Weaving the Web of Belief into the Semantic Web

Li Ding
University of Maryland, Baltimore County
CSEE department, 1000 Hilltop Circle
Baltimore, Maryland 21250, USA
dingli1@csee.umbc.edu

Timothy Finin
University of Maryland, Baltimore County
CSEE department, 1000 Hilltop Circle
Baltimore, Maryland 21250, USA
finin@csee.umbc.edu

ABSTRACT

Collaboration, especially knowledge sharing, enables the advance of science as well as human society. In cyberspace, socializing the traditionally isolated intelligent software agents is an ultimate goal of the emerging Semantic Web activity. When making collaboration decisions, an agent usually needs explicitly represented facts about the agent world, such as "who knows what" and "who can do what". However, the limited computation and storage resources forbid an agent to independently maintain rational beliefs on all facts about the agent world. So the full picture of the agent world has to be distributed in the knowledge sharing social network of those resident agents. In this paper, we propose a generic representation framework for this distributed knowledge network, which is also called the reminiscent of Quine's *web of belief*. The framework includes: the RDF based *semantic relation model*, which is a cognitive data model for the agent world; a general OWL ontology, which facilitates representing agent properties (such as knowledge and capability) and finer inter-agent trust relations; and the practical issues on maintaining the web of belief for distributed inference.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms

Languages, Theory

Keywords

ontology, trust, Semantic Web, web of belief, knowledge sharing

1. INTRODUCTION

Knowledge sharing is a distinctive human intelligent behavior that enables human beings to acquire knowledge from their ancestors as well as their contemporaries. By cognitively viewing knowledge as individuals' rational beliefs about the world, individuals share knowledge and form a distributed knowledge network, which is called the *web of belief*, where *rational belief* links individuals with world facts and *trust* interlinks individuals as external information sources. One important feature of the *web of belief* is that it enables an individual to selectively acquire knowledge from other information sources. A careful reader, for example, might believe facts from New York Times over those appearing in a local newspaper. Different individuals might customize their views about the same world, and thus make the *web of belief* a complex knowledge network, which contains richer information than the simple sum of all its members' knowledge. For example, a judge may make final judgment by investigating and evaluating the inconsistent witness reports in a court trial. Another important feature of the *web of belief* is that it serves as a distributed knowledge base for planning. Travelers may choose to use a particular travel agency according the recommendations from trusted information sources such as friends and newsgroup recommendations.

The concept of a software "agent" [20, 33] is widely used in many disciplines [34], and agent collaboration is needed in many complex tasks such as business process management [19] and electronic commerce [16]. Moreover, recognizing agents as providers of web services, the Semantic Web Services activity [1] stresses that "The power of Web services ... is that they can then be combined in a loosely coupled way in order to achieve complex operations". However, agent collaboration does not scale well because it is hard to build the *web of belief*.

The obstacles, like many bad things, come in threes: (1) syntactical difference – agents use different languages and dialects to encode their knowledge; (2) semantic difference – agents use different conceptual systems to model their common environment; and (3) distrust – agents must use caution in blindly accepting information and knowledge from other agents. Although the Semantic Web resolves the syntactical difference with RDF/XML, its OWL ontology language can't fully resolve the semantic difference. Therefore, this paper aims at the last two difficulties in the Semantic Web.

In the rest of this paper, we offer a sketch of how to build the *web of belief* on the Semantic Web. Section 2 proposes the semantic relation model to cognitively capture the semantics of the agent world. Section 3 develops the core of an OWL ontology for the web of belief that focuses on representing the agent world, especially agent knowledge and capability, and capturing knowledge provenance. Section 4 adds the "trust" ontology allowing agents to finely represent their trust relations with other agents and to reason about trust relations with the associated axioms. Section 5 discusses practical issues on maintaining the *web of belief* for distributed inference in the dynamic and open agent world, including: agent memory management, trust relation maintenance, and inconsistent knowledge handling. Section 6 discusses related work and Section 7 offers some conclusions and outlines work yet to be done.

2. THE SEMANTIC RELATION MODEL

Both simulation and logical representation can be used to model agents [30], and we propose a RDF based data model, the Semantic Relation Model (SRM), as the cognitive basis for describing the

WWW2004, May 17–22, 2004, New York, NY USA.
ACM xxx.xxx.

[1] http://www.w3.org/2002/ws/Activity

agent world in the Semantic Web. RDF is chosen because of its *expressiveness*, i.e., both data and metadata are represented under the same framework and higher-order statements are supported, its *data independence*, i.e., the logical level RDF graphic representation is independent of its physical representation, and its *conditional consistency*, i.e., individual agent can customize a consistent view (a sub-graph) of the inherently inconsistent big RDF graph, which is the union of all available RDF file on the Semantic web.

The SRM focuses on three basic notions related to RDF graph: Resource, Relation, and Axiom. A *resource* corresponds to a resource in RDF, i.e., a node in an RDF graph. The SRM further splits Resource into three categories: meta-resource (resources defined in rdf and rdf schema, such as rdfs:Class), class-resource (any resource whose rdf:type is rdfs:Class), and individual-resource (otherwise). A *relation* corresponds to a statement in RDF, i.e., a subgraph (two nodes and the arc that links them) in an RDF graph. The SRM further divides Relation into two categories: factual-relation and hypothetical-relation (a hypothetical relation can summarize a collection of factual relations). An *axiom* is an inference rule that enables RDF graph reduction or expansion: it can remove or add arcs in an RDF graph when it matches a certain subgraph (pattern).

Our preliminary work on representing the agent world with the SRM focuses on agent collaboration and knowledge sharing context. Based on a standard view of agents (e.g., [37]), we identify three important class-resources: Agent, Action, and World fact. *Agent* is the collection of computational facilities, each of which knows some world facts and can take some actions. *Action* is the collection of abstract actions. *World fact* is the collection of facts about the agent world. The individuals of Agent and Action are usually referenced as an URI, however, the World fact is the union of rdf:Statement and all of built-in datatypes. In addition, the SRM restrict that an individual-resource should be an instance of any of the three class-resources.

We are particularly interested in the *generic* relations between Agent, Action and World fact as shown in figure 1.

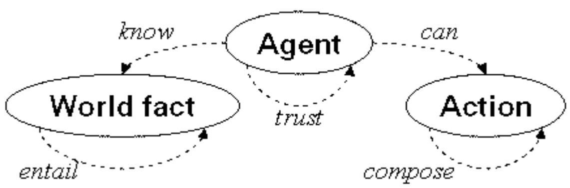

Figure 1: The generic relations in the agent world

- *trust* is an agent-agent social hypothetical relation. It encodes an agent's context specific relation to other agents, such as "whether an agent knows a lot about the sport domain" and "whether a travel agent can skillfully help me book a hotel". Besides trust, there are many other social relations in social network analysis literature [13].

- *know* is an agent-fact factual relation. It encodes an agent's relation to particular world facts, i.e. whether a world fact is in the agent's knowledge base. This relation captures the meaning of *agent knowledge*, i.e., a world fact can be credited as knowledge unless it is known by an agent. It is one type of *agent mental states*, and similar ones are the belief state, the desire state, the obligation state, and etc..

- *can* is an agent-action hypothetical relation. It encodes an agent's relation to particular actions. This relation captures the meaning of *agent capability*, i.e. if an agent is capable of executing a certain type of action in the future. Another useful agent-action factual relation is "did", which associates an agent with an action instance in the past.

- *entail* is a fact-fact factual relation. By viewing world fact as logical sentence, it shows the logical consequence relation between two world facts. Though it enables explicit inference, it also raises the challenge of resolving inconsistency.

- *compose* is an action-action factual relation. It shows how simple actions could be used to compose complex actions.

It is notable that Relation is a subclass of rdf:Statement, which is a subclass of World fact. Therefore, World fact is recursively defined, e.g. an agent knowing a world fact is also a world fact. Such recursive definition makes the RDF graph a hyper graph, but it could be converted to flat graph through reification. In the following context, we will use a four tuple to denote an instance of rdf:Statement

$$(\_u, s, p, o)$$

where $\_u$ is the statement's ID which is unique within the statement's residential RDF file, and the remaining part is the statement's content: s for subject, p for predicate, and o for object.

One important issue is the granularity of a world fact (WF). The recursive definition of world fact corresponds to different sub-graph structure in RDF graph: a simplest WF is a datatype value, which corresponds to a node; a simple WF is a relation, which corresponds to two non-blank nodes and the directed arc that links them; a complex WF is an relation with some embedded world facts, i.e., at least one of the relation's two nodes is blank. A complex WF should be treated as an indivisible sub-graph, and its extent can be determined by expanding the subgraph so that any possible arc from its blank nodes links to one of its nodes.

3. THE WEB OF BELIEF ONTOLOGY

Based on the SRM, we propose an OWL ontology for the *web of belief*. Our ontology consists of two parts: the *core part* that models the agent world especially individual agents' knowledge and capability, and the *trust enhanced part* that models trust to support inter-agent collaboration and knowledge sharing. We will discuss the core part in the rest of this section and leave the trust enhanced ontology to section 4.

Since we focus on the relations in the SRM, we only interest in the referential identity of class-resources, i.e., we assume all class-resources are URIs. Additionally, we only consider the very generic resource classes and relation classes, and we assume the detailed ontology or taxonomy could be found elsewhere.

3.1 Agent Statement

Since web of belief ontology is agent centric, we use a special subclass of rdf:Statement to unify all world facts describing agents. We first define *AgentPredicate* as a subclass of rdf:Property: it takes only Agent as the range of domain property. Then we define *AgentStatement* as a subclass of rdf:Statement: it takes only Agent as the range of subject property and AgentPredicate as the range of predicate property. For instance, a tuple $(\_x, A, know, S)$ means that an agent A knows S, which is an instance of rdf:Statement. The OWL ontology for AgentStatement is listed below, and the complete ontology is available online [2].

[2] http://daml.umbc.edu/ontologies/webofbelief

```
< owl : Class   rdf : ID = “AgentStatement” >
  < rdfs : subClassOf   rdf : resource = “&rdf; Statement”/ >
  < rdfs : subClassOf >
    < owl : Restriction >
      < owl : onProperty   rdf : resource = “&rdf; subject”/ >
      < owl : allValuesFrom   rdf : resource = “&wobm; Agent”/ >
    < /owl : Restriction >
  < /rdfs : subClassOf >
  < rdfs : subClassOf >
    < owl : Restriction >
      < owl : onProperty   rdf : resource = “&rdf; predicate”/ >
      < owl : allValuesFrom   rdf : resource = “#AgentPredicate”/ >
    < /owl : Restriction >
  < /rdfs : subClassOf >
< /owl : Class >
```

3.2 Agent knowledge

Among those relations between agent and world fact, we are interested in how to represent an agent's knowledge state over the world facts. We argue that an agent can build its knowledge base by customizing the big RDF graph of the entire Semantic Web. In this process, the agent's knowledge base is derived from the logical RDF graph but not directly from the physical RDF/XML files. In practice, an agent only selects the world fact it knows into its knowledge base, and complex world fact should be fully copied.

3.2.1 *Know and Ignore*

In the agent world, an agent A's knowledge state refers to its awareness of a certain world fact S, i.e., whether S is within A's knowledge base. There are two knowledge states: *know* and *ignore*. These two states are disjointed and complement to one another. It is notable ignore means being ignorant of but not neglecting.

3.2.2 *Believe, disbelieve and nonbelieve*

Quine suggested [38] that an agent can have three exclusive belief states about a world fact S: *believe*, *disbelieve*, and *nonbelieve*. By assuming the equivalence of “believe” and “know” in the sense of agent awareness of a world fact, we can represent these belief states with “know” and “ignore”. Given an agent A and a world fact S, A *believe* S can be represented by $(\_x, A, know, S)$ which means that S is in A's knowledge base; A *disbelieve* S can be represented by $(\_x, A, know, (\_y, A, ignore, S))$ which means that S should never be in A's knowledge base; A *nonbelieve* S can be represented by the union of $(\_x, A, ignore, (\_y, A, know, S))$ and $(\_m, A, ignore, (\_n, A, ignore, S))$ which means that A has not yet come in contact with S.

3.2.3 *Unknown and ignorance*

The seemingly infinite amount of world facts in the Semantic Web and the recursively defined agent statement make it computationally impossible for an agent to enumerate all its possible belief states. Not all nonblieve relations (NRs) could be perceived and memorized by an agent. Therefore, we call the perceived NRs as *unknown* and the unperceived NRs as *ignorance*. Though NRs do not help much in deriving deterministic world facts, they could help an agent to detect knowledge inconsistency. Therefore, it is appropriate to maintain some NRs with caching mechanism.

3.2.4 *Axioms*

There are also some axioms (or inference rules) in the SRM which helps us remove redundant facts and generate new facts. These axioms are common sense knowledge throughout the agent world. However, the rule representation languages, such as Rule ML [3] and OWL Rules Language [4], are still under discussion in the Semantic Web activity. So for the time being, we simply leave these axioms in rule form and let the inference engine apply them in reasoning. Grounded by Modal Logic [26, 18, 25], we describe some axioms for Agent statement as the following. A is an agent. P, Q and S are instances of World fact.

Necessitation axiom The necessitation axiom shows that an agent A knows all world facts that are valid to it. S_A means that S is valid to A, i.e., S could be used in inference by A. Since we already assume that axioms are the only common sense knowledge in the agent world, they are valid to and thus known by all agents.

$$S_A \rightarrow (\_x, A, know, S)$$

Distribution axiom The distribution axiom shows that an agent A could perform Modus Pones inference in its knowledge base. Since RDF is defined within propositional logic and any rdf:Statement instance is also a propositional statement, Modus Pones inference is guaranteed.

$$if\ (\_x, A, know, P)\ and\ (\_y, A, know, (\_z, P, entail, Q)\ then (\_r, A, know, Q)$$

Knowledge axiom The knowledge axiom shows that all world facts known by an agent A are also valid to A.

$$(\_x, A, know, S) \rightarrow S_A$$

Positive introspection axiom The positive introspection axiom shows that A is aware of what statements are in its knowledge base, i.e., if S in A's knowledge base, A knows that.

$$(\_x, A, know, S) \rightarrow (\_y, A, know, (\_x, A, know, S))$$

3.2.5 *Inconsistency detection*

The exclusive nature of knowledge forbids an agent to have dual knowledge states on the same world fact S, i.e. $(\_x, A, know, S)$ and $(\_y, A, ignore, S)$ can't coexist in any agent's knowledge base. Moreover, when we consider the entailment relations between statements, we will face a more complex situation. Suppose that agent A knows $(\_x, P, entail, Q)$, where P and Q are two world facts, we list the consistency status of all possible combinations of A's knowledge states about P and Q in table 1. According to *distribution axiom*, inconsistency only occurs when A knows both $(\_z1, A, know, P)$ and $(\_y2, A, ignore, Q)$.

Table 1: A's knowledge consistency status when P entails Q

	$(\_y1, A, know, Q)$	$(\_y2, A, ignore, Q)$
$(\_z1, A, know, P)$	yes	inconsistent
$(\_z2, A, ignore, P)$	yes	yes

3.3 Agent Capability

Agent capability answers what action an agent can execute. The Semantic Web Services Activity suggests that “A Web service is viewed as an abstract notion that must be implemented by a concrete agent”, i.e., web services are actions that allow agents to participate distributed collaboration with explicit interface description.

[3] http://www.dfki.uni-kl.de/ruleml/

[4] http://www.daml.org/rules/proposal/

Although an agent can execute many types of actions, we concentrate on agent capability in the Semantic Web scale collaboration, especially in providing/composing web services.

3.3.1 action, collaboration and protocol

An *action* is an event executed by an agent actor (initiator); a *collaboration* is a process that a group of agents interact to achieve some goals; and a *protocol* is a set of rules that describe and standardize a certain collaboration process. There are some well known protocols such as FIPA [5] Agent Interaction Protocol, OASIS [6] Business Transaction Protocol. In practice, a protocol can be reduced to a set of agent communicative actions in partial temporal/casual order. For instance, a request interaction protocol includes a request communicative action initiated by sender and then an inform communicative action executed by the receiver as response.

3.3.2 Can

Capability is a hypothesis if an agent can perform any instance of Action. In our ontology represents, an agent's capability is represented as an hypothetical relation $(\_x, A, can, AC)$ where A refers to an agent and AC refers to a subclass of Action (esp. communicative action). It could be derived from A's past actions or simply nowhere, and it usually predicts A's future action as well as summarize its past action instances.

3.3.3 communicative action

Representing an action has been long studied in AI literature [6, 17, 28], especially in planning. Our ontology focuses on communicative actions in agent collaboration context. A *communicative action* (CA) is an agent action that enables an agent to communicate with other agents to fulfill certain purpose as a protocol step.

It is notable that language plays an important role in CA –"two basic ways in which language serves us are these: as a means of getting others to do what we want them to, and as a means of learning from others what we want to know"(Quine) [38]. CA can be represented, based on speech act theory [3], as the combination of *speech act type* which captures the unique semantics of CA, and *content* which records the message, i.e., either world facts or action commands, passed in CA. An important contribution of speech act theory is that it develops the speech act semantics taxonomy, which enables researchers to better capture the meaning of a CA. One well known practice is KQML [10] which is then included in FIPA ACL [11], but it neither covers all speech act semantics nor provides formal explanation for speech act semantics [6]. Elaborating the speech act taxonomy is beyond this paper's scope, and we will only list some frequently used CAs as the following:

- A *request* CA enables an agent A to command another agent B taking a certain action X. It assumes that B can understand and execute X. In addition to action request, the content part might include some world facts as credentials for B's authorization process. Normally, B needs to acknowledge A if it can execute X and then inform A the action result if agreed.

- A *query* CA enables an agent A to ask another agent B a question Q. It assumes that B understands Q. Query is similar to request, and the major difference is that the content of query is question/knowledge instead of action command.

- An *agree* CA enables an agent A to positively acknowledge another agent B's request/query. It shows that A understands and will process B's request/query.

- A *refuse* CA enables an agent A to negatively acknowledge another agent B's request/query. It shows that A will not process B's request/query.

- An *inform* CA enables an agent A to tell another agent B about its knowledge. It could be used in the protocols where A voluntarily informs B its knowledge, or the protocols where A needs to send back some knowledge as the results of B's request/query. Inform is different to request and query in that it doesn't necessarily require acknowledgement or reply.

[5]http://fipa.org

[6]http://www.oasis-open.org/

3.4 Grouping

Grouping is a widely used technique for reducing representation complexity. By grouping entities according to their similarities, we can compress the large amount of individual-individual relations into several individual-class or class-class relations. Additionally, grouping enables us to define hypothesis – an important way to estimate the future world facts or actions which do not yet exist.

The flexible OWL class constructs, especially "restriction", enable us to define an anonymous class based on common properties. Instead of predefining a complex class hierarchy, we adopt the annotation approach that attaches topic (i.e., classification information) to each rdf:Statement instance. This technique is especially useful for grouping relations when their semantics are too complicated to be expressed by a class hierarchy. An important research issue is to derive topics for complex world fact which has embedded statements. The simplest approach is to union the topics of all the embedded statements. Beside the "Topic" class and the "aboutTopic" property, our ontology also defines "subTopicOf" property for more expressiveness.

4. THE TRUST ENHANCED ONTOLOGY

An agent A can explicitly represent its knowledge about the agent world using the core ontology, but how can A make use the agent statement $(\_x, A, know, (\_y, B, inform, S))$, in other words, can A add agent B's utterance S into its knowledge base? *Trust* is the key to this problem. With the explicitly represented trust knowledge, agents can collaboratively adapt knowledge from other agents and selectively plan inter-agent collaborations.

4.1 Characterizing trust

We develop our trust ontology by reviewing the characteristics of trust exhibited in the human world [12] and the agent world [15, 27] as the following:

- From *knowledge representation* aspect, trust is an agent's personal situational rational belief about other agents. Trust is usually used with certain restrictions, e.g., I would only trust a doctor's medical expertise. Trust usually serves for personal use and is not always sharable, e.g., I can't trust a car dealer just because someone trusts him/her.

- From *cognition* aspect, trust is an agent's subjective hypothesis. Trust is learned by an agent from its observations about the world, and it reflects the agent's hypotheses about the world – explaining the past and predicting the future. The commonly used forms of hypotheses are statistical distributions and logical predicates.

- From *sociology* aspect, trust is inter-agent social relation. It interconnects agents to form a social network. It is also a meta-level social relation in the agent world, i.e., many other social relations are dependents of trust relation.

4.2 Representing trust

The traditional agents simply assume *blind trust* – agents trust one another by default. Then the security researchers started using *absolute trust* – trust is explicitly represented as agent-agent relation with binary trust value, and the atom trustable object is agent. Then the trust management researchers suggested *policy based trust* – trust is practically captured by policies which take credentials as proofs of trust, and the atom trustable object is agent action. However, it is difficult or impossible for these approaches to finely capture the rich semantics of trust, e.g., "a doctor is trusted to be knowledgeable only in medical area", "a mechanic is trusted to provide a good car maintenance service", and "the bank approves a withdraw transaction because it trusts the customer who has provided the correct account number and personal identification number (PIN)". To address this difficulties, our ontology provide the highly expressive *trustable statement* to capture trustable objects and the *trust statement* to represent the directed inter-agent trust relation.

trustable statement A *trustable statement* refers to the hypothetical relations starting from an agent, such as the estimation of agent knowledge. Given an instance of trustable statement, $(\_y, B, tp, SC))$, it is firstly a hypothesis about agent B, the semantics of the relation is represented by the agent predicate tp (such as knowledgeable, skillful, and cooperative), and SC refers to an class-resource (such as subclass of action or world fact). A trustable statement MUST be used as the embedded part of a trust statement.

trust statement A *trust statement* is a complex agent statement that explicitly represent the directed trust relation between two agents. It takes trustable statement as the range of its object property.

$$(\_x, A, trust, (\_y, B, tp, SC))$$

4.3 Trust for web services

The semantics of trust is usually determined by application context. For the applications related to Internet services, Grandison and Sloman [15] identified five types of trust: delegation, resource access control, provision of service, certification of trustee, and infrastructure trust. Inspired by this application centric trust classification, we suggest a leveled trust classification for web service as shown in Table 2. We assume that a web service only runs under only one interaction protocol. Additionally, we assume that any complex web service, which is composed of multiple other web services, runs under only one complex interaction protocol.

Table 2: Trust in web services

Level	Trust relations
operation	assurance, authorization
protocol	cooperative, accomplishable, skillful
knowledge	knowledgeable, honest, believable

These three levels of trust serve different purposes. *Operation* trust focuses on one agent interaction: service requestors need to be assured about the service providers' credibility before sending private information, and the service providers need to trust the requestors' eligibility before authorizing service access. *Protocol* trust focuses on one interaction protocol: a protocol initiator selects the services providers according its trust knowledge about their qualities of service (QoS). *Knowledge* trust focuses on knowledge sharing: upon receiving a piece of knowledge S from another agent B, an agent A needs to trust B before incorporating S into A's knowledge base. In the remaining section, we will elaborate protocol trust and knowledge trust.

4.3.1 protocol trust

At protocol level, we are interested in an agent's opinion about the QoS information of a service provider. We assume that the atom web services are running under either FIPA Request protocol or FIPA query protocol. We also simplify web service access protocol by assuming it exception free. Therefore, a common scenario in web service access runs in the following sequence: (1) an initiator agent A requests a service provider SP to execute an action or queries SP for some knowledge, (2) SP agrees/refuses processing the request or query, (3) if agreed, SP will execute the action or process the query, and finally inform A about the result. In such scenario, we identify three types of commonly used protocol trust that characterize the quality of a web service provided by a certain agent.

- *cooperative*: we use $(\_p1, SP, cooperative, WS)$ to represent if SP always agrees to process incoming service request on a class of web service WS. For example, the connection success rate indicates if a telephone base customer service is cooperative. Since agreeing a request is normally the result of authorization, which highly depends on the credentials supplied by the service requestor, cooperative is subjective and is hard to be shared among service requestors.
- *accomplishable*: we use $(\_p2, SP, accomplishable, WS)$ to represent if SP always fulfills the agreed service requests on a class of web service WS. For example, the fight cancellation rate indicates if an airline company is accomplishable in managing its service. Since being able to accomplish diverse service requests is mostly determined by the service provider, accomplish is relatively objective and easy to be shared among service requestors.
- *skillful*: we use $(\_p2, SP, skillful, WS)$ to represent if a SP could perfectly fulfills the agreed service requests on a class of web service WS. For example, the on-time flight rate indicates if an airline company is skillful in managing its service. It is notable that there might be many ways to fulfill a service request, but only those with good performance are skillful. For example, both the rookie drivers and the experienced drivers could drive, but only the latter are skillful because they can drive more smoothly and safely. In addition, different agents might have their own utility functions to evaluate the service result, and they could make skillful assessment differently. For example, suppose that the answers to "where are you working" could be "Earth", "Maryland", or "UMBC", "Maryland" is treated as skillful answer to an international friend, but it is not skillful answer to a USPS mail person. Therefore, skillful is a highly subjective concept and could be shared only when the meaning is clearly determined.

Now we use a travel market example to explain how protocol trust could be used in web service planning. There are many online travel services such as *orbitz.com, travelocity.com, hotwire.com*, and we assume that their web service interfaces is similar to their web interfaces. It is obvious that most of them are cooperative since these websites are always ready for travel schedule query on the web. The only exceptions are that some services might request the users to be registered first. In addition, most of them could accomplish the query and inform the users some results. However, their skillful assessments have different meaning: Travelocity is skillful because its flexible query interface helps customers finding best fare without specifying departure/arrival days; Hotwire is skillful because it helps customers finding high quality hotels with lower cost; and Orbitz.com is skillful because it helps customers finding cheap multiple destination flights. In this example, being cooperative and accomplishable are both a MUST feature of web service, however, being skillful is an optional feature and its meaning should be disambiguated before sharing.

4.3.2 knowledge trust

Knowledge trust is specially designed for knowledge sharing protocols, such as FIPA query protocol and FIPA inform protocol. In knowledge sharing protocols, a common scenario is that agent B informs agent A about some statements. In this scenario, A's trust knowledge reflects A's opinion on the hypotheses of B's knowledge base or B's utterance (i.e., the statements informed by B). We also notice that the semantics of those hypotheses are also characterized by axioms. Our ontology supplies three types of knowledge trust as the following:

- *knowledgeable* shows an agent's knowledge completeness, e.g., $(\_k11, A, trust, (\_k12, B, knowledgeable, SC))$ means that agent A trusts that agent B's knowledge base has most statements of statement class SC (Since a statement class might have infinite members, we use "most" to instead of "entire", and the meaning of most is defined by heuristics, such as "B can answer 90 percents of A's questions"). Additionally, we restrict the meaning of knowledgeable such that it doesn't imply believable. For example, a car dealer might be knowledgeable in car performance, but a customer can't fully trust his honesty or believe all his words. The semantic of knowledgeable is shown by axiom 1:

$$\begin{aligned}&(\_k11, A, trust, (\_k12, B, knowledgeable, SC))\\ &\wedge(\_k13, S, \mathrm{rdf:type}, SC)\\ &\rightarrow (\_k21, A, know(\_k22, B, know, S))\\ &\quad\vee(\_k31, A, know(\_k32, B, know, (\_k34, B, ignore, S)))\end{aligned} \tag{1}$$

- *honest* shows if an agent's utterance conforms to its knowledge, e.g., $(\_f11, A, trust, (\_f12, B, honest, SC))$ means that any B's utterance S in statement class SC should be in B's knowledge base. For example, an honest kid might not be knowledgeable to tell Columbus' discovery, and he/she might even honestly tell "1+1=3". The semantic of honest is shown by axiom 2:

$$\begin{aligned}&(\_f11, A, trust, (\_f12, B, honest, SC))\\ &\wedge(\_f13, S, \mathrm{rdf:type}, SC) \wedge (\_f14, B, inform, S)\\ &\rightarrow (\_f21, B, know, S)\end{aligned} \tag{2}$$

- *believable* shows if an agent B's utterance will conform to the observer agent A's knowledge base, e.g., $(\_t11, A, trust, (\_t12, B, believable, SC))$ means that any statement S in statement class SC informed by B is true to A (i.e., A can add S to its knowledge base). For example, a kid might trust his/her teachers as a believable information source without considering honesty or knowledgeable issues. The semantic of honest is shown in the axiom 3:

$$\begin{aligned}&(\_t11, A, trust, (\_t12, B, believable, SC))\\ &\wedge(\_t13, S, \mathrm{rdf:type}, SC) \wedge (\_t14, B, inform, S)\\ &\rightarrow (\_t21, A, know, S)\end{aligned} \tag{3}$$

We use an example to demonstrate the expressiveness of our trust ontology. The preconditions and post conditions are shown in Table 3. There are three agents A1, A2, and A3, and S is a statement. When A1 informs A2 about S, A2 knows $(\_ex11, A1, know, S)$ according to honest axiom. However, A2 remains ignorant about S. Then A2 informs A3 about $(\_ex11, A1, know, S)$, and A3 knows $(_ex23, A2, know, (\_ex11, A1, know, S))$ according to the honest axiom. Moreover, according to believable axiom, A3 learns S since it trusts both A2 and A1 believable. An interesting observation is that even all three agents know the statement $(\_, A1, know, S)$, not all of them know S. In fact, A2 causes A3 to know S, but A2 itself doesn't know S.

Table 3: Knowledge trust example

Precondition: Common sense: $(\_ex01, S, rdf:type, SC)$ A1: $(\_ex11, A1, know, S)$ A2: $(\_ex21, A2, trust, (\_ex22, A1, honest, SC))$ A3: $(\_ex31, A3, trust, (\_ex32, A2, honest, SC))$ $(\_ex33, A3, trust, (\_ex34, A2, believable, SC))$ $(\_ex35, A3, trust, (\_ex36, A1, believable, SC))$
Post condition (only new facts listed): A1: nothing new A2: $(\_ex11, A1, know, S)$ A3: $(\_ex11, A1, know, S)$, S

5. MAINTAINING THE WEB OF BELIEF

The web of belief is indeed a large scale open knowledge grid in the agent world, and it has direct applications in distributed inference. However, how do we maintain the web of belief in an open agent world? In the rest of this section, we will briefly discuss three important issues.

5.1 Maintaining agent memory

Although all world facts are distributed in the agent world, an individual agent's physical memory limited the amount of statements in its knowledge base. Since the axioms discussed in the section 3 and 4 could generate infinite amount of knowledge, maintaining a minimum set of knowledge for backward chaining inference is preferred. The redundant statements could be removed by using knowledge axiom.

5.2 Maintaining trust network

In an open agent world, new world facts are generated all the time, which result in the change of agents' knowledge. By dynamically maintaining trust knowledge estimating the other agent's up-to-date knowledge, an agent could make better collaboration choice. Due to the bounded storage/computation resources, the political reasons, and the privacy considerations, it is neither possible nor necessary to require each agent to independently maintain trust knowledge about all other agents' knowledge bases. A *trust network* [14, 32] enables agents to derive peer-to-peer trust relation by sharing other agents' trust knowledge. The trust network is part of the web of believe, and it also enables the web of belief to run in a distributed environment. The computational complexity of deriving a peer-to-peer trust relation could be guaranteed by its scale-free network topology [2] with the small world property [29, 36].

5.3 Inconsistency handling

As discussed in the section 3, inconsistency happens when an agent concurrently holds different knowledge states on the same world fact. By assuming that every agent's knowledge base is consistent, there are two common reasons that cause inconsistency: (1) Change of trust. Since trust is a dynamically learned hypothesis, a trust relation could valid at time t but invalid at time (t+1). For example, a patient might stop trusting a doctor after an unsuccessful surgery. (2) Inaccurate trust. An agent might learn a statement inconsistent to its knowledge base from an inaccurately trusted agent. For example, it is inappropriate for a patient to trust a doctor's financial expertise.

To deal with the change of trust, it is desired not to cache any trusted world facts derived from trust axioms. However, that will also increase the complexity of detecting inaccurate trust, which requires forward chaining inference in an agent's knowledge base. As long as trust being a dynamic hypothesis, it is impossible and unnecessary to strictly maintain its accuracy. We suggest a simple inconsistency handling heuristic: an agent use limited memory to cache the world facts derived from trust axioms; it only checks the consistency between newly learned knowledge with cached derived knowledge and the its own knowledge; and it revises trust knowledge and flush out invalid derived world facts when inconsistency is detected. Though this heuristic might not detect all inaccurate trust, it can intuitively keep the frequently used trust knowledge accurate at the cost of constant time for consistency validation of each new statement.

6. RELATED WORKS

The general philosophical background of the *Web of belief* is the Modal Logic [26, 18, 25]. In AI field, McCarthy's *Some Philosophical Problems from the Standpoint of Artificial Intelligence* [28] first brought logical formalization of agent and action. Cohen and H. Levesque's *Intention Is Choice with Commitment* [5] developed an influential formalism of belief and intention. A modal operator may stand for many different semantics in literature: logic of knowledge – epistemic logic [35]; logic of belief – doxastic logic; logic of obligation – deontic logic; and etc. [30]. A modal operator is also associated with some inference axioms as the following:

$$\begin{array}{ll} \text{(NEC)} & \mathrm{From} \alpha \mathrm{isatheoremof} K, \mathrm{infer} K\alpha (\mathrm{necessitation}) \\ \text{(K)} & K\alpha \wedge K(\alpha \rightarrow \beta) \rightarrow K\beta \quad (\mathrm{distibutionaxiom}) \\ \text{(T)} & K\alpha \rightarrow \alpha \quad (\mathrm{knowledgeaxiom}) \\ \text{(D)} & K\alpha \rightarrow \neg K\neg\alpha \quad (\mathrm{knowledgeaxiom}) \\ \text{(4)} & K\alpha \rightarrow KK\alpha \quad (\mathrm{positive-introspectionaxiom}) \\ \text{(5)} & \neg K\alpha \rightarrow K\neg K\alpha \quad (\mathrm{negative-introspectionaxiom}) \end{array}$$

There are two popular formalisms of agent knowledge in literature: epistemic modal logic S5, which uses "knows" as modal operator and support axioms NEC, K, T, 4, 5; and doxastic modal logic KD45, which uses "believes" as modal operator and support axioms NEC, K, D, 4, 5. By dividing formula into two categories: true formula and false formula, the important distinction between "knows" and "believes" is that an agent only "knows" true formula but it can "believe" any formula. However, we argue that axiom 5 (negative-introspection axiom) takes close world assumption and might not be appropriate since an agent can't fully enumerate/know its ignorance. Therefore, our ontology is basically S4, which only supports axioms NEC, K, T, 4.

Logical formalisms, such as First Order Logic and Modal Logic, have also been used to represent and reason about trust in distributed agent society context [15]: Regan's formal framework of belief and trust [31]; Josang's subjective logic approach [23, 24]; Abdul-Rahman's social trust model [1]; and Jones and Firozabadi's integrated logic model [21] which uses Kanger-Porn-Lindahl's modal logic for action [9], Jones-Sergot's situation conditional logic [22], and deontic logic. These logical formalisms have suggested high level theories for the *Web of belief*, however, lots of real-world issues still need to be considered, such as computational complexity and limited resource [8], dynamical agent knowledge, inconsistent handling in open agent world. Our work, including the SRM and the web of belief ontology, aims to ground these theoretical formalisms of trust to the real applications in the Semantic Web context.

7. CONCLUSIONS AND FUTURE WORK

The web of belief is a general framework enabling explicit representation of the rational beliefs and facts about the agent world in the Semantic Web, and it serves as the distributed knowledge base for knowledge sharing and agent collaboration. We limited the applicable scope of the web of belief within the agent world in the Semantic Web because such world is a symbolic world which is less complex than human society. The web of belief is characterized by the following features:

- *Semantic relation model*. The semantic relation model is the basis of the web of belief. It models the agent world in the Semantic Web with three basic notions (1) resource, which captures agent, world fact, and action in class level and individual level; (2) relation, which captures know, can, trust and etc. as semantic relations between entities and is represented as subclass of world fact; and (3) axiom, which enables agents to manipulate world facts in RDF graph. The semantic relation model is more general than model logic representation since it doesn't limited in representing agent knowledge and agent capability.

- *Explicit trust*. By explicitly representing and using trust, which is located on the top of Tim-Berners Lee's Semantic Web layered cake, knowledge sharing and agent collaboration could be better facilitated: *protocol level trust* enables agents to selectively use existing web services, *knowledge level trust* enables agents to selectively use other agents' knowledge.

- *Semantic Web representation*. Our web of belief ontology is written in OWL and thus enables a "written" version of the web of belief in the Semantic Web (though the axioms can't be explicitly represented because of the undecided rule ontology language). By linking agents with world facts, our ontology also explicitly captures *data provenance* [4], i.e., who knows what. Additionally, we also suggest using the provenance information to help agents build their knowledge base and share knowledge. The web of belief ontology is available at http://daml.umbc.edu/ontologies/webofbelief/.

- *Support distributed inference*. The explicitly represented web of belief also supports distributed inference. We suggest that backward chaining inference is preferred due to the limited computation and storage resource. Caching trusted world facts can be used to detect inaccurate trust. In addition, both machine learning and logical reasoning could be used to dynamically maintain trust knowledge [7].

Our preliminary work on the web of belief focuses on the representation part, our work on dynamic trust network maintenance can be found in [7]. In the future, we aim to provide refine the web of belief ontology, to implement the computational inexpensive distributed inference model, and to experiment the web of belief in information access context and web service composition context.

8. ACKNOWLEDGMENTS

Thanks go to colleagues in UMBC including Dr. Lina Zhou, Filip Perich, Dipanjan Chakraborty, Dr. Anupam Joshi, Marriott Nielson, and Zhongli Ding for valuble comments. This work was supported in part by DARPA contract F30602-00-2-0591.

9. REFERENCES

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *HICSS*, 2000.

[2] R. Albert, H. Jeong, and A. Barabasi. Diameter of the world wide web. *Nature*, (401), 1999.

[3] J. Austin. *How to Do Things with Words*. Oxford University Press, 1962.

[4] P. Buneman, S. Khanna, and W.-C. Tan. Why and where: A characterization of data provenance. In *International Conference on Database Theory (ICDT)*, 2001.

[5] P. Cohen and H. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–361, 1990.

[6] P. R. Cohen and H. J. Levesque. Communicative actions for artificial agents. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95)*, pages 65–72. The MIT Press: Cambridge, MA, USA, 1995.

[7] L. Ding, L. Zhou, and T. Finin. Trust based knowledge outsourcing for semantic web agents. In *IEEE/WIC International Conference on Web Intelligence*, 2003.

[8] H. N. Duc. *Resource-Bounded Reasoning about Knowledge*. PhD thesis, Leipzig University, 2001.

[9] D. Elgesem. The modal logic of agency. *Nordic Journal of Philosophical Logic*, 2(2):1–46, 1997.

[10] T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an Agent Communication Language. In *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, pages 456–463. ACM Press, 1994.

[11] Foundation for Intelligent Physical Agents. Fipa acl message structure specification, 2002.

[12] D. Gambetta. *Trust: Making and Breaking Cooperative Relations*. Department of Sociology, University of Oxford, 2000.

[13] L. Garton, C. Haythornthwaite, and B. Wellman. Studying online social networks. *Journal of Computer Mediated Communication*, 3(1), 1997.

[14] J. Golbeck, B. Parsia, and J. Hendler. Trust networks on the semantic web, 2003.

[15] T. Grandison and M. Sloman. A survey of trust in internet application. *IEEE Communications Surveys & Tutorials (Fourth Quarter)*, 3(4), 2000.

[16] R. Guttman, A. Moukas, and P. Maes. Agent-mediated electronic commerce: A survey. *Knowledge Engineering Review*, 13(2):147–159, 1998.

[17] J. Y. Halpern and R. Fagin. A formal model of knowledge, action, and communication in distributed systems: preliminary report. In *Proceedings of the fourth annual ACM symposium on Principles of distributed computing*, pages 224–236. ACM Press, 1985.

[18] J. Hinitikka. *Knowledge and Belief*. Conell Univerisity Press, 1962.

[19] N. R. Jennings, P. Faratin, M. J. Johnson, P. O'Brien, and M. E. Wiegand. Agent-based business process management. *International Journal of Cooperative Information Systems*, 5(2,3):105– 130, 1996.

[20] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.

[21] A. Jones and B. S. Firozabadi. *On the characterisation of a trusting agent - aspects of a formal approach*, pages 157–168. Kluwer Academic Publishers, 2001.

[22] A. J. Jones and M. Sergot. A formal characterisation of institutionalised power. *Journal of the IGPL*, 4(3):429–445, 1996.

[23] A. Josang. Artificial reasoning with subjective logic. In *the Second Australian Workshop on Commonsense Reasoning*, 1997.

[24] A. Josang. Prospectives for modelling trust in information security. In *Australasian Conference on Information Security and Privacy*, pages 2–13, 1997.

[25] S. Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

[26] C. Lewis. *A Survey of symbolic logic*. Univerisity of California Press, 1918.

[27] S. P. Marsh. *Formalising trust as a computational Concept*. Ph.D. dissertation, University of Stirling, 1994.

[28] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[29] S. Milgram. The small world problem. *Psychology Today*, 1(1):60–67, 1967.

[30] N. J. Nilsson. *Aritificial Intelligence, A New Synthesis*. Morgan Kaufmann, 1998.

[31] P. V. Rangan. An axiomatic basis of trust in distributed systems. In *Proc. IEEE Symposium on Security and Privacy*, pages 204–211, 1988.

[32] M. Richardson, R. Agrawal, and P. Domingos. Building the semantic web by mass collaboration, 2003.

[33] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92, 1993.

[34] W. van der Hoek and M. Wooldridge. Towards a logic of rational agency. *Logic Journal of the IGPL*, 11(2):133–157, 2003.

[35] W. van der Hoek Rineke Verbrugge. Epistemic logic: A survey. *Game Theory and Applications*, pages 53–94, 2002.

[36] D. Watts. *Small worlds : the dynamics of networks between order and randomness*. Princeton University Press, Princeton, 1999.

[37] M. Wooldridge and N. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

[38] W.V.Quine and J.S.Ullian. *The Web of Belief*. Random House, second edition, 1978.