Network Composition for Situation Assessment: A “Trusted Meeting” Case Study

Shan Lu
Department of Electrical and Computer Engineering
Northeastern University
Boston, Massachusetts 02115
shanlu715@gmail.com

Alexey Tazin
Department of Electrical and Computer Engineering
Northeastern University
Boston, Massachusetts 02115
a_tazin@yahoo.com

Mieczyslaw M. Kokar
Department of Electrical and Computer Engineering
Northeastern University
Boston, Massachusetts 02115
kokar@coe.neu.edu

***Abstract*—Nowadays situation assessment in various scenarios depends on communication networks, information stored in various data stores (information networks), and knowledge of the roles and relations that particular people are involved in (social networks). In order to come up with a high-fidelity situation assessment, information from these three networks must be integrated consistently into a coherent situation model. This is a difficult task for humans to perform due to various reasons, e.g., interruptions, stress (emergency, war) and complexities of social relationships (trust). A tool (Network Composer) that would support people in assessing situations is desirable, but non-existent. In this paper we analyze the behavior of a Network Composer on a scenario in which two types of network are involved. The scenario deals with organizing a “trusted meeting” (treated as a situation) over a communication network, following some rules of trust. Information about the status of the meeting and the various people is stored in data stores associated with communication nodes. In this scenario, Network Composer would support a human in answering the question whether the preconditions for such a meeting have been satisfied, i.e., whether it is possible that the meeting will happen. To assess the concept of Network Composer we represent information flowing though these networks in the Web Ontology Language (OWL) and rules, and emulate the inferential behavior of the tool using a generic inference engine. The tool is analyzed using various scenarios of information flow in the different types of network.**

I. Introduction

The complexity of the situation assessment process is mainly due to the fact that assessing a situation involves not only detection of objects in the environment but also relations among objects, referred to as comprehension in[1]. Since relations among objects are not directly measured by sensors, situation awareness requires drawing conclusions from large amounts and varieties of information. Relevant and complete information content is critical to the quality of situation assessment [2].

In today’s networked world, situation assessment highly depends on the quality and reliability of communication networks. It also depends on information stored in various data stores, as well as on the knowledge of the roles of and relations among the particular human actors. In other words, situation assessment depends on the availability of information of three types: information stored in various data stores or collected by various sources (sensors), knowledge of how to access and exchange information among the communication nodes, and knowledge of the the roles that particular people play in the generation and consumption of information.

We view this complicated world as three types of network - information, communication and social - all interrelated through various types of dependencies. In general, at an abstract level, a network can be viewed as a graph, i.e., a collection of nodes and links between or among the nodes. A communication network is a collection of communication nodes and links that connect some of the nodes. An information network is a collection of information stores, where the links are various logical dependencies among the stores. For instance, a database can be viewed as an information node, while the links can be relations between the databases. E.g., the link may represent the fact that a given column in one database is the “same as” a column in another database. Nodes in social networks are people and the links among such nodes are various social relations, like “knows”, “likes”, “hates”, or “trusts”.

In order to come up with a high-fidelity situation assessment solution, information from these three networks must be integrated consistently (fused) into a coherent situation model. For instance, in order to determine whether a person is in danger of being attacked while in a specific area, it would be very useful to know not only who else is in the area (object detection and recognition), but also whether the people in the area are somehow related, e.g., being from the same paramilitary organization, the same street gang, or anything else. This kind of information could be delivered to the person via electroinic communication channels, provided there is connectivity in the area. Moreover, the connectivity would allow for access to databases that could possibly contain information about the recognized people in the area.

The main motivation for our work lies in the scenarios in which information about particular objects of interest is contained in networked systems. Moreover, as is usually the case with such systems, the information is temporarily out of sync and inconsistent, and thus needs re-synchronization. Synchronizing information contained in networked systems, and making it consistent across the networks, is a very difficult task for humans to perform. Such a task is even more difficult when the connectivity is intermittent. A tool that would perform this task automatically would be helpful. Since in this scenario the information of interest is about three different types of network - communication, information and social - the synchronization operation is called *network composition*. The tool that supports network composition is called *Network Composer*.

There are many reasons why the composition operation is complicated. First of all, the actual networks are becoming more and more complex and sophisticated, since the underlying networks are from diverse domains, holding vast amounts of information in different ways [3]. Communication networks facilitate communication and sharing of resources and information among interconnected computers and devices. Information networks provide multiple (possibly interrelated) stores of information about objects, events and so on. Social networks discover implicit, previously unknown, and potentially useful knowledge or information by studying social relationships. However, all three kinds of information may be interrelated. Thus composition needs to involve "second degree effects" - the integration of information about objects, but also about networks of objects.

Second, the network instability and dis-connectivity is rather typical in many specific scenarios, such as in the military or emergency response operations. So the fact that there may be interruptions in the connectivity of the communication network further increases the complexity of information integration.

Finally, the relational ties in actual social networks are highly diverse [4], such as the feeling a person has for another (friendship, trust), communication (know, invite), or behavioral interactions (cooperate, compete). But most of the existing research treats all types relationships in the same way, not giving sufficient attention to the nuances of differing strengths and types of information in social networks. For instance, trust is an important social relationship for decisions making. In fact, this type of relationship has a very high level of complexity and constitutes a research subject on its own. And yet in most of the research on social networks such relationships are abstracted by a much simpler relationship of "connected".

Therefore, a tool that would support the humans in assessing situations relevant to particular human-defined objectives is highly desirable. Our primary research goal is thus the development of such a tool. In this paper we present the results of our initial investigation into this problem. We discuss some of the functions of a network composition tool and analyze the requirements for such a tool. In particular, we develop a scenario of situation assessment in which two types of network are involved. The scenario deals with an organization of a "trusted meeting", i.e., a meeting of a number of people who are invited following some rules of trust. Meeting is treated as situation. The task of organizing such a meeting uses a communication network. Information about the status of the meeting and about the various people is stored in data stores associated with communication nodes. The main point then is to analyze a possible functionality of a computer tool that would support an actor in answering the question whether all the preconditions for such a meeting have been satisfied, i.e., whether it is possible that the meeting will happen. We view Network Composer as consisting of two parts: the Information Composer that is used to collect and compose all the information in the composite network, and the Inference Engine that takes the information from Information Composer as input to infer answers to various queries.

Generally, there are two ways for implementing an automatic composition operation: procedural (imperative) and logical (declarative). One can develop a software system in which some procedures for the composition operation are hard coded. In this way, only those queries for which procedures have been explicitly coded by the system developer can be answered. In this paper, we use the logical approach in which a generic inference engine is capable of answering any query that is expressible in the query language. To analyze the behavior of Network Composer, we represent the information flowing though the networks in the Web Ontology Language (OWL) and rules, and emulate the inferential behavior of the Network Composer tool. We start with developing ontologies to represent the composite network using Prot*égé* [5]. Then, we use the standard rule-based inference engine BaseVISor to support inference based query answering. BaseVISor permits the representation of complex logical conditions and supports a reasonable set of OWL constructs while remaining sound and tractable [6]. Network Composer's behavior is analyzed on the various scenarios of information flow in the networks.

The rest of this paper is organized as follows: Section II reviews some related work on information integration. Section III gives a short introduction of ontology and OWL. Section IV overviews the main functions of Network Composer. Section V describes the use of Prot*égé* 4 as the editor to build the OWL composite network ontology. Section VI considers four types of composition cases for the trusted meeting scenario to test the Network Composer, and analyzes the requirements for this tool based on the results. Finally, conclusions and future work are discussed in Section VII.

II. Related Work

Information integration is the merging of information from disparate sources with differing conceptual, contextual and typographical representations [7]. It provides an integrated and coherent view of data stored in multiple and inhomogeneous information sources. Information integration is closely related to data fusion, where the input is a collection of data from multiple sensors and related information from associated databases, and the goal is to improve the accuracy of decisions and to support specific inferences [8]. Data integration can be either virtual or materialized [9]. In virtual integration, the integration system acts as an interface between the user and the sources, where the sources may be multidatabases and distributed databases. In the materialized integration, the system maintains a replicated view of the data at the sources.

There are two basic approaches to the information integration problem: procedural and declarative. The first approach integrates data in by designing suitable software modules that access the data sources [10]. The second approach achieves the integration by modeling the data at the sources by means of a suitable language to construct a unified representation [9]. The latter approach allows for maintaining a consistent global view of the information sources. In declarative case, there are two critical factors for the design and maintenance of information integration: the conceptual modelling of the domain, and the possibility of reasoning over the conceptual representation.

Information integration on the Web involves a number of architectural building blocks that are the focus of work of the W3C and the Semantic Web community, including mechanisms for information encoding and manipulation, and ontology construction and reasoning [11]. H. Wache [12] analyzes existing solutions with special focus on the use of ontologies in these approaches. The overview includes SIMS, TSIMMIS, OBSERVER, CARNOT, Infosleuth, KRAFT, PICSEL, DWQ, Ontobroker, SHOE and other information integration approaches.

Enterprise Information Integration (EII), is a process of information integration, which provides tools for viewing all the data within an organization, and a single set of structures and naming conventions to represent this data [13]. A variety of information integration tools already exist. P. A. Bernstein and L. M. Haas [14] give a review of several such tools and their core technologies, for instance, Data Warehouse Loading, Extract-Transform-Load (ETL), Virtual Data Integration, Message Mapping, Object-to-Relational Mappers, Document Management, Portal Management. However, none of the tools/technologies above address composition of information from all three types of network. Therefore, we will discuss the main functions of this kind of system and analyze the requirements for such a tool in the rest of this paper.

III. Ontology and OWL

Ontology, a term from philosophy, is a science of representing a common understanding about concepts and relationships of a specific domain. In AI, the definition of ontology is [15]: "Definitions that associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms." Typically, it specifies the classes of objects that exist, the relationships amongst those classes, the possible relationships amongst instances of the classes, and constraints over those instances.

The Web Ontology Language (OWL) is a knowledge representation language for specifying ontologies [16]. OWL uses Uniform Resource Identifiers (URIs) for naming. OWL builds on RDF and RDF Schema and adds more vocabulary for describing properties and classes - among others, relations between classes (e.g., disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g., symmetry), and enumerated classes. It also supports advanced capabilities such as logical inference.

Prot*égé* is a free, open source ontology editor and knowledge-base framework. It allows us to check the consistency of the ontology and automatically compute the ontology class hierarchy using embedded reasoners.

IV. Network Composer System

A. Basic Definition of Composite Networks

In order to analyze the requirements and main functions of Network Composer, we consider a complex system called *composite network*, which consists of three underlying networks from different domains: communication network, information network and social network. Each network has its own nodes and edges, all associated with computer platforms. Information flows among these nodes from different domains through the edges (links) of the communication network. Below we provide a first level of formality for the three types of network.

1) Communication Networks: From the communication perspective, a network is a collection of computers and devices (called hosts or terminals) interconnected by a single technology (e.g., communications channels). The network uses a set of rules called protocols to achieve communication. The most important and widely used communication network is the Internet.

Definition 1. *A Communication Network (CNet) consists of a set of communication nodes (CN) and edges (links) - communication links:*

$$link : CN \times CN \rightarrow Boolean \quad (1)$$

$$CNet = < CN, link > \quad (2)$$

The Boolean values of this function (this is a representation of the relation between nodes using a function) define which of the nodes are connected.

2) Information Networks: From the informational perspective, a large number of individual components using networking technologies (e.g., wireless communication) for distributing and sharing information with a specific set of components, forms an information network. The network of citations between academic papers is the classic example of an information network. A collection of inter-related databases is another example.

Definition 2. *An information network (INet) consists of a set of information nodes (IN) and a set of relational links constituting a relation (*map*).*

$$map : IN \times IN \rightarrow Boolean \quad (3)$$

$$INet = < IN, map > \quad (4)$$

The information nodes, e.g., databases, are distributed storage units. They provide different access rights to different users following their access policies. The information nodes allow authorized operations to be performed on their records, such as query, add, remove, modify. The map links between information nodes represent the knowledge sharing conditions.

3) Social Networks: From the social perspective, a network is a structure made of social entities (e.g., individuals, corporations, collective social units, organizations), which are linked by some specific types of interdependency (e.g., kinship, friendship, common interest, beliefs). Facebook is one of the most famous applications of social networks.

Definition 3. *A Social Network (SNet) is a collection of social nodes (SN) and a collection of relations (edges) (*relationships*).*

$$relationship : SN \times SN \rightarrow Boolean \quad (5)$$

$$SNet =< SN, \{relationship\} > \quad (6)$$

Note that while for a communication network we are mainly interested in one type of edge (the communication links), for social networks the relationships can be of many different types, so consequently in the definition we have a set of relations, rather than just one relation.

Since in our approach we deal only with communication network based interaction among social entities, we assume that each social node is associated with at least one communication node, which includes a computer as a part of it. Social nodes are then represented by *user profiles* stored on a computer. While qualitatively this kind of information could be considered as information nodes, we model them separately due to the special role they play in the net centric world.

Social nodes represent intelligent entities (humans) and thus it would be over optimistic to assume that the profiles capture the knowledge and inference capabilities of the particular users. Such nodes need to communicate with other social nodes to accomplish their tasks and achieve their goals. The intelligence of social nodes can (at least partially) be represented by a knowledge representation scheme, while their inference capabilities could be, again only partially, by the inference engines which automatically collect information, do the reasoning and derive decisions. The social nodes are linked by social relationships among them. They can send and receive messages, and establish new relationships as a result of such communication acts. Interactions among social nodes should follow some social policies, which could be approximated by policy rules interpreted by automatic inference engines.

4) Situations: A situation can be defined as an object (instance) of class *Situation* along with a set of other objects that are relevant to the situation and relations of different types (relational types) among the objects [17]. The objects in a situation are related to the situation via the property *relevantObject*, while the relevant relations are related via the *relevantRelation* property. The properties *relevantObject* and *relevantRelation* come from the Situation Theory Ontology (STO) described in [17]. In this paper, for the sake of simplicity, we do not show all of the details of how the networks are represented in STO, but instead, we directly show domain specific properties, like *hasNetwork*, *hasNode*, and so on.

In the scenario discussed in this paper, both the objects and the relations change with both time and location. Situations are often of a high complexity and dynamic because they consist of multiple different components of different networks that interact with each other, and their activities evolve over time. This all serves as justification for a tool for network composition.

5) Composite Networks: The formalization of the three types of networks provided above is just a fist step to a more full definition of *Composite Networks*. The rest of the formalization of this concept has been implemented in the OWL language as part of an ontology for network representation and composition. The ontology is partially shown later in this paper.

B. Trusted Meeting Scenario Description

We use a simple scenario called "trusted meeting" as a step towards the development of a specification of network composition. In this scenario, a group of individuals execute the mission of organizing a "trusted meeting". The meeting is called "trusted" because the individuals will attend the meeting if and only if they are invited by an individual whom they trust. As a first step, the meeting planner sends invitations to all the individuals that are necessary for this meeting. Later on, the meeting planner queries the composite (in our case the Network Composer system) whether this meeting will happen or not. The details of this scenario are as follows.

There are three persons in the social network: $Alice$, Bob and $Carl$. Additionally, there are three communication nodes in the communication network: A, B and C. $Alice$ is the meeting planner. She invites two members she needs for this meeting (Bob and $Carl$) by sending invitations to both using her computer that is connected to the communication node A. The request for the meeting sent by $Alice$ includes a specification who should attend. It is assumed that the requests reach all of the invitees. Later $Alice$ sends a query of whether the meeting will happen. Each node is aware only of its own knowledge. Network connectivity is necessary to infer whether meeting happens. In some of our scenarios, the connectivity is not assumed. If network is disconnected, chains of relationships may be broken. When network is reconnected, knowledge representations of disconnected components must be recomposed.

While the communication links change over the run of some of the scenarios defined below, the social and informational relationships are assumed to be constant. These relations for the scenarios discussed in this paper are specified below. Some of them are expressed in natural language and some use a semi-formal notation. All of them have been implemented as OWL constraints or rules and processed by the inference engine.

- Need: the trusted meeting requires attendance of all the members invited by the meeting planner.
- Trust: trust is transitive.

$$trust(x, z) \wedge trust(z, y) \Rightarrow trust(x, y) \quad (7)$$

- Accept: members will accept the invitation to a trusted meeting only if invited by a member they trust.

$$invite(y, x, M) \wedge trust(x, y) \Rightarrow accept(x, M) \quad (8)$$

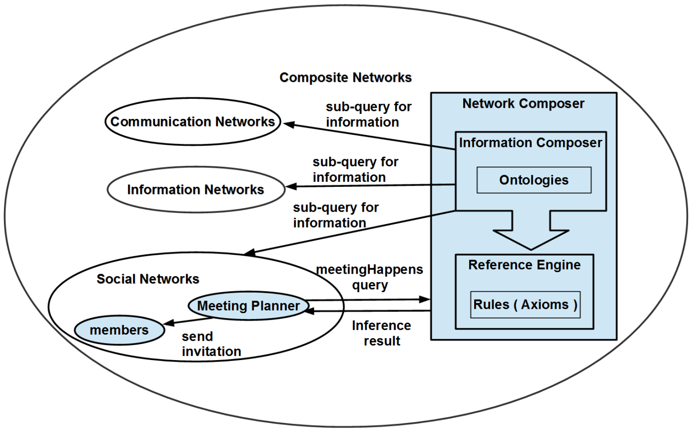

Fig. 1. The overview of the network composer system

- MeetingHappens: only if all of the invitees receive/accept the invitation.

$$accept(y1, M) \wedge accept(y2, M) \ldots \Rightarrow meetingHappens(M) \tag{9}$$

C. Network Composer System Overview

In this paper, we propose a Network Composer system that could support (among others) meeting planners to organize a trusted meeting by providing the capability of replying to queries sent to this system. Figure 1 illustrates an overview of the interactions with the Network Composer system.

Network Composer consists of two parts: Information Composer and Inference Engine. The Information Composer part collects all the related information in the composite network and integrates this information consistently into a coherent situation model. In Section V, we discuss how this information is expressed using an OWL ontology. The Inference Engine part takes the OWL ontologies and rules as input and does the specific reasoning for the queries. We use the standard rule-based inference engine BaseVISor for the reasoning.

The sequence of operations of Network Composer is as shown below.

1) Break the received query into sub-queries and send the sub-queries to the particular nodes.
2) Collect all the replies to the sub-queries (in this meeting scenario, sub-queries are about trust relationships).
3) Compose the information into a coherent situation model.
4) Perform the inference automatically based on the coherent situation model and rules (axioms).
5) Provide the inference results to the querier (in this case to the meeting planner) whether the meeting happens.

As we mentioned before, disconnections of the communication network may negatively impact the correctness of the assessment decision. So Information Composer should integrate information from these three networks consistently and synchronously. The crucial issue in this part is how to perform this kind of composition. OWL supports the sharing and reuse

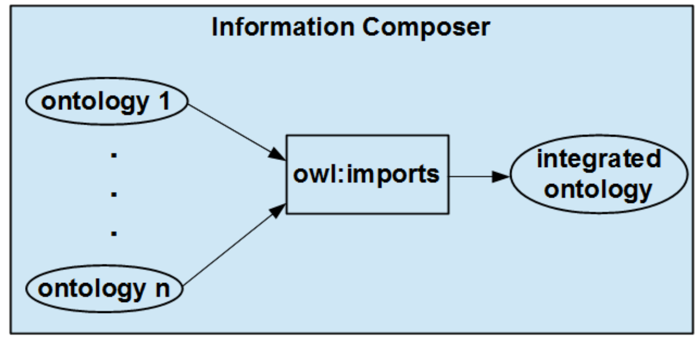

Fig. 2. Information Composer

of ontologies by making it possible for one ontology to import another ontology. “owl:imports” lists other ontologies whose content is assumed to be part of the current document. When an ontology imports another ontology, all of the class, property and individual definitions that are in the imported ontology are available for use in the importing ontology. In this paper, we will use “owl:imports” to achieve the information composition described in Section VI. The Information Composer we used in this study is shown in Figure 2. Clearly, this approach is too simplistic for any practical composition operations.

V. Composite Network Ontology Development

An OWL ontology consists of Classes, Properties, and Individuals.

A. Classes

OWL classes are a concrete representation of concepts in the specific domain. Classes can be interpreted as sets that contain instances of concepts. We can organize concepts into a taxonomy. It means classes can be structured into a superclass-subclass hierarchy, in which the subclasses specialize their superclasses. Figure 3 illustrates the class taxonomy of the composite network ontology that we have developed.

B. Individuals

Individuals represent specific objects in the domain in which we are interested. Individuals are instances of classes. In our composite network ontology, there are four types of individuals based on the statement of the trusted meeting scenario. The red boxes in Figure 4 show the individuals in the composite network ontology.

C. Properties

Properties are binary relationships between individuals of two classes. There are two main types of properties: object properties and datatype properties. In our composite network ontology, we only consider the object properties that are relationships between two individuals (binary properties). More general (n-ary) properties would require some special means to represent in OWL, e.g., properties could be represented as classes. Properties link individuals from the domain to individuals from the range. Table I shows the properties in the composite network ontology.

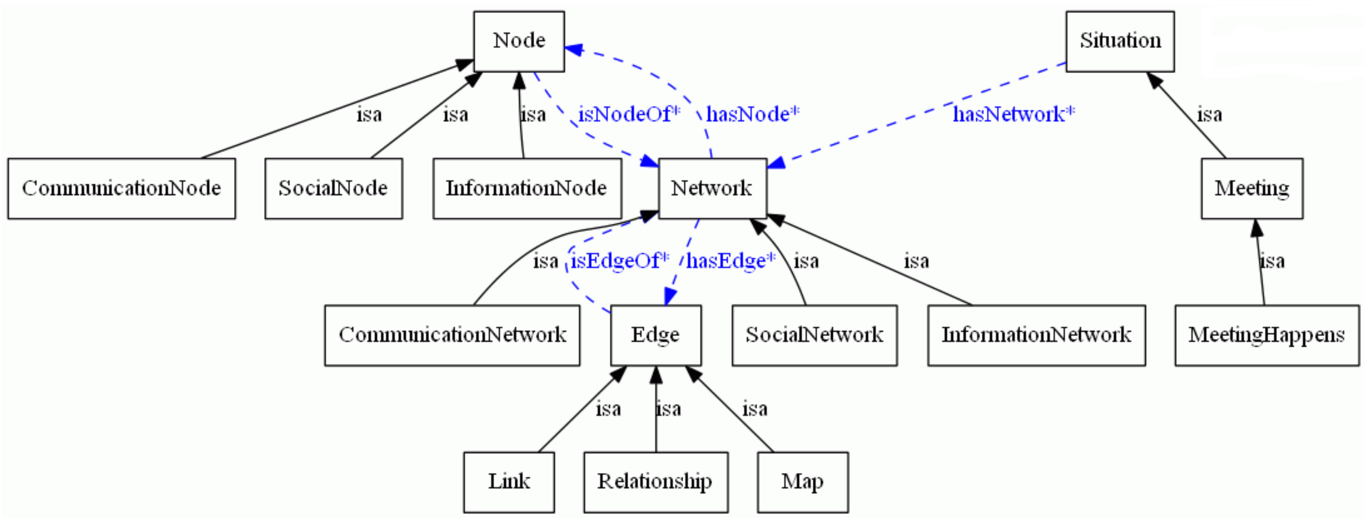

Fig. 3. The class taxonomy of the composite network ontology

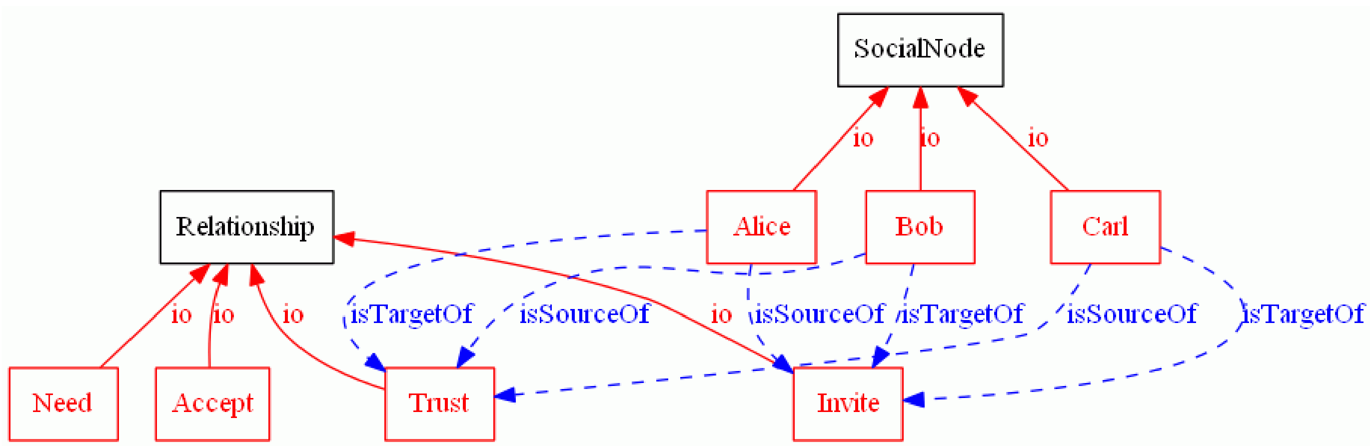

Fig. 4. Some of the individuals in the trusted meeting scenario

TABLE I
THE OBJECT PROPERTIES OF THE COMPOSITE NETWORK ONTOLOGY

Property	Domain	Range
hasEdge	*Network*	*Edge*
hasNode	*Network*	*Node*
hasNetwork	*Situation*	*Network*
hasSource	*Edge*	*Node,Situation*
hasTarget	*Edge*	*Node,Situation*
isEdgeOf	*Edge*	*Network*
isNodeOf	*Node*	*Network*
isSourceOf	*Node,Situation*	*Edge*
isTargetOf	*Node,Situation*	*Edge*

VI. Composition and Inference

An ontology can be classified into two different groups of statements: "T box" and "A box". T box statements, which describe a conceptualization, a set of concepts and properties for these concepts, are the basic terms in a specific domain. A box are T box-compliant statements about instances belonging to those concepts. Usually, T box includes classes and properties while A box includes individuals and property links between the individuals.

In our work so far, we simulated the various scenarios by using the basic ontology of composite networks - the T box - and modifying the A boxes. In the discussion in this paper we consider four types of composition cases in this trusted meeting example, and then compare the ground truth and the inference results of our composition system. Consequently, we discuss four different A Boxes, one for each of the types of scenarios. These four types of composition cases are based on the same initial facts:

- $Alice$ trusts all the other social nodes
- Bob trusts $Alice$, $Carl$ trusts Bob
- $Alice$ invites Bob and $Carl$ to the $TrustedMeeting$

A. Case 1

The sequence of interactions for this scenario is shown in Figure 5. The assumption here is that the communication network is perfectly connected all the time.

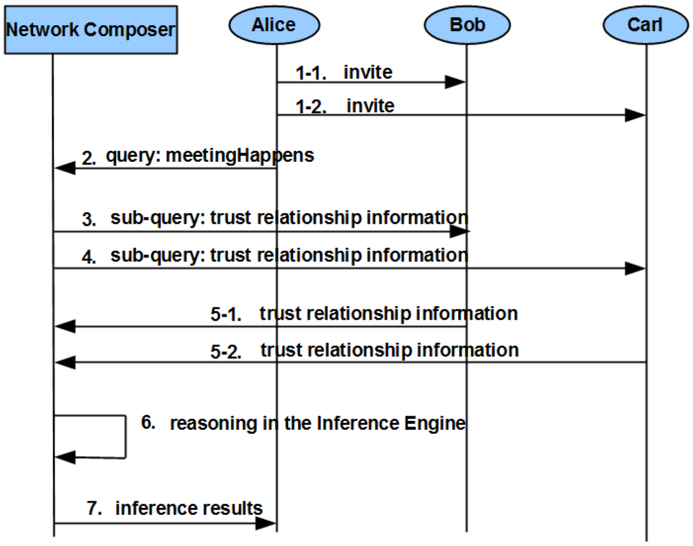

Fig. 5. Case 1

Based on the rules mentioned in Section IV-B, we can easily get the ground truth by human reasoning:

- Bob trusts $Alice$, so Bob accepts the invitation.
- $Carl$ trusts Bob, so $Carl$ trusts $Alice$. Bob also accepts the invitation.
- $TrustedMeeting$ will happen.

To answer the query automatically, we use Network Composer, which uses BaseVISor for reasoning. The ground truth and the inference results of the inference for all the four types of scenarios are shown in Table II. As can be seen from this table, Network Composer infers that the result of the query for this case should be Yes.

B. Case 2

The interaction sequence for this scenario is a bit different than the one shown in Figure 5. The assumptions for this scenario are that:

- The communication network is perfectly connected during the invite time.
- The communication network is suddenly disconnected during the query time.

In this case, the invitations are sent when the communication network is perfectly connected. So communication events 1-1 and 1-2 shown in Figure 5 will take place. The ground truth derived by human reasoning is that the $TrustedMeeting$ will happen, similarly like in Case 1. However, the communication network is disconnected during the query period, so Network Composer is unable to collect sufficient information from the the networks. Consequently, communication events 3, 4, 5-1 and 5-2 will not be accomplished. As a result, the inference result for the $meetingHappens$ query is *Unknown*. The $Unknown$ result is due to the Open World Assumption in OWL; it means that if a fact cannot be proved neither true nor false, it is unknown.

C. Case 3

The assumptions for this scenario are that:

- The communication network is disconnected during the invite time.
- The communication network is re-connected during the query time.

Without communication connection, the invitations can't be transmitted or forwarded. Thus messages 1-1 and 1-2 will not get through. But since the communication is re-established very quickly, Network Composer, after receiving the query, will be able to contact *Bob* and *Carl* about their trust relationships. For this case, the ground truth by human reasoning is that the $TrustedMeeting$ will not happen. However, since in our composer system the whole information is broken into local ontologies (ABoxes) for each of the social network nodes, in query reply the import operation is used for the integration of information about the reconnected network. So the inference result for the query$meetingHappens$ is *Yes*, thus wrongly concluding that the $TrustedMeeting$ will happen. Clearly, the inference result is incorrect. This kind of error is called *false positive*.

D. Case 4

The assumption for this scenario is that the communication network is disconnected both during the invite time and the query time. Thus messages 1-1, 1-2, 3, 4, 4-1 and 5-2 will not be delivered. In this case, the ground truth is clearly *No*, and the inference result is *Unknown*.

E. Requirements Analysis and Discussion

In this paper we discussed four types of scenario developed for the cause of analyzing requirements of a network composition system.

- Described scenarios in a formal language (OWL).
- Expressed queries in SPARQL (a query language for the Semantic Web; not discussed in this paper).
- Used an inference engine (BaseVISor) to derive answers to queries.

Comparing the four composition cases discussed above, we can see that whether the mission of organizing a trusted meeting can be accomplished with the support of a network composition tool depends on the way the composition of information from the three different types of network is performed. As we can see from Table II, the inference results of our Network Composer resulted in false positives and false negatives. The correct and efficient composition should minimize both the false positives and the false negatives. In fact, there are many potential possible conditions that may result in false positives or false negatives if only a too simple operation (like $owl : imports$) is used for network composition.

After network re-connection, there may be multiple different individual terms in the integrated ontology that represent the same individual that existed in the original ontology before disconnection. In other words, there may be overlaps among

TABLE II
FOUR TYPES OF COMPOSITION CASES

	Query	Network connectivity		Ground truth	Inference result
		during *invite*	during *query*		
Case 1	*meetingHappens*	connected	connected	yes	yes
Case 2	*meetingHappens*	connected	disconnected	yes	unknown
Case 3	*meetingHappens*	disconnected	connected	no	yes
Case 4	*meetingHappens*	disconnected	disconnected	no	unknown

those separated ontologies. So, the composer system should unify the terms representing the items that are "the same".

Moreover, some information may be accumulated in component networks during the disconnect. At the same time, some information may be removed. For example, some individual terms are deleted from the component networks and some terms are added into the component networks. When the network is reconnected, the question is how to update and synchronize this accumulated information from component networks into the integrated ontology is a challenge. So the composer system should use some technologies or theories to satisfy such an objective.

Last but not the least, the relationships among the individuals also may change during the operation of the network. The composer system should try to ensure that all the relationships among the terms from the component ontologies are preserved in the resultant composed ontology. Clearly, using just the $owl : imports$ operation for this purpose is not sufficient. Towards this aim, some extensions to the capabilities of OWL language are needed. For our composer system, there will be many more requirements to be specified. Currently we are investigating various theories and technologies to be taken into account in the development of both the requirements and in the ultimate implementation. In particular, we looking into the use of mereology and category theory (colimit) for the composition operation.

VII. CONCLUSION

Situation assessment depends on the way the network composition is performed. Disconnection of communication network may negatively impact the correctness of the assessment decision. In this paper, we discussed a first prototype of Network Composer that we developed and analyzed the requirements for such a system on a scenario of a "trusted meeting" in which all two types of network are involved. The prototype makes use of ontologies expressed in OWL. We analyzed four composition cases for testing the Network Composer. So far we used only a basic $owl : imports$ operation for network composition. But clearly this operation is insufficient for real composition applications. Several requirements for minimizing false positives and false negatives were discussed. In the future, we plan to take more theories and technologies (e.g., category theory, mereology) into account to develop a network composition tool that satisfies these requirements.

REFERENCES

[1] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32–64, 1995.

[2] Y. Fischer, A. Bauer, and J. Beyerer, "A conceptual framework for automatic situation assessment," *2011 IEEE International MultiDisciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support CogSIMA 2011*, pp. 234–239, 2011.

[3] M. E. J. Newman, "The structure and function of complex networks," *SIAM REVIEW*, vol. 45, pp. 167–256, 2003.

[4] M. Szell, R. Lambiotte, and S. Thurner, "Multirelational organization of large-scale social networks in an online world," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, no. 31, pp. 13 636–13 641, 2010.

[5] "Protégé, stanford center for biomedical informatics research," 2011. [Online]. Available: http://protege.stanford.edu/

[6] C. J. Matheus, K. Baclawski, and M. M. Kokar, "Basevisor: A triples-based inference engine outfitted to process ruleml and r-entailment rules." in *RuleML'06*, 2006, pp. 67–74.

[7] G. Wiederhold., "Intelligent integration of information," *Journal of intelligent information systems*, vol. 6, no. 2-3, 1996.

[8] D. Hall and J. Llinas, "An introduction to multisensor data fusion," in *Proceedings of the IEEE*, vol. 85, no. 1, 1997, pp. 6–23.

[9] D. Calvanese, G. D. Giacomo, M. Lenzerini, D. Nardi, and R. Rosati, "Description logic framework for information integration," in *Proc. of the 6th Int. Conf. on the Principles of Knowledge Representation and Reasoning*, 1998, pp. 2–13.

[10] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom, "The tsimmis project: Integration of heterogeneous information sources," in *In Proc. of IPSI-94 Conference*, 1994.

[11] M. Doerr, J. Hunter, and C. Lagoze, "Towards a core ontology for information integration," *Journal of Digital Information*, vol. 4, no. 1, 2003.

[12] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner, "Ontology-based integration of information - a survey of existing approaches," in *Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing*, 2001, pp. 108–117.

[13] A. Y. Halevy, N. Ashish, D. Bitton, M. Carey, D. Draper, J. Pollock, A. Rosenthal, and V. Sikka, "Enterprise information integration: successes, challenges and controversies," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '05, 2005, pp. 778–787.

[14] P. A. Bernstein and L. M. Haas, "Information integration in the enterprise," *Communications of the ACM*, vol. 51, no. 9, pp. 72–79, September 2008.

[15] T. Grube, "What is an ontology?" [Online]. Available: http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

[16] W3C, "Web ontology language reference, ow." [Online]. Available: http://www.w3.org/2004/OWL/

[17] M. M. Kokar, C. J. Matheus, and K. Baclawski, "Ontology-based situation awareness," *Inf. Fusion*, vol. 10, no. 1, pp. 83–98, January 2009.