# Trust Model for Semantic Sensor and Social Networks: A Preliminary Report

Pramod Anantharam, Cory A. Henson, Krishnaprasad Thirunarayan and, Amit P. Sheth
Ohio Center of Excellence in Knowledge-enabled Computing (Kno.e.sis),
Wright State University,
Dayton, Ohio, USA.
{pramod, cory, tkprasad, amit}@knoesis.org

***Abstract*—Trust is an amorphous concept that is becoming Increasingly important in many domains, such as P2P networks, E-commerce, social networks, and sensor networks. While we all have an intuitive notion of trust, the literature is scattered with a wide assortment of differing definitions and descriptions; often these descriptions are highly dependent on a single domain or application of interest. In addition, they often discuss orthogonal aspects of trust while continuing to use the general term "trust". In order to make sense of the situation, we have developed an ontology of trust that integrates and relates its various aspects into a single model.**

## I. Introduction

A trust relationship may exist between people (e.g., social networks), between two machines or agents (e.g., sensor networks) and, between people and machines (e.g., E-commerce). The goal of this paper is to illustrate a trust ontology that integrates and relates various aspects of trust within several domains;- allowing us to represent, organize and reason over trust. Within this ontology, we model a general trust relationship between two agents, the *trustor* and the *trustee*, that distinguishes between the semantics of the trust relationship, the scope of interest, the quantitative or qualitative value, and the method of creating and maintaining this value. The Trust Ontology is encoded in the Web Ontology Language (OWL) [13]. Section II talks about the motivation for our work. Section III discusses the general trust model and illustrates its application to social networks and sensor networks, Section IV describes the ontology in concrete terms by describing the classes and properties in the OWL ontology, Section V shows a sample query that can be executed against a knowledge base using this trust ontology. Finally, we conclude with Section VI discussing future research directions.

## II. Motivation

The large amounts of data being generated in sensor networks and social networks is becoming increasingly difficult to manage. Even though the openness of web has substantial benefits for sharing information and opinions, it has created problems with regards to data quality. Sensor networks often employ large numbers of low cost sensors as opposed to a few expensive high fidelity sensors, also leading to low quality data. For these reasons we often rely on middleware  a layer between the network generating data and the applications that consumes the data – for improving overall quality and reliability of data. The work presented in this paper adds value to this middleware through a formal representation of trust. There has been a lot of work on trust in social networks and sensor networks. The approaches for representing, reasoning and updating trust values are diverse. Some works deal with trust links without considering the context while treating it as implicit [2]. Some works on trust consider trust links of single type [6] while others consider different types of links [7]. Several different approaches to compute trust values include reputation-based systems [5], policy-based systems [12], evidence based systems [9] and entropy based systems [8]. Trust values are represented in diverse ways, such as natural numbers, real numbers and partial orderings [1]. The trust model presented attempts to tie together all different aspects of trust, including those mentioned above. We attempt to capture the semantics of the trust relationship using this trust model and design a trust ontology that serves as an upper level ontology for use across multiple domains. Using this trust ontology, we can ask questions like: What are the trust relationships that an agent is participating ? Is there a trust relationship between agent X and agent Y ? What is the scope of a trust relationship ? What process was used to arrive at this trust value ? These questions are formulated as queries using the trust ontology in Section V.

## III. Trust Model

A model of trust should capture and relate essential aspects of the trust relationship. In this section we will discuss the general trust model and in the next section we provide a concrete realization by defining classes and properties in OWL. We model the trust relationship between two agents as a six tuple relationship *trustor, type, scope, value, process, trustee* (as shown in *Figure 1*).

The trust relationship between two agents is represented as a six tuple. The agent who trusts another agent is called the *trustor* and the agent being trusted is called the *trustee*. Each trust relationship is further qualified with:

*1) Trust Type:* The trust type captures the semantics of the trust relationship. Trust type can be *functional*, *referral* or *non-functional*. Each of these trust types are discussed in detail in [1]. Below is a quick review of these trust types:-

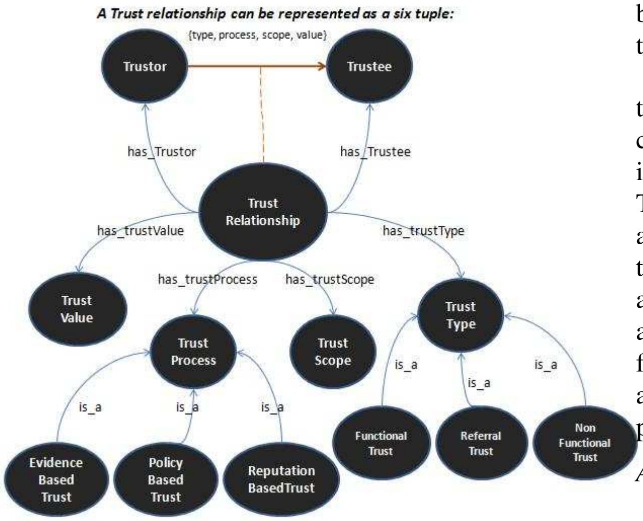


Fig. 1. *Trust Model* illustrating all the concepts and relationships between the concepts.

*a) Functional Trust:* Trust relationship established with direct interactions between two agents. One agent trusts another agent's ability to carry out a particular task.

*b) Referral Trust:* Trust relationship established for conceiving an agent's referral of another agent. An agent trusts another agent's ability to recommend a third agent.

*c) Non-Functional Trust:* Distrust in agent's competence or behavior established.
Note that referral trust is transitive within the same scope, while functional trust is not [1].

*2) Trust Scope:* Trust Scope captures the context in which the trust relationship is valid. A trust relationship is valid only in a prescribed scope. An agent that trusts another agent in one scope may distrust the same agent in another scope. For instance, an agent A can have functional trust in agent B for music and, at the same time, have non-functional trust in agent B for books.

*3) Trust Value:* Trust value is a way to quantify or compare trust relationship. Value can be a natural number, real number in the range [-1, 1], or it a partial ordering [1] of trust relationships.

*4) Trust Process:* The process by which we arrive at trust values is termed as *Trust Process*. The trust process will indicate the way in which trust values are computed and updated, essentially leading to trust management. This can include specific trust computation algorithms and application specific techniques for trust computation, aggregation and management. Some examples of trust processes are described below:

*a) Policy Based Trust:* An agent trusts another agent based on some policy or rules. For instance, if a company is ISO 9001 certified, then we can expect a certain quality enforcement in the products they deliver.

*b) Reputation Based Trust:* If an agent has a record of previous interactions with another agent, then this can act as a basis for inferring trust and this is termed as reputation based trust process.

*c) Evidence Based Trust:* Evidence-based trust is the the process of arriving at trust values by seeking additional confirmatory evidence for a known fact in order to validate or invalidate what is already known.
The idea of trust process is to abstract the method of arriving at trust values and managing them. There is no universal trust algorithm that fits all domains and applications. This abstraction will allow us to talk about trust across domains and use application specific or domain specific trust algorithms for each class of problems. Reputation based algorithms [5] and entropy based algorithms [4] are some examples of trust processes used within sensor networks.

### A. Trust Ontology Applied to Sensor Networks

Sensor networks produce an avalanche of data. Reliability is critical in order to act on the data gathered by these sensor networks. Due to extreme operating conditions and use of inexpensive mass-produced sensors, sensors are error prone and often generate erroneous data. The trust ontology has the capability to define scope of trust relationship. This naturally fits into the sensor network scenario, especially in the domain of weather sensors where each station has different types of sensors. Individual sensors for temperature, pressure, humidity etc, are seldom deployed in practice. All these sensors are housed on a single base called the *mote* which has a battery for power supply driving a circuitry to transfer sensor data to a base station. There may be situations where we may rely on a station for temperature readings but not for pressure readings. Such a fine level of detail can be represented using the proposed trust model.

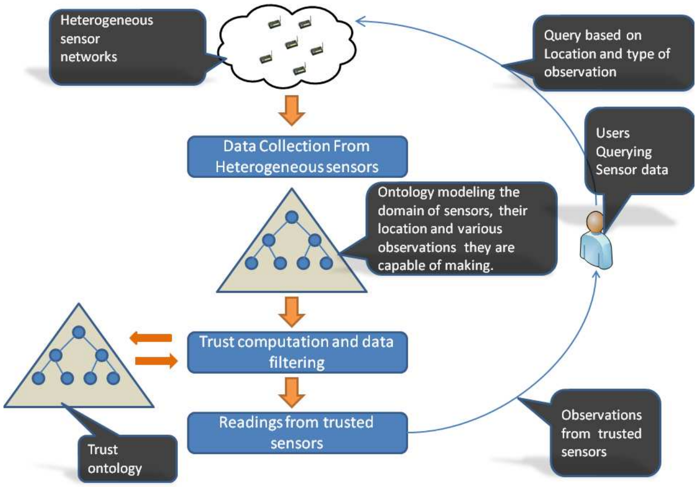

Heterogeneous sensor networks
Query based on Location and type of observation
Data Collection From Heterogeneous sensors
Users Querying Sensor data
Ontology modeling the domain of sensors, their location and various observations they are capable of making.
Trust computation and data filtering
Readings from trusted sensors
Observations from trusted sensors
Trust ontology

Fig. 2. *Trust Ontology* appled to Sensor Network scenario.

### B. Trust Ontology Applied to Social Networks

Trust plays an important role in our everyday life. Consider a scenario of purchasing a product on Amazon.com. Amazon.com contains thousands of vendors and products along

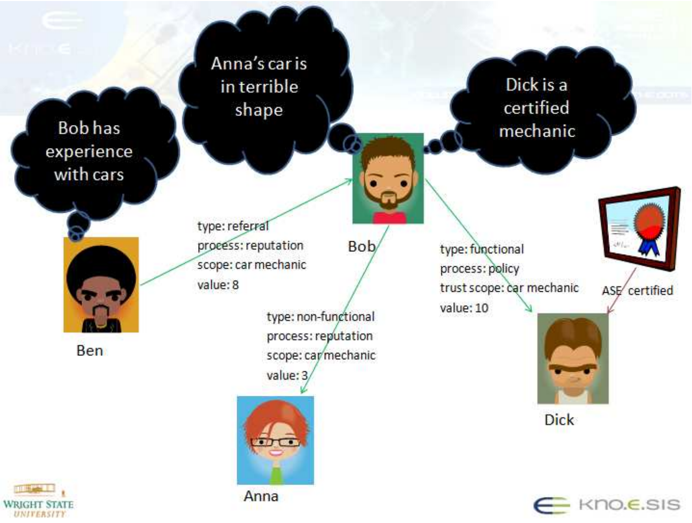


Fig. 3. *Trust Ontology* applied to Social Network Scenario

with their ratings. Any person who wants to make an on-line purchase would first look at the product reviews and ratings. Once the product is finalized, the next step is to decide the vendor from which the product should be ordered. The Trust Ontology allows us to represent such a scenario by modeling the trust network existing within an e-commerce website, such as Amazon.com.

Epinions.com [18] is a website where people can share their experience and opinions regarding products such as books, music, electronics, cars etc., and rate them, justifying their rating with key facts and figures. Twitter.com [19] is a microblogging platform where millions of people share their opinions, observations and perceptions using short 140 character massages. We will discuss how concepts in our ontology can be used in the context of Epinions and Twitter.

### C. Epinions Scenario

The relationships that exist among the users of Epinions can be viewed through the lens of our trust model. For this purpose, we will mention salient Epinions concepts and then show how they can be mapped to concepts within the trust ontology. A *reviewer* is an agent that writes a review for a product and assigning the product a value of 1 to 5 stars. A *Consumer* is an agent that consumes a review and rates the review as *not helpful, somewhat helpful, helpful,* or *very helpful*. A *Category* lead is an agent that takes responsibility for a particular domain like electronics, books etc. A top reviewer is an agent whose reviews are rated as top quality. An *Advisor* is an agent that advices reviewers in a particular domain by providing them with comments on the reviews they write. A *trust circle* is a set of reviewers that are trusted by a consumer. Finally, a *block list* is a set of reviewers who are not trusted by a consumer.
Trust ontology can be applied to Epinions data as follows:
*Trust type:*

- Referral: consumer suggests review to another consumer (e.g., *very helpful, helpful, not helpful*, etc.)
- Functional: consumer believes opinions of reviewer (e.g., *trust circle*)
- Non-Functional: consumer does not believe opinions of reviewer (e.g., *block list*)

*Trust Process:*

- Policy-based: trustworthiness labels sanctioned by the system (e.g., *category leads, top reviewer, advisor* )
- Reputation-based: through aggregate high ratings

*Trust Value:* boolean values *(+1, -1)*
*Trust Scope: movies, books, electronics,* etc.

### D. Twitter Scenario

Before we show how Twitter can be viewed through the lenses of our trust model, we would like to introduce a few salient concepts of Twitter. Twitter consists of a network of users who generate and distribute small messages called *tweets*. *Suggested users* are a set of users who are well known and have been recommended by Twitter. *Follow* relationships exists between users if one user opts to receive tweets from another user. *Re-tweet* is a way of forwarding a message to another user. *Lists* groups people with similar interests. *Hashtags* are annotations that relate a topic to a tweet.
Trust model applied to twitter data:
*Trust Type:*

- Referral: one user sends another user's tweet (e.g., re-tweet).
- Functional: one user likes the tweets of another user and decides to follow (follow relationship).
- Non-Functional: un-follow may be regarded as an instance of non-functional trust.

*Trust Process:*

- Policy-based: one user follows another user based on some criteria (e.g., suggested user, affiliation)
- Reputation-based: one user follows another user based on past behavior (e.g., whose tweets are often re-tweeted)

Trust Value: Not represented within twitter network.
Trust Scope: topic of a tweet (e.g., hashtag topics, twitter lists)

## IV. TRUST ONTOLOGY

Semantic Web [14] technologies and techniques allow us to represent knowledge on the Web. OWL (Web Ontology Language) [13] is a Semantic Web language used to develop ontologies. There are various factors to consider while building a trust ontology, that spans multiple domains. The trust ontology provides for the representation, reasoning and querying over trust relationships existing within a social or sensor network.

### *A. Classes in the Ontology*

**Trust Relationship**: Trust established between two agents is represented using a *Trust Relationship* class. All other attributes of the trust relationship are modelled as properties of this class. Trust links in any social network or sensor network are represented as instances of this class.

**Agent**: Agents can be people or machines. Thus all people and machines within the trust network are represented as instances of this class.

**Trust Type**: *Trust Type* is a class and each of the different trust types – namely functional trust, referral trust and non-functional trust – are represented as subclasses. This finer level of representation will allow us to represent and query trust relationships in a greater detail for inferring transitivity of trust relationships. Referral trust is transitive but functional trust is not. Associating a type to a trust relationship will allow us to capture this in semantics suitably.

**Trust Scope**: The context in which the trust relationship is established is a Class, whose instances contain different scopes (e.g., music, books, car mechanic, electronics etc). Each trust relationship has a scope in which the trust relationship is valid and trust relationships with same scope are candidates for chaining. One agent may trust another agent in a scope (functional trust, scope:car mechanic) and distrust the same agent in a different scope(non-functional trust, scope:books). Figure 3 in Section III gives a visualization of trust relationships existing between agents in a social network. As mentioned earlier, we infer trust relationship by considering the link type and scope. This is intuitive in a real world scenario like referral for a car mechanic. If an agent A1 has referral trust in another agent A2 due to agent A2's extensive experience in the domain of cars, then agent A1 may believe the referrals from agent A2 regarding a car mechanic.

**Trust Process**: We need to consider various ways of creating and maintaining a trust relationship such as reputation, policy, and evidence. The process of generating trust relationships is captured by the trust process which is represented as a class in the trust ontology. Subclasses include reputation based trust, policy based trust, and evidence based trust. Instances of these classes would be specific methods and, implementations of processes that of quantify trust. For example, the policy based trust class can have sub-classes like CMM level certification and ISO certification. The reason for representing each of the trust process methods as classes is to allow the ontology to distinguish between various trust processes involved in establishing a trust relationship.

**Trust Value**: Trust values quantify and ranks trust relationships. Trust value is modeled as a class and can have instances such as numbers. The output of a trust process is generally a value or an ordering of trust relationship.

### *B. Properties*

Properties relate classes in the ontology. We will define all the properties in the trust ontology. These properties are defined using the notation:

P:D → R, where P is a property, D is the domain and R is the range.

**has_trusor**:**Trust_Relationship → Agent**
Given a trust link, it returns a trustor.

**has_trustee**:**Trust_Relationship → Agent**
Given a trust link, it returns a trustee.

**has_trustType**:**Trust_Relationship → Trust Type**
Given a trust link, it returns a trust type (e.g. referral trust, functional trust or non-functional trust).

**has_trustScope**:**Trust_Relationship → Trust Scope**
Given a trust link, it returns Trust Scope (e.g. books, music, cars etc).

**has_trustValue**:**Trust_Relationship → Trust Value**
Given a trust link, it returns trust value (e.g. real numbers, natural numbers etc).

**has_trustProcess**:**Trust_Relationship → Trust Process**
In our ontology, has_policy_based_trustProcess, has_reputation_based_trustProcess and, has_evidence_based_trustProcess are sub-properties of the property has_trustProcess.

**has_policy_based_trustProcess**:**Trust_Relationship → Policy Based Trust**

**has_reputation_based_trustProcess**:**Trust_Relationship → Reputation Based Trust**

**has_evidence_based_trustProcess**:**Trust_Relationship → Evidence Based Trust**

Inferring a triple form a sub-property relationship:
*Given triple:*
:TR1 :has_policy_based_trustProcess :ISOCertification
*Inferred triple:*
:TR1 :has_trustProcess :ISOCertification

## V. Sample Queries

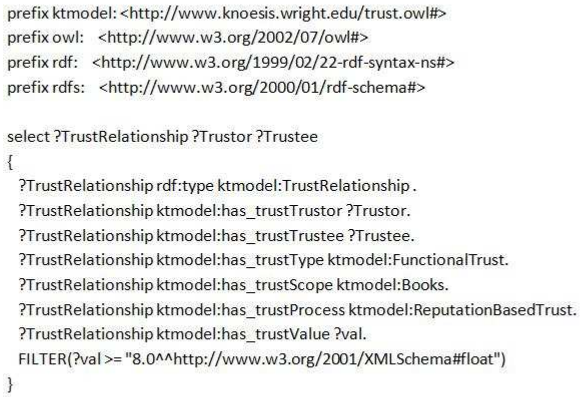
Trust relationships are encoded in the form of triples, using RDF [15] representation and OWL [13] semantics. A sample instance TR1 of trust relationship class is shown below in *turtle* [16] syntax:

```
:TR1 rdf:type :TrustRelationship.
:TR1 :has_trustType :FunctionalTrust.
:TR1 :has_trustProcess :ReputationBasedTrust.
:TR1 :has_trustScope :books.
:TR1 :has_trustValue :10.
:TR1 :has_trustTrustor :Ben.
:TR1 :has_trustTrustee :Bob.
```

This knowledge is stored in a knowledge base that can be queried using SPARQL [17], a query language for querying RDF data. An example query is given below and *Figure 4* is the same query in SPARQL:

"Give all instances of trust relationships and their associated trustor and trustee, that is of type *functional trust* and has scope *books* derived using *reputation based trust* process with a *trust value* greater than or equal to 8".

Trust ontology can be used to formulate such queries and it is available for download at [20].



```
prefix ktmodel: <http://www.knoesis.wright.edu/trust.owl#>
prefix owl:   <http://www.w3.org/2002/07/owl#>
prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#>

select ?TrustRelationship ?Trustor ?Trustee
{
  ?TrustRelationship rdf:type ktmodel:TrustRelationship .
  ?TrustRelationship ktmodel:has_trustTrustor ?Trustor.
  ?TrustRelationship ktmodel:has_trustTrustee ?Trustee.
  ?TrustRelationship ktmodel:has_trustType ktmodel:FunctionalTrust.
  ?TrustRelationship ktmodel:has_trustScope ktmodel:Books.
  ?TrustRelationship ktmodel:has_trustProcess ktmodel:ReputationBasedTrust.
  ?TrustRelationship ktmodel:has_trustValue ?val.
  FILTER(?val >= "8.0^^http://www.w3.org/2001/XMLSchema#float")
}
```

Fig. 4. *Query*

## VI. Conclusion and Future Work

In this paper, we proposed a general ontology of trust that is independent of any specific domain. With this rich representational framework that uses Semantic Web technologies, we are able to represent, reason, query and update trust information. We demonstrated what types of questions can be answered by a knowledge base using this ontology. With this ontology as a foundation, we would like to explore trust processes that can be used to glean trust information from sensor networks and social networks. When dealing with data from various domains we anticipate refinement of the existing model and techniques for trust computation. It would be interesting to combine trust information between social networks and sensor networks. In such a scenario of combining trust information, either one complements the other or serves as an evidence to the other depending on its availability. Though the trust ontology helps to model trust across domains, we need more experimental data to test and refine the trust ontology in an iterative way.

## References

[1] K. Thirunarayan, D. K. Althuru, C. A. Henson, and A. P. Sheth *A Local Qualitative Approach to Referral and Functional Trust*, Proceedings of the The 4th Indian International Conference on Artificial Intelligence (IICAI-09), page(s): 574-588, India, 2009.

[2] J. Golbeck, B. Parsia, J. Hendler, *Trust Networks on the Semantic Web*, Proceedings of Cooperative Intelligent Agents, page(s): 238-249, 2003.

[3] K. Thirunarayan, P. Anantharam, C. A. Henson, A. P. Sheth, *Some Trust Issues in Social Networks and Sensor Networks*, International Symposium on Collaborative Technologies and Systems (CTS 2010), page(s): 573-580, Chicago, IL 2010.

[4] H. Dai and Z. Jia and X. Dong *An Entropy-based Trust Modeling and Evaluation for Wireless Sensor Networks*, Proceedings of the 2008 International Conference on Embedded Software and Systems, page(s): 27-34, Washington, 2008.

[5] S. Ganeriwala, L. K. Balzano, AND M. B. Srivatsava *Reputation-based framework for high integrity sensor networks*, Proceedings of the 2nd ACM Transactions on Sensor Networks (TOSN), Vol. 4 , No. 3 page(s): 1-37, May 2008.

[6] M. Richardson and R. Agrawal and P. Domingos *Trust Management for the Semantic Web*, Proceedings of the Second International Semantic Web Conference, ISWC-2003, LNCS 2870, Springer, page(s): 351-368, 2003.

[7] R. Guha and R. Kumar and P. Raghavan and A. Tomkins *Propagation of Trust and Distrust*, Proceedings International World Wide Web Conference (WWW2004), page(s): 403-412, 2004.

[8] H. Luo and J. Tao and Y. Sun *Entropy-Based Trust Management for Data Collection in Wireless Sensor Networks*, Proceedings of WiCom '09. 5th International Conference on Wireless Communications, Networking and Mobile Computing, page(s): 1-4, 2009.

[9] D. Huang and S. Bracher *Towards evidence-based trust brokering*, Proceedings of the SECOVAL Workshop, held in conjunction with the 1st IEEE/CREATE-NET SecureComm, 2005.

[10] L. Javier and R. Rodrigo and A. Isaac and F. Carmen *Trust management systems for wireless sensor networks: Best practices*, Comput. Commun, Volume 33, Issue 9, page(s):1086-1093, June 2010.

[11] N. F. Noy and D. L. mcguinness *Ontology development 101: A guide to creating your first ontology*, Stanford Medical Informatics Technical Report, March, 2001.

[12] V. D. S. Almendra and D. Schwabe *Trust Policies for Semantic Web Repositories*, Proceedings of 2nd International Semantic Web Policy Workshop (SWPW'06), at the 5th International Semantic Web Conference (ISWC-2006), page(s): 5-9, 2006.

[13] Web Ontology Language(OWL) specifications http://www.w3.org/TR/owl-features

[14] Wikipedia page for Semant Web http://en.wikipedia.org/wiki/Semantic_Web

[15] Resource Description Framework specifications http://www.w3.org/RDF/

[16] Turtle - Terse RDF Triple Language http://www.w3.org/TeamSubmission/turtle/

[17] SPARQL specifications http://www.w3.org/TR/rdf-sparql-query/

[18] Epinions website http://www.epinions.com/

[19] twitter website http://www.twitter.com/

[20] Our Trust Ontology http://knoesis1.cs.wright.edu/∼w024pxa/trustontology.owl