)
21. Nagy, M., Vargas-Vera, M., Motta, E.: Multi agent ontology mapping framework in the aqua question answering system. In: International Mexican Conference on Artificial Intelligence (MICAI-2005). pp. 70-79 (2005)
22. Nagy, M., Vargas-Vera, M., Motta, E.: Multi-agent ontology mapping with uncertainty on the semantic web. In: Proceedings of the 3rd IEEE International Conference on Intelligent Computer Communication and Processing. pp. 49-56 (2007)
23. Nagy, M., Vargas-Vera, M., Motta, E.: Managing conflicting beliefs with fuzzy trust on the semantic web. In: The 7th Mexican International Conference on Artificial Intelligence (MICAI 2008). pp. 827-837 (2008)
24. Nagy, M., Vargas-Vera, M., Stolarski, P.: Dssim results for oaei 2008. In: Proceedings of the 3rd International Workshop on Ontology Matching. pp. 147-159 (2008)
25. Sanjay, M., Felix, F.: Towards a generic model of configuration tasks. In: IJCAI'89: Proceedings of the 11th international joint conference on Artificial intelligence. pp. 1395-1401. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1989)
26. Seddiqui, M.H., Aono, M.: Anchor-ood: Results for oaei 2009. In: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009). CEUR Workshop Proceedings, vol. 551, pp. 127-134 (2009)

27. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press (1976)
28. Wang, P., Xu, B.: Lily: Ontology alignment results for oaei 2009. In: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009). CEUR Workshop Proceedings, vol. 551, pp. 186-192 (2009)
29. Zhang, X., Zhong, Q., Shi, F., Li, J., Tang, J.: Rimom results for oaei 2009. In: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009). CEUR Workshop Proceedings, vol. 551, pp. 208-215 (2009)
30. Zygmunt, A., Ko_Zlak, J., Siwik, L.: Agent-based environment for knowledge integration. In: ICCS 2009: Proceedings of the 9th International Conference on Computational Science. pp. 885-894. Springer-Verlag, Berlin, Heidelberg (2009)