Semantic Depth and Markup Complexity*

Guofei Jiang and **George Cybenko**
Institute for Security Technology Studies and
Thayer School of Engineering
Dartmouth College, Hanover, NH 03755
{gfj, gvc}@ dartmouth.edu

James A. Hendler
Department of Computer Science
University of Maryland
College Park, MD 20742
hendler@cs.umd.edu

Abstract - *In order to achieve interoperability among heterogeneous systems, markup languages such as XML and DAML are being used to describe distributed systems and data. The ability to successfully interoperate based on semantic markup depends on the ability to create, use and manage shared ontologies of concepts and their interrelationships. Specifically, communicating systems in a networked environment have to achieve a certain level of semantic agreement for them to understand and process exchanged data. A challenging question is how deep the semantic agreement has to be in order to satisfy the communication needs in an environment. Additionally, what is the markup complexity resulting from pursuing that depth of semantic agreement? This paper introduces the concept of semantic depth and markup complexity and proposes models to measure the markup complexity. Furthermore, it is shown that markup complexity can be reduced by employing hierarchical ontologies after partitioning the domain into smaller sub-domains.*

Keywords: Semantic depth, markup complexity, semantic interoperability, hierarchical ontology, domain partition

1 Introduction

Distributed computing is migrating from tightly coupled architectures to loosely coupled distributed environments. Many new technologies such as Grid computing [6] and Web Service [1] are being developed to expedite this migration. In a loosely coupled environment, computing and data resources are located throughout networks and may not be centrally created or administered. In order to achieve interoperability among heterogeneous systems, markup languages such as Extensible Markup Language (XML) and DARPA Agent Markup Language (DAML) [2] are being used to describe distributed systems and data.

The ability to successfully interoperate based on semantic markup depends on the ability to create, use and manage shared ontologies of concepts and their interrelationships. Specifically, communicating systems in a networked environment have to achieve a certain level of semantic agreement for them to understand and process exchanged data. Without common vocabularies, a machine itself has no way to understand the terminology in the structured data. An interesting question is how deep the semantic agreement has to be in order to satisfy the interoperability needs in an environment. However, when deeper semantics is required, the markup complexity of that environment usually increases. Therefore, what is the markup complexity resulting from pursuing that depth of semantic agreement?

In this paper, we introduce the concept of semantic depth and markup complexity and propose models to measure the markup complexity. Furthermore, we analyze how markup complexity can be reduced by employing hierarchical ontologies after partitioning the domain into smaller sub-domains that may be easier to semantically describe in a consensus manner.

The remainder of the paper is organized as follows: In Section 2, we introduce the concept of semantic depth and markup complexity. Section 3 and 4 propose the model and parameters needed to measure markup complexity. In Section 5, we compute and analyze how markup complexity can be reduced after partitioning domain ontology. In Section 6, we analyze and discuss our results.

2 Semantic depth

We note that semantics for system interoperability typically includes two kinds of semantics: *implicit* semantics and *explicit* semantics. Implicit semantics represent the pre-assumed semantics between two systems, such as hard-coded routines and APIs available to them. Implicit semantics are not embedded in the exchanged data. Instead, communicating systems interpret the implicit semantics of data based on their pre-assumed agreements and understanding. Explicit semantics refer to explicit common representations and descriptions required for effective interoperability between systems. Communicating systems could interpret the explicit semantics of data based on their embedded data description. When two systems could possibly interpret implicit semantics differently, more explicit semantics

* **0-7803-7952-7/03/$17.00 © 2003 IEEE.**

must be used to resolve the differences existing in their interpretations. Here we introduce the concept of *Semantic Depth,* which captures how deep explicit semantics need to be in order to achieve a desired degree of interoperability.

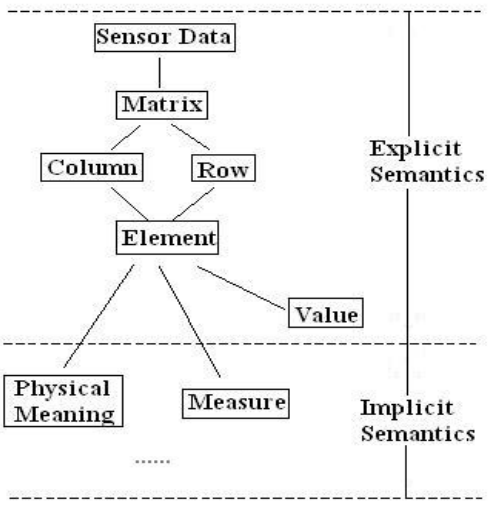

Figure 1: An example of implicit and explicit semantics

An example of implicit semantics and explicit semantics is shown in Figure 1. Assume that a sensor's data can be represented with a matrix. The matrix is represented with multiple columns or rows. Each element in the matrix has its own physical meaning, value and measure, for example, temperature, 40, centigrade. One system describes the above sensor data and sends it to another system with the following metadata:

```
<SensorData>
   <Matrix>
      <Column>
         <Element Value=40>
         <Element Value=13>
         <Element Value=21>
      </Column>
      <Column>
         <Element Value=15>
         <Element Value=27>
         .....
   </Matrix>
</SensorData>
```

In this message, the matrix's structure and each element's value are explicitly represented. This is an example of explicit semantics. But each element's physical meaning and measure properties are not described in the metadata. Instead, both systems are implicitly assumed to understand the physical meaning and measurement units based on each element's relative position in that matrix. For example, the element at the first column third row always represents the temperature of the observed object and the temperature is measured in Celsius. Therefore, the receiving system has to interpret this message based on these implicit assumptions.

Communicating systems may have misunderstanding of implicit semantics since it is not explicitly represented in the exchanged data. In the above example, the receiving system may incorrectly assume that the temperature is measured with Fahrenheit. In this case, extra explicit semantics are needed to represent the measurement in the exchanged data.

In this paper, we use *Markup Complexity* to represent how many properties need to be explicitly described in an environment in order to achieve a desired level of semantic depth. When semantic depth grows, markup complexity usually increases because more properties need to be described explicitly. When developed and used in a small community, communicating systems tend to use implicit semantics to leverage communication efficiency. Scientific jargon is often created to replace the lengthy explicit explanation of certain concepts. The use of implicit semantics could reduce markup complexity in data processing and bandwidth demands for communication only if all communicating systems can interpret the implicit assumptions correctly.

In a large group by contrast, communicating systems could interpret some implicit semantics differently because of their different understanding on the concepts.

3 Models

In a specific domain, n systems need to create a common ontology to markup their data. Assume that the $i^{th}(i=1,2,\cdots n)$ system requires $k_i(i=1,2,\cdots,n)$ attributes in the ontology to describe the properties of interest. Here we denote these attributes as the system's "K-attributes". Additionally, we assume that this domain can be naturally partitioned into m sub-domains by categories. Every sub-domain clusters $n_j(j=1,2,\cdots,m)$ systems respectively and every system is associated with only one sub-domain. After domain partition, systems within the same sub-domain will require deep semantics for their intensive intra-domain communication. Meantime, systems across sub-domains only need shallow semantics for their limited interactions. That is, only a small percentage of property descriptions are needed for their interactions. The hierarchical ontology after such domain partitioning is shown in Figure 2.

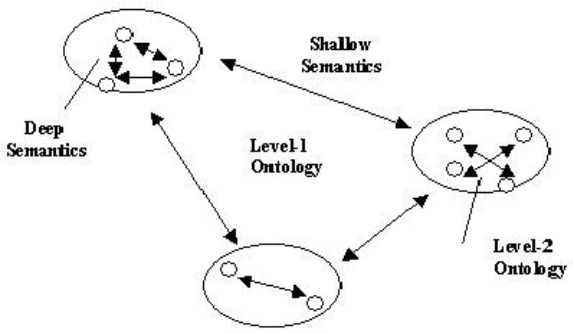

Figure 2: Semantic depth and domain partition

In our model, we make the following rules to guarantee adequate semantic depth required by the domain. Committing to these rules, all systems in this domain can achieve expected interoperability with adequate explicit semantics in their data.

Rule 1: Every system's K-attributes have to be fully described in the common domain ontology.

Rule 2: Every system has to markup its data with all attributes in its domain ontology.

With Rule 1, the common domain ontology includes all property descriptions of each system's interest. Furthermore, Rule 2 guarantees adequate explicit semantics in each system's data and adequate semantic depth in the domain. Following these rules, we propose the following markup solutions and then compare their markup complexity.

Solution 1: A common ontology is created to include all n systems' K-attributes. All these n systems are marked up with this common ontology.

Solution 2: For each sub-domain $j(\ j = 1,2,\cdots,m\ )$, a sub-domain ontology is created to include K-attributes of $n_j(\ j = 1,2,\cdots,m\ )$ systems in that sub-domain. Meanwhile, a higher level inter-domain ontology is created to include some attributes used for inter-domain interactions. Every system is marked up with its specific sub-domain ontology for internal communication and the inter-domain ontology for inter-domain communication.

In fact, each sub-domain can be partitioned into smaller sub-sub-domains and then we have multiple level ontology systems. In this paper, we use two-level domain partition for our complexity analysis. Later we will discuss how our results can be easily extended to multiple level domain partitions with hierarchical ontologies.

4 Parameters

In this section, we list the parameters needed in our modeling and analysis. If A is a set and a is the element of this set, we use $|A|$ to denote the number of elements in the set A and use the union, $\bigcup_{a\in A} a$, to denote the set that includes no duplicate elements.

n : the number of systems in a specific domain d ;

k_i : the number of K-attributes requested by system i, $1 \le i \le n$;

a_{il} : the l^{th} attribute requested by system i, $1 \le i \le n$, $0 \le l \le k_i$.

m : the number of sub-domains, $m \le n$;

d_j : the sub-domain j , $1 \le j \le m$;

d : the whole domain, $d = \bigcup_{1\le j\le m} d_j$;

n_j : the number of systems clustered in the sub-domain d_j , $1 \le j \le m$, $n = \sum_{j=1}^{m} n_j$;

w : the proportion of the largest sub-domain to the whole domain d , $w = \frac{\max\limits_{j=1,2,\cdots m} n_j}{n}$, $\frac{1}{m} \le w \le 1$;

r : the number of total K-attributes (of all n systems) in the domain d , $r = \sum_{i=1}^{n} k_i$;

u : the set of unique attributes (of all n systems) in the domain d , $u = \bigcup_{\substack{1\le i\le n \\ 0\le l\le k_i}} a_{il}$.

p : the overlap rate of total K-attributes (of n systems) in the domain d , $0 \le p \le 1$, $p = 1 - \frac{|u|}{r}$;

r_j : the number of total K-attributes (of all n_j systems) in the sub-domain d_j , $r_j = \sum_{s\in d_j} k_s$, $1 \le j \le m$;

u_j : the number of unique attributes (of all n_j systems) in the sub-domain d_j , $u_j = \bigcup_{\substack{s\in d_j \\ 0\le l\le k_s}} a_{sl}$, $1 \le j \le m$;

p_j : the overlap rate of total K-attributes (of n_j systems) in the sub-domain d_j , $p_j = 1 - \frac{|u_j|}{r_j}$, $1 \le j \le m$;

p_{inter} : the overlap rate of inter-domains' attributes, $p_{inter} = 1 - \frac{\left|\bigcup_{1\le j\le m} u_j\right|}{\sum_{j=1}^{m} |u_j|}$;

t : the percentage of total unique sub-domain's attributes used for inter-domain communication, $0 \le t \le 1$;

C_{d_j} : the markup complexity for intra-domain communication in the sub-domain d_j , $1 \le j \le m$;

C_{inter} : the markup complexity for inter-domain communication;

C_1 : the overall markup complexity of solution 1;

C_2 : the overall markup complexity of solution 2.

5 Markup complexity

In this section, at first we analyze and compute the mark complexity for the solution 1 and 2. Then we compare the markup complexity of these two solutions.

5.1 Complexity computation

In the solution 1, every system requires k_i K-attributes in the common ontology and so we have total $\sum_{i=1}^{n} k_i$ K-attributes determined by the n systems in the whole domain d. Define $r = \sum_{i=1}^{n} k_i$. Since some systems may require the same attributes, we only need $\left| \bigcup_{\substack{1 \le i \le n \\ 0 \le l \le k_i}} a_{il} \right|$ attributes in the common ontology to satisfy every system and satisfy Rule 1. Every system needs to be marked up with these attributes according to Rule 2 and the total markup complexity C_1 of these n systems is

$$C_1 = n \cdot \left| \bigcup_{\substack{1 \le i \le n \\ 0 \le l \le k_i}} a_{il} \right| = n \cdot |u|, \tag{1}$$

where $u = \bigcup_{\substack{1 \le i \le n \\ 0 \le l \le k_i}} a_{il}$. The overlap rate of all attributes is $p = 1 - \frac{|u|}{r}$.

In solution 2, the whole domain d has m sub-domains and each sub-domain d_j includes n_j systems. Every system has $k_s (s \in d_j)$ K-attributes and then we have a total of $r_j = \sum_{s \in d_j} k_s$ K-attributes required by n_j systems in the sub-domain d_j. After we remove the duplicate attributes required by these systems, we need $\left| \bigcup_{\substack{s \in d_j \\ 0 \le l \le k_s}} a_{sl} \right|$ attributes in the common sub-domain ontology to satisfy these n_j systems. According to Rule 2, the overall markup complexity for all n_j systems in the sub-domain d_j is

$$C_{d_j} = n_j \cdot \left| \bigcup_{\substack{s \in d_j \\ 0 \le l \le k_s}} a_{sl} \right| = n_j \cdot |u_j|, 1 \le j \le m, \tag{2}$$

where $u_j = \bigcup_{\substack{s \in d_j \\ 0 \le l \le k_s}} a_{sl}$. The overlap rate of sub-domain's attributes is $p_j = 1 - \frac{|u_j|}{r_j}$, $1 \le j \le m$. For all m sub-domains, the total markup complexity is

$$\sum_{j=1}^{m} C_{d_j} = \sum_{j=1}^{m} n_j \left| \bigcup_{\substack{s \in d_j \\ 0 \le l \le k_s}} a_{sl} \right| = \sum_{j=1}^{m} n_j \cdot |u_j|. \tag{3}$$

Each sub-domain includes $|u_j|$ non-duplicate attributes and the total attributes in m sub-domains are $\sum_{j=1}^{m} |u_j|$. As we mentioned above, systems across sub-domains may only need shallow semantics for their limited interactions. Here we assume that t percentage of the total unique sub-domain attributes are used for inter-domain communications. Since some sub-domains may request the same attributes for inter-domain communication, we only need $\left| \bigcup_{1 \le j \le m} u_j \right| \cdot t$ attributes to satisfy every sub-domain for inter-domain interactions. According to Rule 2, all n systems have to be marked up with these inter-domain attributes and the overall markup complexity for inter-domain interaction is

$$C_{inter} = n \cdot \left| \bigcup_{1 \le j \le m} u_j \right| \cdot t. \tag{4}$$

Meanwhile, we have

$$u = \bigcup_{\substack{1 \le i \le n \\ 0 \le l \le k_i}} a_{il} = \bigcup_{1 \le j \le m} \left(\bigcup_{\substack{s \in d_j \\ 0 \le l \le k_s}} a_{sl} \right) = \bigcup_{1 \le j \le m} u_j \tag{5}$$

Therefore, the Equation (4) can be rewritten as

$$C_{inter} = n \cdot \left| \bigcup_{1 \le j \le m} u_j \right| \cdot t = n \cdot |u| \cdot t. \tag{6}$$

Since some sub-domain ontologies may have included some of attributes required by the inter-domain ontology, the overall markup complexity of solution 2 is

$$\begin{aligned} C_2 &\le C_{inter} + \sum_{j=1}^{m} C_{d_j} \\ &= n \cdot |u| \cdot t + \sum_{j=1}^{m} n_j \cdot |u_j|. \end{aligned} \tag{7}$$

5.2 Complexity comparison

With Equations (1) and Inequality (7), we can compare the markup complexity of two solutions with the following inequality,

$$\frac{C_2}{C_1} \le \frac{n \cdot |u| \cdot t + \sum_{j=1}^{m} n_j \cdot |u_j|}{n \cdot |u|}. \tag{8}$$

Define $w = \frac{\max_{j=1,2,\cdots m} n_j}{n}$, then we have

$$n_j \le w \cdot n, \quad j = 1,2,\cdots,m; \tag{9}$$

Denote the overlap rate of inter-domain attribute as p_{inter}. With the definition of overlap rates p_j and p, we have

$$p_{inter} = 1 - \frac{\left|\bigcup_{1\le j\le m} u_j\right|}{\sum_{j=1}^{m}|u_j|} = 1 - \frac{(1-p)r}{\sum_{j=1}^{m}(1-p_j)r_j}. \tag{10}$$

Therefore Inequality (8) can be rewritten as

$$\begin{aligned}\frac{C_2}{C_1} &\le \frac{n\cdot|u|\cdot t + \sum_{j=1}^{m} n_j \cdot |u_j|}{n\cdot|u|} \\ &\le \frac{n\cdot|u|\cdot t + \sum_{j=1}^{m} n\cdot w\cdot|u_j|}{n\cdot|u|} \\ &= t + w\cdot\frac{\sum_{j=1}^{m}|u_j|}{|u|} \\ &= t + \frac{w}{1-p_{inter}}\end{aligned} \tag{11}$$

5.3 Result analysis

From Inequality (11), it is straightforward to see that smaller w, t and p_{inter} can lead to less markup complexity in the solution 2.

- The parameter w is defined as $w = \frac{\max_{j=1,2,\cdots m} n_j}{n}$. If the size of sub-domain clusters is smaller, w could be smaller. $\frac{1}{m} \le w \le 1$.

- The parameter t is the percentage of sub-domain attributes used for inter-domain communication. While inter-domain communication requires shallower semantic depth, t is smaller. Note that $0 \le t \le 1$.

- The parameter p_{inter} is defined as $p_{inter} = 1 - \frac{\left|\bigcup_{1\le j\le m} u_j\right|}{\sum_{j=1}^{m}|u_j|}$.

The value of p_{inter} is determined by how much overlap the sub-domains' attributes have. If m set of sub-domain attributes $u_j, 1 \le j \le m$ have no overlap at all, we have $\left|\bigcup_{1\le j\le m} u_j\right| = \sum_{j=1}^{m} u_j$ and $p_{inter} = 1 - \frac{\left|\bigcup_{1\le j\le m} u_j\right|}{\sum_{j=1}^{m}|u_j|} = 0$. Conversely, if m sets of sub-domain attributes $u_j, 1 \le j \le m$ are total overlapped, i.e., every set is the same, we have $\left|\bigcup_{1\le j\le m} u_j\right| = \frac{1}{m}\cdot\sum_{j=1}^{m} u_j$ and $p_{inter} = 1 - \frac{1}{m}$. Therefore, $0 \le p_{inter} \le 1 - \frac{1}{m}$. From Equation (10), we can conclude the relationship between the overlap rates p_{inter}, p_j and p. Define $p_{min} = \min_{j=1,2,\ldots,m} p_j$ and $p_{max} = \max_{j=1,2,\ldots,m} p_j$. With $r = \sum_{i=1}^{n} k_i = \sum_{j=1}^{m}\sum_{s\in d_j} k_s = \sum_{j=1}^{m} r_j$, we have

$$\begin{aligned}p_{inter} &= 1 - \frac{(1-p)r}{\sum_{j=1}^{m}(1-p_j)r_j} \le 1 - \frac{(1-p)r}{(1-p_{min})\sum_{j=1}^{m} r_j} \\ &= 1 - \frac{1-p}{1-p_{min}}.\end{aligned}$$

Therefore, we have

$$\frac{1-p}{1-p_{min}} \le 1 - p_{inter} \le \frac{1-p}{1-p_{max}}. \tag{12}$$

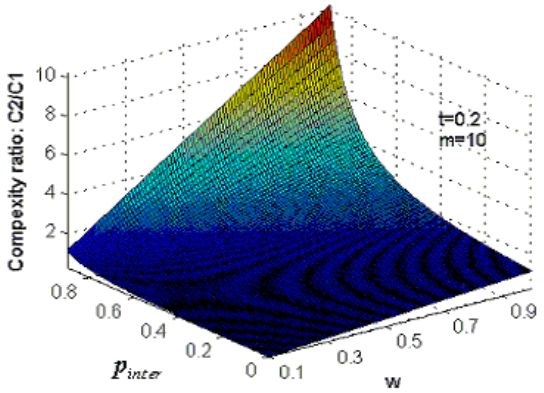

Figure 3: Complexity ratio C2/C1 vs. parameters w and p_{inter} ($t = 0.2$, $m = 10$)

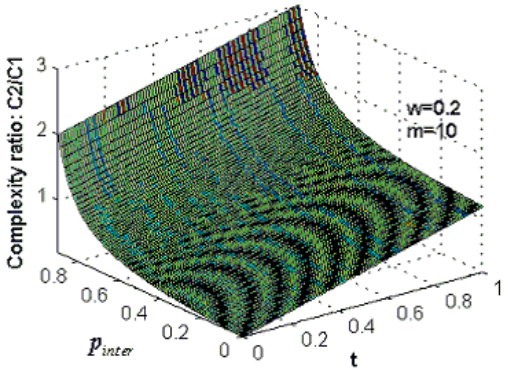

Figure 4: Complexity ratio C2/C1 vs. parameters t and p_{inter} ($w = 0.2$, $m = 10$)

We believe that there are complex relationships between the parameters w, t and p_{inter}. If the domain is evenly partitioned into smaller sub-domains, w could be smaller. However, smaller sub-domains could demand deeper inter-domain semantics and then t, p_{inter} could become larger. Assuming $t = 0.2$ and $m = 10$, it is shown in Figure 3 how mark complexity can be reduced with smaller w and p_{inter}. Meantime, assuming $w = 0.2$ and $m = 10$, Figure 4 shows how mark complexity can be reduced with smaller t and p_{inter}. The parameters w and p_{inter} have constraints $\frac{1}{m} \le w \le 1$ and $0 \le p_{inter} \le 1 - \frac{1}{m}$ respectively. In fact, p_{inter} could be subject to tighter constraints such as Inequality (12), but for convenience we don't reflect that in Figure 3 and Figure 4.

For example, if $t = 0.2$, $w = 0.2$ and $p_{inter} = 0.2$, we have $C_2/C_1 \le 0.45$ according to Inequality (11). Therefore, in this case, solution 2 only needs 45% of the mark complexity of solution 1. However, as seen in Figure 3 and 4, if we partition a tightly coupled domain into many sub-domains, the mark complexity of solution 2 could be much higher than that of solution 1. Obviously not every domain can be partitioned to reduce the markup complexity. For a specific domain, we can obtain the parameters from real data and use Inequality (11) to determine whether the domain should be partitioned to reduce the markup complexity. In fact, this was the motivation for us to develop this result.

6 Discussions

Our model can be easily extended into multiple level partitions of domain ontologies. For n level ontologies, the markup complexity ratio C_n/C_1 can be computed with $\frac{C_n}{C_1} = \frac{C_2}{C_1} \cdot \frac{C_3}{C_2} \cdot \cdots\cdots \frac{C_n}{C_{n-1}}$. Each item on the right side can be computed with Inequality (11). With hierarchical ontologies, we can reduce markup complexity while still keeping adequate semantic depth in communication. With a smaller sub-domain, it is also easier for members to agree on a common ontology. For example, usually it is easier for ten people to agree on a set of concepts than one hundred people. Moreover, we cluster members based on their similarities (high overlap rate in properties of interest), which can also make easier for members to concur on a common ontology. However, it is not reflected in the concept of markup complexity how much effort is needed for a group to concur on a common ontology. If sub-domains do not use a common ontology to mark up their data, ontology mapping has to be used to achieve semantic interoperability in an environment [4].

We note that many clustering and partition algorithms are available in data mining and management [5]. In this paper, we propose a model and parameters to analyze semantic depth and markup complexity in a quantitative way. While domain partitioning and data clustering are well researched in other areas [3], we are interested on how to reduce markup complexity while still achieving adequate semantic depth in communication. To the best of our knowledge, we have not seen any similar work in this area.

7 Conclusions

Communicating systems in a networked environment have to achieve a certain level of semantic agreement for them to understand and process exchanged data. A challenging question is how deep the semantic agreement has to be in order to achieve a desired degree of interoperability. Meanwhile, what is the markup complexity resulting from pursuing that depth of semantic agreement? In this paper, we analyze the implicit semantics and explicit semantics in communication and introduce the concept of semantic depth and markup complexity. Furthermore we propose models and parameters to analyze semantic depth and markup complexity in a quantitative way. With examples, we have illustrated how we can reduce mark complexity with hierarchical domain ontologies while still achieving adequate semantic depth in communication.

Acknowledgements

This research was partially supported by: Defense Advanced Research Projects Agency projects F30602-00-2-0585 and F30602-98-2-0107; the Office of Justice Programs, National Institute of Justice, Department of Justice award number 2000-DT-CX-K001 (S-1).

References

[1] E. Cerami, *Web Services Essentials*, O'Reilly , 2002.

[2] DARPA Agent Markup Language:http://www.daml.org

[3] A. K. Jain, M. N. Murty and P. J. Flynn, "Data clustering: a review", *ACM Computing Surveys,* Vol. 31 No. 3, 1999.

[4] G. Jiang, G. Cybenko and J. Hendler, "Semantic interoperability and information fluidity", in preparation.

[5] G. Karypis, E.-H. Han and V. Kumar, "Chameleon: A hierarchical clustering algorithm using dynamic modeling", *IEEE Computer: Special Issue on Data Analysis and Mining,* Vol. 32, No. 8, 1999.

[6] The Globus Project: http://www.globus.org.