Workspaces in the Semantic Web

Shawn R. Wolfe, Richard M. Keller

National Aeronautics and Space Administration
Ames Research Center, Moffett Field, CA 94035-1000
{Shawn.R.Wolfe, Richard.M.Keller}@nasa.gov

Abstract

Due to the recency and relatively limited adoption of Semantic Web technologies, practical issues related to technology scaling have received less attention than foundational issues. Nonetheless, these issues must be addressed if the Semantic Web is to realize its full potential. In particular, we concentrate on the lack of scoping methods that reduce the size of semantic information spaces so they are more efficient to work with and more relevant to an agent's needs. We provide some intuition to motivate the need for such reduced information spaces, called *workspaces*, give a formal definition, and suggest possible methods of deriving them.

Introduction

The technologies of the Semantic Web have yet to achieve the widespread adoption of the World Wide Web. To date, researchers have focused more on foundational issues (e.g., representational formats and capabilities) than on pragmatic issues of scale or efficiency. Ultimately, these practical issues will need to be addressed if the Semantic Web is to gain widespread adoption. In this paper, we focus on one such important issue involving mechanisms for filtering and restricting the set of knowledge statements (e.g., RDF triples) available within a semantic information space, depending on the application context. There are numerous pragmatic reasons why one needs to restrict a semantic space, for example to decrease the search space, limit the scope of reasoning, to improve reasoning efficiency, to reduce information overload, and to customize visual presentations for human users.

As an example, consider an agent searching for "in-plan" providers of a specific medical treatment, as described in Berners-Lee et al.'s influential *Scientific American* article on the Semantic Web (Berners-Lee et al. 2001). Let us presume that there is a semantically marked-up data source that serves as a directory of medical providers. In this case, what steps must be taken for the agent to find the appropriate information? First, it is unlikely that the agent and the directory use the same ontology, so some form of ontology alignment will probably be necessary; this problem has received considerable attention (Kalfoglou and Schorlemmer 2003; Noy 2004). Second, the directory is not likely to be structured in a way that is best suited for the agent's search. The directory may include providers outside the local geographic area, or providers in the wrong specialty area, or it may not make any mention of which providers belong to which insurance plans. In essence, the agent is faced with finding a needle in a haystack; the information it seeks is in the repository, along with a great amount of irrelevant information, and there is no easy way to separate the relevant from the irrelevant. The result is information overload.

One approach to identify the relevant information is to access all potentially relevant information in the directory and use reasoning to restrict the scope. The problem with this approach is one of scale; the more information that is accessed, the more time and computing resources required to store and process the data. Alternatively, if the directory supports searching, the agent may try to scope the space by forming a query that more accurately describes the information request. This approach, too, has its drawbacks. The directory may not support sophisticated queries. Differences in the agent and directory ontologies may require that the query scope be broadened. Finally, the precise query may be very complex, making it difficult to derive and verify that the query will return exactly the desired information.

Information Overload in SemanticOrganizer

We have repeatedly encountered the need to restrict the information space in our work on SemanticOrganizer (Keller et al. 2004), a semantic repository that allows users to store knowledge about work-related items (such as documents, datasets, persons, and other domain-specific concepts) and the interrelationships among these items. SemanticOrganizer has over 500 registered users ranging from occasional users to those who use SemanticOrganizer on a regular basis as the primary storage and retrieval system for their work-related knowledge products. Its single ontology covers a wide variety of domains, from project management to scientific inquiry to accident investigation. SemanticOrganizer has over 400 ontology classes defined, with 45,000 instances of those classes and 150,000 semantic links between these instances.

The SemanticOrganizer system supports various methods of searching and browsing of this information, but as the size of the repository grows, it produces more dense information displays and voluminous search results -- even though much of the information displayed to a user may be irrelevant to their current needs and work context. This problem has forced us to consider methods of restricting the user's information space.

Access permissions, defined on instances within SemanticOrganizer, reduce the amount of information available to a given user but do not fully solve the problem. Because access permissions are intended to prevent unauthorized access rather than access to irrelevant information, they are not an appropriate mechanism for restricting the information space based on relevancy: the problem is not what is accessible, but what is relevant to the user. SemanticOrganizer partially addresses this by allowing users to restrict their semantic space to only instances of certain concepts (i.e., filtering out instances of irrelevant classes). Nonetheless, finer-grained techniques are needed to further reduce information overload -- there may be irrelevant instances of a relevant concept, and irrelevant knowledge statements (i.e., RDF triples) that refer to a relevant instance.

It is possible to view the process of restricting an agent's information space in terms of a series of filtering operations. Consider the following example. Imagine that an accident investigator is browsing information in the SemanticOrganizer repository to orientate herself with a near-miss accident involving equipment failure during an experiment performed in a wind tunnel. Some information would be protected through access permissions and would not be available to the investigator, for instance, the salaries of the employees stationed at the wind tunnel. However, additional information could also be filtered out as irrelevant to *any* investigation, for instance, the investigator's salary. Finally, information that is both relevant to investigations in general and accessible to the investigator, but *not relevant* to the investigation at hand could be filtered out; for instance, water samples taken at the wind tunnel during a previous investigation of a *Legionella pneumophila* outbreak. The information that remains after all the filtering operations are complete is considered part of the investigator's current *workspace*.

The information-scoping problem we have encountered in SemanticOrganizer is a specialization of the more general problem of establishing a common context for communication between two agents, with our specific agents being SemanticOrganizer, on one hand, and a human user, on the other. Our human agents are resource bound just as software agents are, with limits on time and processing power. By establishing a shared context appropriate for the current situation, users can increase their efficiency when "conversing" with SemanticOrganizer. In particular, users will spend less time aligning their mental models to that of SemanticOrganizer. In addition, since the amount of information in a workspace is a subset of the overall information space, users will spend less time sifting through irrelevant information.

Related Work

The problem of restricting an information space to a relevant subset has been the focus of information retrieval (IR), where the problem is usually regarded as retrieving a set of documents from a corpus (see (Salton 1983) for an overview). Typically, the user selects some set of keywords that capture the area of interest, and these keywords are used to query the corpus. The bulk of information retrieval techniques do not make explicit use of semantics, and instead use statistical methods to retrieve relevant documents.

Search queries can be viewed as another way of restricting one's view to a relevant subset. Unlike information retrieval techniques, the search terms must explicitly characterize the subset. Query languages are usually quite expressive, but precise query results often require highly complex queries. As a result, query languages alone are not ideal for adequately scoping the relevant subset. Query languages for the Semantic Web are still evolving, with a variety of languages currently available (Haase et al. 2004). In databases, views defined by queries have been used to limit the scope of subsequent operations. Similarly, variants of RQL have been designed to define a view on a Semantic Web (Maganaraki et al. 2004; Volz et al. 2002).

We have previously suggested a method of restricting a user's view of a semantic repository by choosing a subset of an ontology called an *application module* (Keller et al. 2004). Each application module contains only the classes that are relevant to a particular domain. Knowledge statements that refer to instances of classes not in the application module are filtered out. In addition to filtering, application modules provide some presentation characteristics that allowed users to view instances using their own terminology.

Noy and Musen devised a method for specifying a subset of an ontology through traversal (Noy and Musen 2004). Their focus was primarily on facilitating ontology re-use. Rather than exporting an entire ontology, a user could formulate the relevant portion of the ontology by specifying key concepts and then traversing to related concepts using a traversal directive. Since a procedure can be specified to define the desired subset of the ontology, rather than explicitly choosing the ontology, traversal views offer a greater flexibility and dynamism than the application modules of Keller et al.

Examples of Workspaces

What constitutes an effective workspace will change over time, depending on the intent of the agent. To illustrate the circumstantial nature of workspaces, we present illustrative examples describing the types of workspaces required by an investigator named John during various phases of his work as part of an accident investigation team.

Workspaces Based on the Domain

As John joins the investigation team, his first objective is to familiarize himself with the investigation conducted thus far. John is primarily browsing through the information related to the investigation at this point; he does not have specific information to search for nor does he know what kind of information is available. To support this initial browsing activity, it makes sense to restrict the workspace to only those knowledge statements that apply directly to the investigation at hand. Other information, such as similar investigations at other sites or other investigations at the same site might prove useful to John at a later time, but would currently only make his initial orientation more difficult.

Workspaces Based on a Specific Goal

As John becomes more familiar with the investigation, he naturally proceeds to develop hypotheses, for instance, that poor maintenance procedures led to the failure of a particular machine part. To test his hypothesis, John wishes to restrict his view to only those knowledge statements that relate to the machine of interest and/or maintenance. However, John may choose to consider historical information from other investigations relating to these topics, to find other examples of failures, changes in maintenance procedures, or previous uses of the failed part.

Workspaces Based on Time

Over time, the shape of the investigation changes; new evidence has eliminated some hypotheses and led to new areas of inquiry. To keep abreast of the growing areas of the investigation, John restricts his workspace to include only knowledge statements that have been recently added, for instance statements added during the last week. By doing so, John is directed towards new evidence that would need to be evaluated as well as new hypotheses developed by his co-investigators. Older knowledge statements are no less true, but are no longer novel and therefore of less interest.

Workspaces Based on Task

Finally, as the investigation wraps up, John is tasked with developing a report of the investigation's findings and recommendations. John needs to consider information from all phases of the investigation now, not just the most recently added. However, he is less interested in the details of supporting evidence than in the proven hypotheses, and has no interest at all in the disproven hypotheses. Though John is primarily interested in the current investigation, he wants to bring information from other investigations into his workspace, for example if they discussed findings related to the investigation at hand.

These examples support our viewpoint that the notion of an "appropriate" workspace within SemanticOrganizer is a highly situated notion; the subset of knowledge statements that are relevant to the user at any given point in time depends on the user's work context.

Workspace Definition

Having developed our intuition about workspaces, we now present a more formal definition, illustrated in Figure 1. A workspace is defined with respect to two agents, one a source of information (an information-providing agent: IPA), and the other a requestor of information (an information-requesting agent: IRA).

Let KS_{IPA} be the set of knowledge statements held true by the IPA.

Let $P_{IRA} \subseteq KS_{IPA}$ be the subset of statements that the information-providing agent chooses to publish to the information-requesting agent.

Let $R_{IRA} \subseteq KS_{IPA}$ be the subset of statements that fit some notion of relevancy held by IRA.

Let $C_{IRA} \subseteq KS_{IPA}$ be the subset of statements that can be mapped into the vocabulary used by the IRA. (We assume that there is a partial mapping from statements in the IPA's vocabulary to statements in the IRA's vocabulary – an ontology alignment.) C_{IRA} constitutes the subset of the IPA's statement that the IRA can understand.

With respect to a given IPA, a workspace, W, is defined for a given IRA as follows:

$$W = P_{IRA} \cap R_{IRA} \cap C_{IRA}$$

The workspace for the information-requesting agent is thus defined as the subset of the information-provider's knowledge that the agent is allowed to see, that it can understand, and that is relevant.

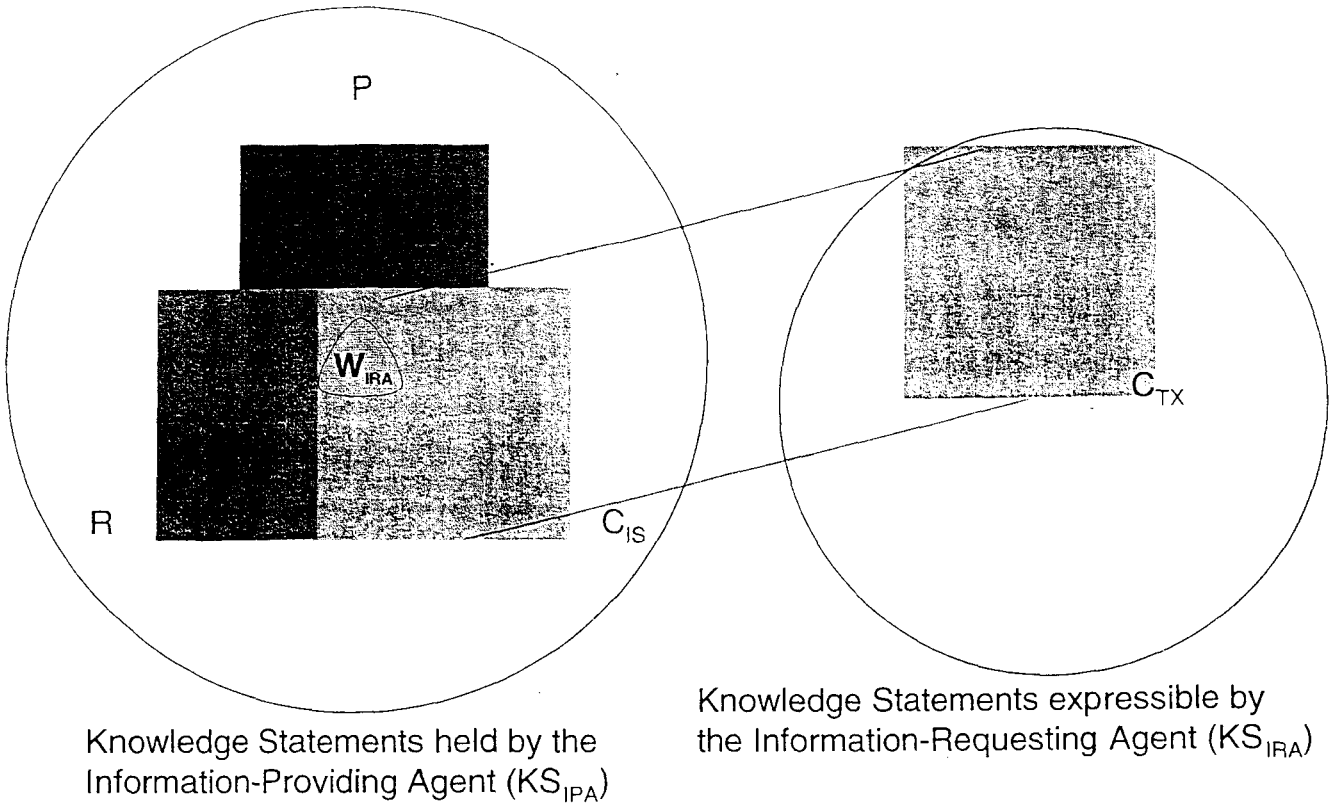

Figure 1: A graphical depiction of a workspace, W, defined for an Information-Requesting Agent (IRA) querying an Information-Providing Agent (IPA). At left is the set of knowledge statements (KS) held true by the IPA; at right is the set of knowledge statements expressible by the IRA. W is defined by the intersection of three subsets of statements held by the IPA: P is the set of statements that the IPA has published to the IRA; R is the set of statements that are relevant to the IRA; and C_{IS} is the set of statements that have a mapping into the vocabulary understood by the IRA

Deriving Workspaces

To derive a workspace, all three of its component subsets must be known. We will presume that the information provider already knows what knowledge statements it is willing to divulge to the information requester, i.e., that it already knows what information it must keep private. Deciding what statements can be translated to the information requester's ontology necessarily involves ontology alignment, another hard problem unto itself that is an area of active research. Within the SemanticOrganizer system, the need to align these ontologies was obviated by *application bundles*, in which ontology specialists customize the master ontology based on the information requester's vocabulary. In what follows, we will concentrate on how to define the third subset – the subset of knowledge statements (R_{IRA}) that fit some notion of relevancy for the information requester. We present three ways to define or derive this *relevant subset*, with each method varying with respect to the amount of semantic interpretation required.

Derivation Via Explicit Selection

The simplest, most obvious method is to manually select the relevant subset of statements, for example by a human knowledge engineer familiar with the agent's context of usage. Manual selection results in the highest quality definition of the relevant subset, but requires the most effort. This method is justified if the manual labor can be amortized over many uses by one or more information requester. For instance, once a workspace is defined for a particular investigation, it could be shared by all the investigators. On the other hand, this method represents no overall reduction of effort if the workspace is used once or infrequently. As with any subset selection method, additions to the overall set of knowledge statements KS_{IPA} would require updating of the relevant subset; since this method is manual, updating can be a significant concern, depending on the frequency of updates.

Derivation Via Description

An alternative to manual selection of relevant knowledge statements is to declaratively describe the relevant subset in terms of a formal language. The description represents an abstraction of the relevant subset and should require less manual effort to construct than the explicit selection. In

contrast to the explicit method above, as knowledge statements are added to KS_{IPA}, the existing description would be used to make the selections, requiring no further effort. This method requires less work than the manual method, but produces a relevant subset that contains a higher number of both irrelevant knowledge statement (false positives) and missing relevant knowledge statements (false negatives).

Derivation Via Ontology-Neutral Learning Methods

Finally, learning methods that use ontology-neutral approaches could be used to drastically reduce the amount of effort required for an agent to define the relevant subset. Such approaches are based on either structural properties of the information space, such as graph connectivity, or meta-concepts and relationships that are relevant across ontologies (for instance, utilizing subsumption or identity relationships, but not domain specific relations). These learning techniques would require limited input if at all – possibly a few training examples. The use of limited amounts of input and lack of domain knowledge will generally result in less accurate results than the previous two more knowledge-intensive methods. Nonetheless, due to the amount of labor involved in manually choosing the relevant subset or describing the relevant subset, such automated methods offer a useful alternative when lower quality subsets are acceptable.

A Simple Experiment

In order to start exploring the space of domain-independent learning approaches, we turned again to the investigation domain of SemanticOrganizer. Four mishap investigations have been supported in SemanticOrganizer (Carvalho et al. 2005): the Columbia shuttle, CONTOUR probe, HELIOS autonomous aircraft, and Canard Rotor Wing (CRW) investigations. Much of the information in these investigations is disjoint, since they occurred at different times, as part of different missions, and involved nearly completely disjoint mission teams. Moreover, most of the common information that could have been included within several investigations was instead (re-)defined separately as part of each new investigation. Therefore, there were very few common *instances* among the investigations. There were a few links crossing between instances included in different investigations, though not many.

Experiment Setup

We considered the case of a single user who has access to information in several investigations, but needs to restrict his view to the subset of information relevant to a single investigation. To define a gold standard for evaluating the formation of the relevant subset, we accessed the accounts of other users who each had involvement in only a single investigation. We used the access permissions of each other user to define the relevant subset of instances for their investigation. In order to simplify the experiment, we focused only on identifying relevant *instances* rather than considering the more numerous relevant *knowledge statements*.

Our goal is to derive these relevant subsets of instances automatically – in this case to derive each subset of instances relevant to a specific investigation. Using the information available to us in SemanticOrganizer, we devised the following experiment. First, we took the union of all the instances and links available to the aforementioned user from all four accounts- this constitutes the items accessible across investigations. Second, to restrict the area to only the domain of investigations, we filtered out all information that was not part of the domain of discourse of investigations (for instance, some information on the ontology itself was represented). Finally, we created a simple algorithm to group the instances into clusters around each investigation.

The Algorithm

Our algorithm takes as input a network of nodes and edges (e.g., an RDF graph), already filtered by permissions and an area of discourse, and *focal instances* that define relevant subsets of instances. Each focal instance is the starting point for a cluster; in our experiment we had four such focal instances, namely each instance of the *Investigation* class. The algorithm produces as output one subset of instances for each focal instance. These subsets may overlap, and the union of these subsets may not include all instances from the original graph. We used the shortest path through the network from an instance to each focal instance as a simple heuristic for deriving the subsets. Each instance was placed within the cluster of the focal instance to which it was closest; if it was equally close to more than one focal instance, it was put in the cluster of each such focal instance. We present the pseudocode of this algorithm below:

For every focal instance F

  Define $S_F = \{\}$

For every instance n in G

  Let C be the set of focal instances closest to n

  For every focal instance F in C

    Add n to S_F

Return: All sets S_F corresponding to each focal instance F

Our intuition was that this algorithm should perform well on this particular task. However, the network was connected, with a path existing from every node to every other node, so it was possible that the algorithm would not perform well at all.

Experimental Results

On this particular experiment, the algorithm outperformed our expectations. We evaluated the quality of the derived subsets of instances in terms of the information retrieval measures of recall, precision and F-measure (Table 1).

	Size of Correct Subset	Size of Derived Subset	Recall	Precision	F-Measure
CRW	349	336	0.82	0.85	0.83
Columbia	4299	4212	0.97	0.998	0.98
CONTOUR	1033	992	0.96	0.998	0.98
Helios	1461	1444	0.99	0.999	0.99

Table 1. Evaluation of derived subsets for each investigation.

Despite these extremely high outcome measures, the conclusions that we can draw from this experiment are very limited. The domain was clearly well-suited to the algorithm's shortest-path heuristic since it had easily-defined subsets that had very little overlap and linkages between subsets. Furthermore, artificial changes to the domain decreased the number of links between subsets: instances that could have been in multiple subsets were often redefined separately in each, and the access permissions on the different areas made linking across subsets difficult. Though we feel that though these circumstances have probably inflated the results somewhat, this algorithm would still perform reasonably without the artificial changes. However, not all domains are likely to have such neatly separated relevant subsets, and the performance of this simple algorithm on such a domain is unknown.

Discussion

While our experiment does not show that the simple shortest-path algorithm presented would be adequate in general, it does show that there is promise in exploring relatively ontology-neutral methods for deriving relevant subsets. Indeed, for the investigations modeled in SemanticOrganizer, we could have used this method to derive the subset of instances relevant to each investigation with excellent results. Though we have not extended the algorithm to consider individual knowledge statements instead of instances, we could do so by including all knowledge statements that refer only to instances in the relevant subset and excluding all that refer to instances outside the subset.

Ultimately, we do not believe that ontology-neutral automated techniques alone will be adequate in most cases. Rather, we suggest that they could be used to generate an initial, rough cut of the relevant subset that could then be refined. For instance, the relevant subset could be further refined by using additional user defined descriptions to add or subtract from the relevant subset. Presumably, such "corrective" abstractions would be simpler to engineer than those that start from scratch. One interesting possibility would be to use the automatically derived subsets to generate the initial abstraction as a starting point, i.e., generating a description that defines a subset that closely matches the automatically derived subset. Finally, if additional refinements were needed, the subsets could be adjusted manually- again with considerably less overall effort than if the entire effort had been manual.

Future work

We have explored the use of a general workspace derivation technique that is independent of a given ontology, but much work remains to develop widely applicable techniques. One possibility for follow-on work would be to continue to evaluate the simple shortest-path algorithm in other domains, and to more fully evaluate its performance in the given experiment. The shortest path algorithm could readily be expanded to a weighted path algorithm that gives different weights for different links, perhaps based on the ontology or other characteristics. Furthermore, the current algorithm should be extended to apply to individual knowledge statements instead of instances and then evaluated.

Other techniques for deriving the relevant subset should also be explored. Heuristics that are not based on properties of the graph but on information retrieval methods, such as TF-IDF, are a possibility. In addition, standard machine learning methods could be explored, such as traditional clustering techniques adapted to a Semantic Web framework or relational data mining methods. We have restricted our experiments to deriving relevant subsets defined by domain, but other kinds of relevant subsets should be considered, for instance subsets defined by a specific task, goal, or timeframe. Finally, incorporating some amount of semantic interpretation into these approaches, as well as having them interact with manually derived abstractions, are directions that we feel will ultimately be the most successful.

Conclusion

As the Semantic Web gains in popularity and acceptance, it will also grow in size. To date, few semantic repositories have grown to a size that their usability suffers, but SemanticOrganizer is one such example. For the vision of the Semantic Web to be realized, these issues of scale must be addressed. We have presented one definition of a restricted view on a semantic network, which we have called a *workspace*. In essence, a workspace is the intersection of three sets; what you have permission to see, what you can understand, and what is relevant in the current situation. Of these three concepts, we felt the latter, what is *relevant*, was the one most in need of our attention

in the context of the developing Semantic Web. We have described some of the techniques that can be used to derive these relevant subsets, and have shown that even a very simple approach with minimal semantic interpretation can be successful in some domains. Ultimately, though, we feel that effective methods will require a combination of both domain independent and domain specific approaches.

Acknowledgements

We would like to thank Ian Sturken, Dan Berrios, and the SemanticOrganizer team for their contributions to this paper. Our work on SemanticOrganizer was funded by the NASA Intelligent Systems Project within the Computing, Information and Communications Technology Program and by the Investigative Methods and Tools Project within the Engineering for Complex Systems Program.

References

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*.

Carvalho, R. E., Williams, J., Sturken, I., Keller, R. M., and Panontin, T. InvestigationOrganizer: The Development and Testing of a Web-Based Tool to Support Mishap Investigations. In *Proceedings of the IEEE Aerospace Conference 2005*, Big Sky, MT, USA.

Haase, P., Broekstra, J., Eberhart, A., and Volz, R. A Comparison of RDF Query Languages. In *Proceedings of the Third International Semantic Web Conference (ISWC-2004)*, Hiroshima, Japan.

Kalfoglou, Y., and Schorlemmer, M. (2003). Ontology Mapping: The State of the Art. *The Knowledge Engineering Review*, 18(1), 1-31.

Keller, R. M., Berrios, D. C., Carvalho, R. E., Hall, D. R., Rich, S. J., I. B. Sturken, Swanson, K. J., and Wolfe, S. R. Semanticorganizer: A Customizable Semantic Repository for Distributed NASA Project Teams. In *Proceedings of the Third International Semantic Web Conference (ISWC-2004)*, Hiroshima, Japan.

Maganaraki, A., Tannen, V., Christophides, V., and Plexousakis, D. (2004). Viewing the Semantic Web through RVL Lenses. *Journal of Web Semantics*.

Noy, N. F. (2004). Semantic Integration: a Survey of Ontology-Based Approaches. *ACM SIGMOD Record*, 33(4).

Noy, N. F., and Musen, M. A. Specifying Ontology Views by Traversal. In *Proceedings of the Third International Semantic Web Conference (ISWC-2004)*, Hiroshima, Japan.

Salton, G. (1983). *Introduction to Modern Information Retrieval*, McGraw-Hill.

Volz, R., Oberle, D., and Studer, R. On Views in the Semantic Web. In *Proceedings of the Proceedings of the 2nd International Workshop on Databases, Documents, and Information Fusion (DBFUSION 02)*, Karlsruhe, Germany.