Analyzing Trust Requirements in Socio-Technical Systems: A Belief-Based Approach

Mohamad Gharib(✉) and Paolo Giorgini

University of Trento - DISI, Povo, 38123 Trento, Italy
{gharib,paolo.giorgini}@disi.unitn.it

Abstract. The Requirements Engineering (RE) community recognizes the importance of trust proposing several approaches to model and analyze trust requirements. However, such approaches mainly focus on trust as social relations without relating them to the requirements of the system's components. We propose a belief-based trust approach based on an extended version of Secure Tropos, where social relations are modeled and analyzed along with beliefs concerning capabilities and motivations of system's components. An example concerning US stock market crash (the Flash Crash) is used to illustrate our approach.

Keywords: Trust analysis · Beliefs · Requirements engineering

1 Introduction

Trust is very important in human societies [19], so it is in Socio-Technical Systems (STS) [7], where organizations and humans are an integral part of the system. Vulnerabilities of a STS are not only originated by technical issues, but they can be directly related to the social interactions of its components (both social and technical actors). For example, the Flash Crash was not caused by a mere technical failure, but it was due to undetected vulnerabilities in the interactions of the STS [21]. In [9], we showed how STS vulnerabilities can be avoided when trust requirements are considered properly during the system design.

The RE community recognizes the importance of trust, proposing several approaches for modeling trust requirements (e.g., Secure Tropos [17], SI* [24], etc.). However, none of them proposes to analyze trust starting from the perspective of socio-technical components of the system. More specifically, local objectives, competencies, motivations, etc. should be considered as main drivers for the definition of the trust requirements. One main contribution of this paper is enriching trust requirements analysis during the early phase of the system development by acquiring information about each socio-technical component (actor) and using such information to derive trust requirements.

To this end, we propose a belief-based trust approach built on an extended version of Secure Tropos [10], where a set of beliefs related to actors' competencies (can do) and motivations (will do) are used to clearly identify "why"

J. Ralyté et al. (Eds.): PoEM 2015, LNBIP 235, pp. 254–270, 2015.
DOI: 10.1007/978-3-319-25897-3_17

an actor may trust/distrust another one. The proposed framework is fully supported by a CASE Tool, and it offers a methodological process to assist analysts during the different phases of the system analysis, along with several reasoning techniques that support the verification of the correctness and consistency of the requirements model. The rest of the paper is organized as follows; Sect. 2 describes the research baseline, and we discuss an example concerning the stock market system in Sect. 3. We detail our approach in Sect. 4, and we implement and evaluate it in Sect. 5. Related work is presented in Sect. 6, and we conclude the paper and discuss the future work in Sect. 7.

2 Research Baseline

Our research baseline is founded on three main research areas:

(1) Goal Oriented Requirement Engineering (GORE): GORE is used to express the system requirements in terms of the actors of the system along with their objectives, entitlements, and capabilities. Among the existing GORE approaches (e.g., SI* [17], Secure Tropos [24], etc.), we adopt an extended version of Secure Tropos [10] as a baseline for our approach. In which, an actor covers two concepts a role and an agent, where roles can be specialized from one another (*is_a*), and an agent can *play* a role or more. Goals can be refined through And/Or decomposition of a root goal into sub-goals, where a goal is achieved if all of its (And-decomposition) or any of its (Or-decomposition) sub-goals are/is achieved. Goals may produce and read information. Finally, it provides concepts for modeling information provision and goals delegation among actors, and it adopts the notion of trust to capture the expectation of actors in one another.

(2) Belief-Desire-Intention (BDI): The BDI paradigm [18] enables for analyzing actors of the system in terms of their beliefs, desires, and intentions. In BDI an agent has a set of beliefs representing its knowledge about the world, desires (goals) it aims to achieve, and intentions (plans) that play an important role in determining their behavior to achieve their desires. The main reason for considering BDI paradigm is our need to capture beliefs about the actors' capabilities and motivations.

(3) Trust Management: There is general consensus that trust is complex and multidimensional concept [6], which motivates several researchers to answer the challenging question "how can trust be constructed (built)?". However, Esfandiari and Chandrasekharan [8] argued that constructing trust can be broadly classified under two main types: cognitive and mathematical, where in the first trust is made up of underlying beliefs (e.g., [1,4]), while in the last trust can be constructed based on some trust metrics (e.g., [8,20,22]).

In particular, our approach adopts concepts from extended Secure Tropos [10] to model the system requirements in terms of the actors of the system along with their objectives, entitlements, and capabilities. However, this framework does not propose techniques for modeling nor analyzing trust requirements

based on actors' capabilities and motivations. Thus, we rely on the BDI paradigm to capture actors' capabilities and motivations through their related beliefs. While, we depend on trust management to understand how different scholars have analyzed/constructed trust.

3 A Stock Market System Example

Our example concerns the Flash Crash, in which the Dow Jones Industrial Average (DJIA) dropped about 1000 points (about 9 % of its value). Several stakeholders of the stock market system can be identified based on [15], including: *stock investors* are individuals or companies, who have a main goal "Make profit from trading securities".

Stock traders are individuals or companies involved in trading securities in *stock markets* either for their own sake or on behalf of their *investors* with a main goal "Make profit by trading securities". *Stock traders* can be broadly classified under: *Market makers*: traders who facilitate trading on particular security and they have the capability of trading large amount of securities; *High-Frequency Traders (HFTs)*: traders who have the capability to trade large amount of securities with very high frequency; and *small traders*: traders who trade small amount of securities with low trading frequency.

Stock markets are places where *traders* gather and trade securities (e.g., NASDAQ, CME, NYSE, etc.), and they have a main goal "Make profit by facilitate security trading". Furthermore, *accounting firms* are specialized for providing companies with reliable information about their economic activities and the status of their assets. Moreover, *auditing firms* are specialized for monitoring the quality of the financial statements produced by companies. *Consulting firms* are specialized for providing professional advices for a fee about securities to *traders* and *investors*. Finally, *credit assessment ratings firms* are specialized for assessing of the credit worthiness of companies' securities.

Figure 1 shows a partial goal model of the stock market to clarify the modeling language main constructs, in which `Fast trading` is an agent that plays the `HFT` role, which is specialized from the `stock trader` role. A `stock trader` has a main goal `Make profit by trading security` that is And-decomposed into `Analyze the market for targeted securities` and `Make profit by producing the right orders`, which `reads Investor's orders` and `produces Trader's orders`. A `market` depends on `trader` for the provision of `Trader's orders`, and a trust relation concerning such provision exist. *Space Co* delegates `Manage company's auditing operations` to *Account and audit Co*, and a trust relation concerning such delegation exist.

Extended Secure Tropos [10] is able to capture the functional and trust requirements of the system in their organizational context. However, unlike functional requirements, which are supported with various kinds of analysis mechanisms that enable for deciding whether they can be achieved or not, trust analysis did not receive such an attention. For example, in order to ensure stable trading environment, a trust relation should hold between the *stock market* and *stock*

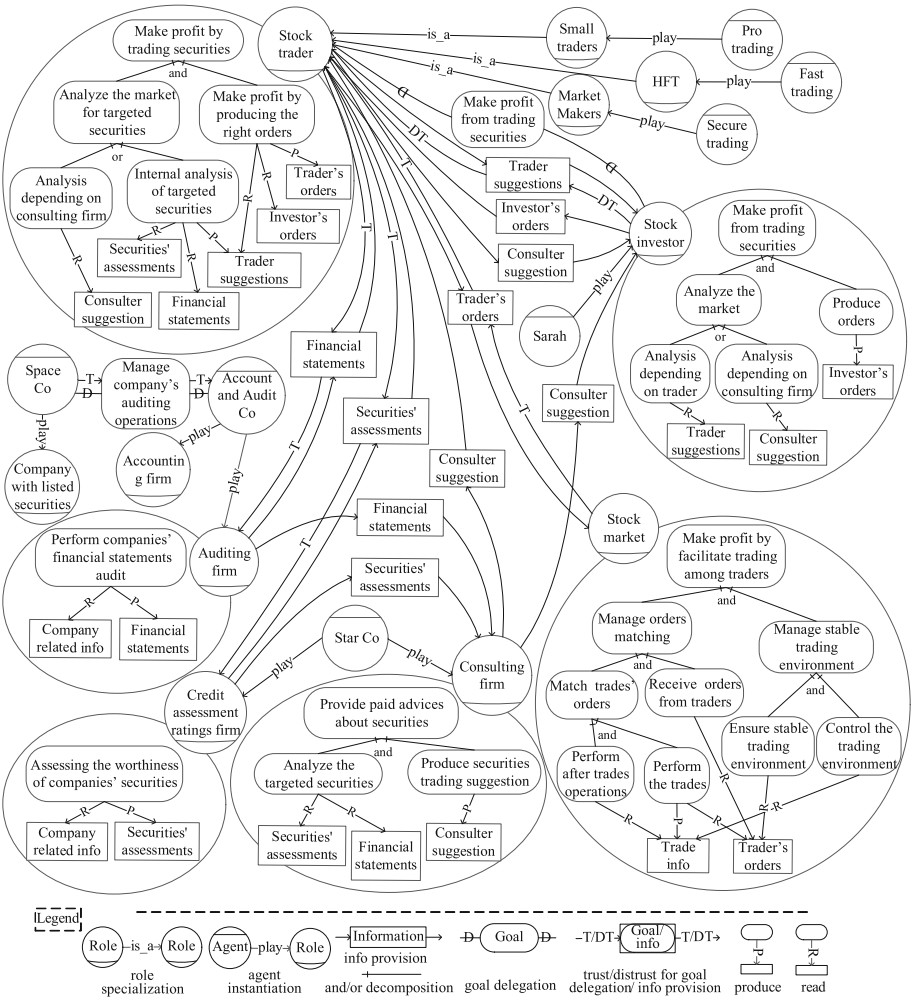

Fig. 1. A partial goal model concerning the U.S stock market structure

traders concerning the trading orders they provide. However, there is no existing technique that enables for analyzing trust relations based on the *traders'* capabilities and motivations, which form the bases of such trust, i.e., we cannot decide if such trust requirement is not conflicting with the *traders'* goals, intentions, etc.

4 A Belief-Based Approach for Analyzing Trust

Our approach proposes to model and analyze trust requirements among actors strictly from sets of beliefs concerning their capabilities and motivations. In

particular, an actor (trustor) may trust/distrust another one (trustee) based on beliefs related to trustee's competencies and motivations. In this section, we detail our belief-based trust approach. First, we discuss our modeling extensions that enable for analyzing trust based on actors' competencies and motivations related beliefs, and then we describe the automated reasoning support that can be used to verify the correctness and consistency of the trust requirements model. Finally, we outline the methodological process that underlies our approach.

4.1 Extended Modeling Language for Analyzing Trust Requirements

The need for trust arises when actors depend on one another either for goals to be achieved or information to be provided[1], since such dependencies might entails risk (threat) [6]. Following [4], to analyze trust we need to identify beliefs related to the trustee's competencies and motivations toward the trustum (e.g., goal achievement or information provision). However, relying only on the trustee's competencies and motivations toward the trustum might not be enough; indeed, other factors might influence the trustee's behavior. In particular, actors are intentional entities and they might have other objective(s) that its/their achievement might threaten/prevent the achievement of the trustum. We call such objectives as *intentional threat* [23] (*threat* for short), since they exists only within the actors' objectives.

In addition, we classify actor's competencies toward fulfilling[2] a trustum/threat under: (1) *Competence*, when there is belief(s) about its capability toward fulfilling the trustum/threat; and (2) *Incompetence*, when there is belief(s) about its incapability toward fulfilling the trustum/threat. While actor's motivations can be classified under: (1) *Positively motivated*, when there is belief(s) about its intentions toward fulfilling the trustum/threat; and (2) *Negatively motivated*, when there is belief(s) about its negative intentions toward fulfilling the trustum/threat. To this end, trust can be analyzed in the trustee from the perspective of the trustor, when there is no trustum related threat as follows:

Definition 1. *Let* $\mathtt{a}$ *and* $\mathtt{b}$ *be two actors, and* $\mathtt{s}$ *be a trustum (e.g., a goal* $\mathtt{g}$ *or information* $\mathtt{i}$*) that a aims for. Moreover,* $\mathtt{C}$ *and* $\mathtt{M}$ *are two sets of beliefs that* $\mathtt{a}$ *has about the capabilities and motivations of* $\mathtt{b}$*, respectively. We say that* $\mathtt{a}$ *trusts* $\mathtt{b}$ *for fulfilling* $\mathtt{s}$*,* $\mathtt{IFF}$ $\exists\ c_1 \in \mathtt{C}, \exists\ m_1 \in \mathtt{M} \mid$ *competence(fulfill, a, b, s)* $\wedge$ *positively_motivated(fulfill, a, b, s), and* $\nexists\ c_2 \in \mathtt{C}, \nexists\ m_2 \in \mathtt{M} \mid$ *incompetence(fulfill, a, b, s)* $\vee$ *negatively_motivated(fulfill, a, b, s), i.e.,* $\mathtt{a}$ *has only beliefs concerning the competency and positive motivation of* $\mathtt{b}$ *toward fulfilling* $\mathtt{s}$*.*

For example, in Fig. 1 a trust relation between the *Space Co* and *Account and audit Co* for fulfilling "Manage company's auditing operations" holds, if *Space Co* has beliefs about the competency and positive motivations of the *Account*

[1] We focus on the trustworthiness of the provided information, not information availability, i.e., whether information is provided or not.

[2] Fulfilling is used to refer to achieving a goal/threat, or provision of information.

and audit Co, and there is no beliefs concerning neither its incompetency nor negative motivations toward fulfilling the goal.

While if there is a trustum related threat(s), an integrated analysis of trust and its related threat(s) is required, i.e., we also need to determine whether the trustee is competence and positively motivated toward fulfilling the related threat(s) or not, and consider such knowledge while analyzing trust. In what follows, first we define a threat to trustum, and then we show how trust in the trustee can be analyzed when there is a trustum related threat(s).

Definition 2. *Let* `s` *be a trustum and* `t` *is a threat, we say that* `t` *is a threat to* `s`*,* `IFF` *the fulfillment of* `t` *threaten the fulfillment of* `s`*.*

For example, the *Space Co* can define "Biasing the audit information" as a threat to the goal "Manage company's auditing operation", where fulfilling such threat can threaten the fulfillment of the goal. Similarly, a *stock market* can define "Manipulate the market by providing falsified/fraud orders" as a threat to "trader's orders" information. After defining trustum related threat, we can enrich the trust analysis by considering the effect of such threat(s) over the trust relation as follows:

Definition 3. *Let* `a` *and* `b` *be actors,* `s` *is a trustum that* `a` *aims for,* `t` *is a threat of* `s`*, and* `a` *depends on* `b` *for fulfilling* `s`*. We say that* `a` *distrust* `b` *for fulfilling* `s`*,* `IF` `a` *has beliefs concerning the competencies and positive motivations of* `b` *toward fulfilling* `t`*.*

For example, in Fig. 1 *Space Co* distrusts *Account and audit Co* for fulfilling "Manage company's auditing operations", if it has beliefs concerning the competencies and positive motivations of *Account and audit Co* toward achieving the trustum related threat "Biasing the audit information". Similarly, the *stock market* distrusts a *stock trader* for providing "trader's orders", if it has beliefs concerning the competencies and positive motivations of the *trader* toward fulfilling the trustum related threat "Manipulate the market by providing falsified/fraud orders". Threat and threaten constructs are shown in Fig. 2.

Capture Actors' Competencies and Motivations Beliefs. Several belief sources can be used, Sabater and Sierra [19] identify three main sources of beliefs, namely: direct experiences, witness information, and sociological information. We mainly rely on *sociological information* that is information describes the social relations between agents and the roles they are playing in the society [19]. In particular, agents' capabilities can be derived from the role(s) they play, where roles' capabilities are predefined based on their different characterization in the society. Formally, this can be defined as follows:

Definition 4. *Let* `a` *be an agent,* `r` *is a role,* `s` *is a trustum, and* `t` *is a threat. We say that* `a` *is capable of fulfilling* `s/t`*,* `IFF` `a` *plays a role* `r` *that has the capability of fulfilling* `s/t`*. Otherwise, we say* `a` *is incapable of fulfilling* `s/t`*.*

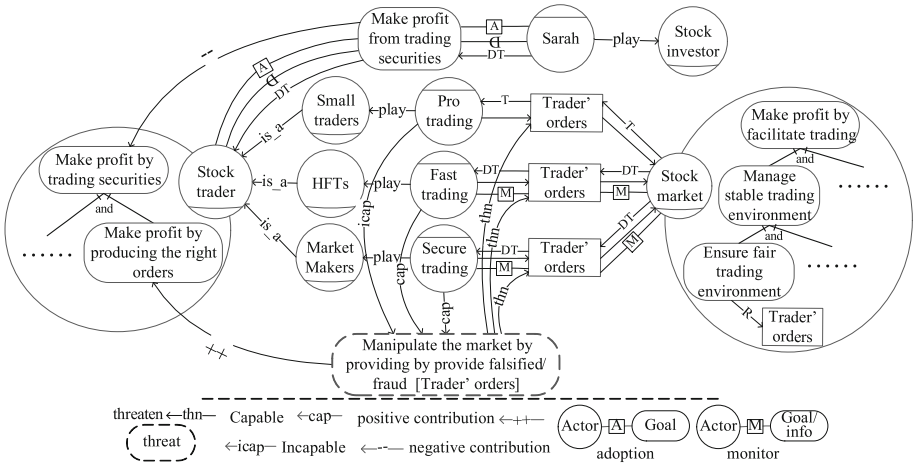

Fig. 2. A partial goal model of the stock market extended with belief-based constructs

Based on [15], in order to fulfill the threat "Manipulate the market by providing falsified/fraud orders", a *stock trader* has to be capable of trading with very high frequency (e.g., *HFT*), or it has to be capable of trading very large amount of securities (e.g., *Market Maker*), where such activities of *HFTs* and *Market Makers* were considered as a main reason of the Flash Crash. Actor's capability/incapability toward fulfilling a threat is modeled as edges between actors and threats labeled with `cap`/icap respectively (Fig. 2).

At the other hand, actors' motivations toward fulfilling a trustum/threat can be derived from the different interrelations among the actors own objectives and the trustum/threat, which can be captured by relying on the qualitative goal relationships [13], in which a goal can contribute positively or negatively towards the achievement of another one. In particular, an actor is positively motivated to fulfill a trustum/threat, if such trustum/threat contributes positively to at least one of its own goals. Similarly, an actor is negatively motivated to fulfill a trustum/threat, if such trustum/threat contributes negatively to at least one of its goals. Formally, this can be defined as follows:

Definition 5. *Let* `a`, `b` *be actors,* `s` *is a trustum,* `t` *is a threat, and* `g` *is a goal that* `b` *aims to achieve. We say that* `a` *beliefs that* `b` *is positively motivated to fulfill* `s/t`, `IFF` `s/t` *contributes positively to* `g`. *And we say that* `a` *beliefs that* `b` *is negatively motivated to fulfill* `s/t`, `IFF` `s/t` *contributes negatively to* `g`.

For example, *Space Co* delegates its goal "Manage company's auditing operations" to *Account and audit Co* that plays an *auditing firm*, which is capable of fulfilling the threat ("Biasing the audit information"), yet it does not have positive motivation towards fulfilling it. However, *Account and audit Co* plays another role that is *accounting firm*, which makes it positively motivated to fulfill the trustum related threat by several reasons (e.g., provides accounting services

Table 1. Analyzing trust based on the related beliefs

	Trustum beliefs				**Threat beliefs**			
	Competence		**Motivation**		**Competence**		**Motivation**	
	Comp	Incomp	Positive	Negative	Comp	Incomp	Positive	Negative
Trust	✓	X	✓	X	X	✓	-	-
Distrust	X	✓	-	-	-	-	-	-
	-	-	X	✓	-	-	-	-
	-	-	-	-	✓	X	✓	X

to a competing company, etc.). Positive and negative contribution among goals is modeled as edges labeled with ++ and −− respectively (Fig. 2).

Trust and Distrust Analysis. Previously, we have discussed how actors' competencies and motivations beliefs can be derived. Here, we show how these beliefs can be used to support trust analysis (shown in Table 1).

Trust: An actor *trusts* another one for a specific trustum, if it has belief(s) about the trustee's *competency* and *positive motivations* toward fulfilling the trustum, and it does not have belief(s) about the trustee's *incompetency* neither *negative motivations* toward fulfilling the trustum. Moreover, a trustor should have belief(s) about the trustee's *incompetence* to fulfill the trustum related threat (if any). For example in Fig. 2, a *stock market* trusts *Pro trading* for "trader's orders", since *Pro trading* has the competencies and it is positively motivated to provide them, and it is incompetent for fulfilling the related threat.

Distrust: An actor *distrusts* another one for a specific trustum, (1) if it does not have belief(s) about the trustee's *competency* toward the trustum, and has belief(s) about the trustee's *incompetency*; (2) if it does not have belief(s) about the *positive motivations* of the trustee toward the trustum, and it has belief(s) about its *negative motivations*; and (3) if it has belief(s) about the trustee's *competency* and *positive motivations* toward fulfilling the trustum related threat, and it does not have belief(s) about the trustee's *incompetency* and *negative motivations* toward fulfilling the threat. For example, in Fig. 2 *Sarah* distrusts the *stock trader* for fulfilling "Make profit from trading securities", since the *trader* might be *negatively motivated* to fulfill a goal that contributes negatively to one of its own goals. While a *stock market* distrusts both *Fast trading* and *Secure trading* for "trader's orders", since both of them have the *capability* and they are *positively motivated* toward fulfilling the trustum related threat "Manipulate the market by providing falsified/fraud orders".

Compensating the Lack of Trust. We propose two techniques to compensate the lack of trust (distrust):

Adopting can be defined as a commitment from the trustee toward the trustor that the first will behave as expected from the last toward the trustum [4]. We use *adopting* when the lacks of trust is due to belief(s) about the trustee's

negative motivations toward fulfilling the trustum. For example, *Sarah* can compensate the lacks of trust in a *stock trader* concerning the goal "Make profit from trading securities" by *adoption*, i.e., *stock trader* commits to *Sarah* to fulfill the goal as expected. Adopting is modeled as an edge labeled with `A` between the trustor and trustee at one hand, and the trustum at the other (Fig. 2).

Monitoring can be defined as the process of observing and analyzing the performance of an actor in order to detect any undesirable performance [24]. We use *monitoring* when the lacks of trust is due to belief(s) about the trustee's *competencies* and *positive motivations* toward fulfilling the trusum related threat. For example, a *stock market* should monitor the "trader's orders" that are provided by both *Fast trading* and *Secure trading* to compensate the distrust in them concerning the orders they provide. Monitoring is modeled as an edge labeled with `M` between the trustor and trustee at one hand, and the trustum at the other (Fig. 2).

4.2 Model Analysis and Verification

In order to enable the automated reasoning support, we used disjunctive Datalog [2] to formalize all the concepts that have been introduced in the paper, along with the required reasoning axioms[3]. Moreover, we defined a set of properties (shown in Table 2) that can be used to verify the trust requirements model, i.e., the model is correct and consistent, if all of these proprieties hold.

Pro1: states that the model should not include any goal that is not achieved from the perspective of the actor, who aims for it. We rely on this property to quickly verify the requirements model, i.e., if this property holds for all goals, we can conclude that the model is correctness and consistency.

Pro2 states that the model should not include any goal delegation/delegation chain, if there is no trust/trust chain between the delegator and the delegatee, or there is at least a compensation of the lack of trust among them. This property aims to detect any delegation with no trust, since such delegation leaves the delegator with no guarantee that its goal will be achieved. For example, in Fig. 1 if the trust relation did not hold between the *Space Co* and *Account and Audit Co* concerning the goal "Manage company's auditing operations", *Pro2* will be able to detect such violation.

Pro3 states that goals should be delegated only to actors, who have the capability to achieve them either by themselves or they have valid delegation to an actor who has such capability. For example, in Fig. 1 if *Space Co* delegates the goal "Manage company's auditing operations" to *Star Co*, which does not play the role of "Auditing firm", and it does not has a valid delegation to an actor that plays such role, *Pro3* is able to detect such false delegation.

Pro4 states that actors should have all information that is required for the achievement of the goals they are responsible of, i.e., this property is specialized for detecting information unavailability related issues. Capturing information

[3] The formalization of the concepts and axioms is omitted due to space limitation, yet it can be found at https://mohamadgharib.wordpress.com/bbta/.

Table 2. Properties of the design

Pro1	`:- aims(A,G), not achieved(A,G)`
Pro2	`:- deleChain(A,B,G), not trustChain(trust,A,B,achieve,G)`
Pro3	`:- deleChain(_,A,G), can_achieve(B,G), not can_achieve(A,G), not deleChain(A,B,G)`
Pro4	`:- needs(A,I), not has(A,I)`
Pro5	`:- prvChain(_,A,I), needs(B,I), not needs(A,I), not prvChain(A,B,I)`
Pro6	`:- need(G,I), is_responsible(A,G), not trusted(A,I)`
Pro7	`:- trustReq(Type,A,B,O,S), not trust(Type,A,B,O,S)`
Pro8	`:- motivation_belief(pos,A,B,achieve,G), motivation_belief(neg,A,B,achieve,G)`
Pro9	`:- motivation_belief(pos,A,B,achieve,G), adopt(A,B,achieve,G)`
Pro10	`:- monitor(A,B,S,O), not trust_threat(A,B,S,O)`

availability is not always an easy task, since goals might be interdependent on one another for information, and if a goal that produces information was not achieved (prevented), all goals that depend on its information will be prevented as well. For example, in Fig. 1 if the *stock trader's* goal "Internal analysis of targeted securities" was not achieved due to unavailability of "securities' assessments" or "financial statements" information, it will not produce "trader suggestions" that is required by both goals "Make profit by producing the right orders" and "Analysis depending on trader", i.e., both of them will not be achieved as well.

Pro5 states that information should be only provided to actors, who require them either for achieving their goals, or they have valid provision chain to an actor who requires them. For example, in Fig. 1 if "consulter suggestion" was provided to any actor but the *stock trader* or *stock investor* (e.g., stock market), *Pro5* will detect such violation and notify the analyst about it.

Pro6 states that the model should not include any untrusted information provision from the perspective of its reader. In other words, there is no guarantee that information is trusted for read if no trust/trust chain holds between information reader and producer or there is a compensation of the lack of trust among them. For example, in Fig. 2 trust will not hold between *Fast trading* and *Secure trading* at one hand and the *stock market* at the other concerning the provision of "Trader's orders". If such provision was not accompanied with a trust compensation mechanism (monitoring), *Pro6* will be able to detect such situation and notify the analyst about it.

Pro7 states that the model should not include any trust/distrust relation that conflicts with the trustee's competencies and motivations. For example, in Fig. 2 if the provision of "Trader's orders" between *Fast trading* and *stock market* was modeled as trust, *Pro7* will notify the analyst that such trust relation is conflicting with the competencies and motivations of *Fast trading* toward the trustum and its related threat, i.e., it should be modeled as distrust. Similarly,

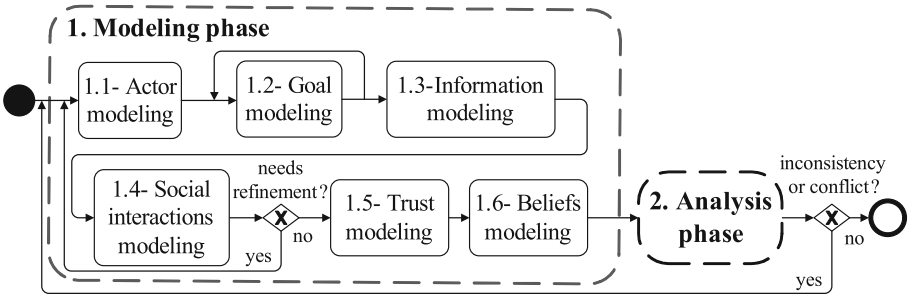

Fig. 3. Methodological process for analyzing trust requirements

if the provision of "Trader's orders" between *Pro trading* and *stock market* was modeled as distrust, *Pro7* will detect such situation and notify the analyst that it should be modeled as trust, since such distrust is conflicting with the capabilities and motivations of *Pro trading* toward the trustum and its related threat.

Pro8 states that the model should not include actors with conflicting motivational beliefs (positively and negatively motivated) toward the same trustum/threat. We avoid having such situation, since our reasoning do not provide techniques for motivations conflict resolution yet.

Pro9 states that the model should not include an adoption relation between a trustor and a trustee, if the first beliefs that the last is positively motivated to fulfill the trustum. For example, if *Space Co* delegates its goal "Manage company's auditing operations" to an *auditing firms*, and there is also an adopting relation concerning the delegation, *Pro9* will notify the analyst that such adoption relation is not required, since there is no beliefs concerning the negative motivations of the *auditing firms* toward fulfilling the delegated goal.

Pro10 states that the model should not include a monitoring relation between a trustor and a trustee, if the last cannot be considered as a possible threat source for the trustum. For example, if the model has a monitoring relation concerning "trader's orders" that are provided from a *small trader* to *stock market*, *Pro10* is able to detect such situation and notify the analyst that such monitoring relation is not required.

4.3 Methodological Process for Analyzing Trust Requirements

Our approach is equipped with an engineering methodological process (shown in Fig. 3) that is specialized for modeling and analyzing trust requirements. The process consists of two main phases; we describe them as follows:

(1) **Modeling Phase**: This phase aims to model the requirements of the system with special emphasis on the trust requirements; and it is composed of six steps: (1.1) *Actors modeling*: in which the stakeholders of the system are identified and modeled in terms of agents and roles along with their objectives, entitlements and capabilities; (1.2) *Goals modeling*: the

stockholders' top-level goals are modeled and refined (if needed) through And/Or-decomposition until reaching their leaf goals, which might leads to new iteration of this step; (1.3) *Information modeling*: in which the different relations between goals and information are modeled (e.g., produce, read, etc.); and (1.4) *Social interactions modeling*: aims to model the different social dependencies among actors concerning information provisions and goal delegation. In other words, based on actors' capabilities some goals might be delegated to actors, who have the capabilities to achieve them; and based on actors' needs, information is provided to them.
After step 4, the model is checked and if it needs to be refined (e.g., adding new actors, refining some goal, etc.), we restart the modeling phase again; otherwise we proceed to the next step. (1.5) *Trust modeling*: trust requirements among actors concerning both goal delegation and information provisions are modeled at this step; (1.6) *Beliefs modeling*: aims to model threats, goals contributions, and the actors' competencies and motivations beliefs toward such threats. After finishing this step, the designer proceeds to the analysis phase.

(2) **Analysis Phase**: The aim of this phase is to support the verification of correctness and consistency of the requirements model. In particular, we define a set of properties of the design that helps in identifying any inconsistency or conflict in the model and notify the designer about them to find proper solutions. After the analysis phase the model is check, and the process reaches its end if the model is verified, otherwise the process restart again from the beginning to address the conflicts/inconsistencies in the model that have been identified (detected) during the analysis phase.

5 Implementation and Evaluation

Evaluation is an important aspect of any research proposal, and it aims to demonstrate the utility and efficacy of a design artifact. To this end, we evaluated our approach on a simulation basis following [14], by developing a prototype tool and test its applicability with artificial data. In particular, we developed IQ-Tool[4] (a screenshot of IQ-Tool is shown in Fig. 4) that is a prototype tool of our framework to test its applicability and effectiveness for modeling and analyzing trust requirements. In what follows, we briefly describe the tool, discuss its applicability and effectiveness over the Flash Crash case study, and then we discuss the scalability experiments we performed. Finally, we theoretically evaluate our proposed modeling language based on the semiotic clarity principle.

Implementation: The tool consist of three main parts: (1) A Graphical User Interface (GUI) that enables for drawing the requirements model by drag-and-drop modeling elements from different palettes; (2) Model-to-text transformation mechanism that supports the translating of the graphical models into disjunctive Datalog formal specifications; and (3) automated reasoning support (DLV

[4] https://mohamadgharib.wordpress.com/bbta/.

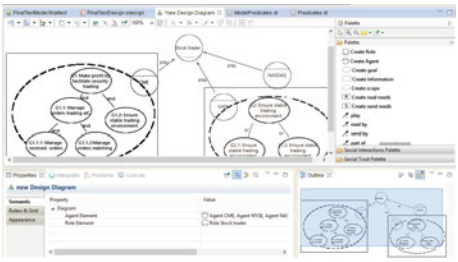

Fig. 4. Screenshot of the IQ-Tool

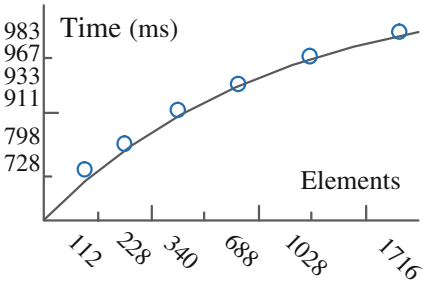

Fig. 5. Scalability experiment result

system[5]) takes the disjunctive Datalog specifications along with already defined reasoning axioms as an input, and then perform the required analysis to verify the trust requirements model against the properties of the design.

Applicability: To test the framework applicability, we modeled the Flash Crash case study, transformed it into disjunctive Datalog specifications, and then we run the automated analysis to test its ability in discovering violations to the properties of the design. The analysis returned several violations, including: a *stock market* considers orders received from any agent that plays either *HFTs* or *Market makers* roles as untrusted information, which we solved by monitoring the trustee. *Space Co* distrusts *Account and audit Co* for achieving the goal "manage company's auditing operation", which we solved by compensating the lack of trust by adopting. Moreover, the analysis detects all situations where agents became negatively motivated to fulfill a trustum, and when they became positively motivated to fulfill a threat when they play several roles that may conflict. For example, *Star Co* plays both *Credit assessment rating firms* and *Consulting firm* roles, which influences the trust in it concerning information it produces (e.g., consulter suggestions and securities assessments). In particular, the proposed analysis techniques were able to capture all the violations to the properties of the design that we consider in this paper, which enable in turn to avoid the different vulnerabilities in the system design that led or contributed to the Flash Crash and resolve them during the early requirements phase.

Experiments on Scalability: To test the scalability of the reasoning technique, we expanded the model shown in Fig. 1 by increasing the number of its modeling elements from 122 to 1716 through six steps, and we investigated the reasoning execution time at each step by repeating the reasoning execution seven times, discarding the fastest and slowest ones, and then computed the average execution time of the rest. The result is shown in Fig. 5, and it is clear that the relation between the size of the model and execution time is not exponential. We have performed the experiment on laptop computer, Intel(R) core(TM) i3-3227U CPU@ 190 GHz, 4 GB RAM, OS Window 8, 64-bit.

[5] http://www.dlvsystem.com/dlv/.

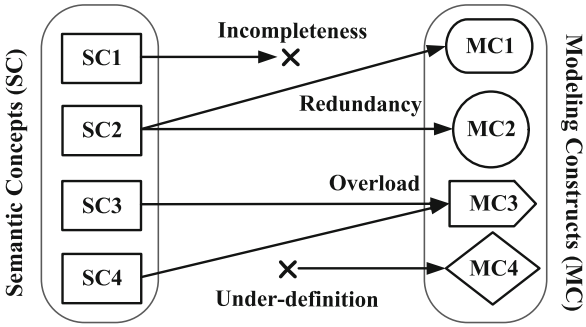

Fig. 6. Principle of semiotic clarity

Theoretical Evaluation: We evaluated our modeling language according to the principle of semiotic clarity [16], which helps in increasing the expressiveness and precision of the language. In this principle there should be a one-to-one mapping between semantic concepts and their corresponding constructs (Fig. 6), i.e., there should be no incompleteness, overload, redundancy, or under-definition between the semantic concepts and their corresponding language constructs.

Incompleteness: Occurs when a semantic concept is not represented by any modeling construct. We have only one case, we did not model the actor capability/incapability toward goals as we did for threats. However, symbol incompleteness is not necessarily a bad thing, since representing only the required concepts helps in keeping the language complexity manageable [16].

Redundancy: Occurs when multiple modeling constructs represent a single semantic concept. We did not observe any symbol redundancy in the language.

Overload: Occurs when a single modeling construct is used to represent multiple semantic concepts. We have the positive/negative contributions among goals and goals/threats, which can be considered as an overload. However, actors can differentiate between them easily based on the context. Moreover, we only represented actor's capability toward threats, and omitted representing its capability toward goals in order to avoid an overload.

Under-definition: Occurs when a modeling construct does not represent any semantic concept. There are no under-definition issues in the language.

6 Related Work

A large body of literature has focused on how trust can be constructed (built). For instance, Marsh [22] proposes one of the earliest trust models that consider direct interaction as a main source of trust related beliefs. Schillo et al. [20] introduce a trust model based on probability theory, which can be used when trust among agents has a Boolean value (good or bad). While in [1], trust can be build based on agents' beliefs in one another concerning their experience and

reputation, and the degrees of trust range from complete distrust to complete trust. Esfandiary and Chandrasekharan [8] propose a cognitive based trust and reputation model. In which, trust can be built based on observation and interaction. Finally, Castelfranchi and Falcone [4] propose a cognitive trust model, in which different types of beliefs can be used to build trust. Although most of these works propose techniques for constructing trust, but they do not offer concepts or constructs for modeling or reasoning about trust requirements.

At the other hand, trust is still a new research thread within the RE community. However, several RE approaches suggested concepts for modeling trust. For instance, in [11] they focus on capturing trust at the level of roles. While in [12], they capture trust at two different levels (roles and agents), and they highlighted the problem that may arise when a trusted role is played by an untrusted individual (agent). In [24] trust was introduced as a fundamental aspect for making decisions on security. While Asnar et al. [3] introduce the notion of trust for assessing risk. Finally, Chopra et al. [5] introduce architectural trust that can be used to assist the trustworthiness of the overall STS. Although, most of these works propose constructs for modeling trust requirements, but none of them consider actors' competencies (capabilities), motivations, etc. as main drivers while analyzing trust requirements, i.e., they do not provide a specialized concepts nor constructs for modeling and analyzing trust requirements based on actors' competencies and motivations.

7 Conclusion and Future Work

We have proposed a belief-based trust approach that combines both cognitive and mathematical trust approaches, introducing concepts and constructs for modeling and analyzing trust based on sets of beliefs related to the actors' competencies and motivations toward the trustum and its related threat(s) (if any). Moreover, we have discussed the automated reasoning techniques our framework offers, and how they can be used to verify the correctness and consistency of the trust requirements model. In addition, we described the methodological process underlies our approach should be followed by designers/analysts during the system design. We evaluated our approach on a simulation basis by developing a prototype tool and test its applicability and effectiveness for modeling and analyzing trust requirements. Finally, we theoretically evaluated our proposed modeling language based on the semiotic clarity principle.

For the future work, we are exploring how to extend our approach by adding new types of beliefs that might influence the trust among actors. Moreover, we aim to enrich our approach with mechanisms to combine different kinds of beliefs while analyzing trust, and also with mechanisms for conflict resolution among different kinds of beliefs. Furthermore, we are planning to evaluate our approach with end users (e.g., designers, analysts, experts, etc.) to assess its adequacy for modeling and reasoning about trust requirements. Finally, we aim to better validate the approach by applying it to several complex case studies that belong to different domains.

Acknowledgments. This research has received funding from the ERC advanced grant 267856, "Lucretius: Foundations for Software Evolution", http://www.lucretius.eu/, and the European Unions Horizon 2020 research and innovation programme under grant agreement No 653642 - VisiOn.

References

1. Abdul-Rahman, A., Hailes. S.: Supporting trust in virtual communities. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000, p. 9. IEEE (2000)
2. Alviano, M., Faber, W., Leone, N., Perri, S., Pfeifer, G., Terracina, G.: The disjunctive datalog system DLV. In: de Moor, O., Gottlob, G., Furche, T., Sellers, A. (eds.) Datalog 2010. LNCS, vol. 6702, pp. 282–301. Springer, Heidelberg (2011)
3. Asnar, Y., Giorgini, P., Massacci, F., Zannone, N.: From trust to dependability through risk analysis (2007)
4. Castelfranchi, C., Falcone, R.: Social trust: a cognitive approach. In: Castelfranchi, C., Tan, Y.-H. (eds.) Trust and Deception in Virtual Societies, pp. 55–90. Springer, Netherlands (2001)
5. Chopra, A.K., Paja, E., Giorgini, P.: Sociotechnical trust: an architectural approach. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 104–117. Springer, Heidelberg (2011)
6. Chopra, K., Wallace, W.A.: Trust in electronic environments. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003, p. 10. IEEE (2003)
7. Emery, F.E., Trist, E.L.: Socio-technical systems. In: Churchman, C.W. (eds.) Management sciences, models and techniques (1960)
8. Esfandiari, B., Chandrasekharan, S.: On how agents make friends: mechanisms for trust acquisition. In: Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies, Montreal, Canada, pp. 27–34 (2001)
9. Gharib, M., Giorgini, P.: Detecting conflicts in information quality requirements: the May 6, 2010 flash crash. Technical report, Universitá degli studi di Trento (2014)
10. Gharib, M., Giorgini, P.: Modeling and reasoning about information quality requirements. In: Fricker, S.A., Schneider, K. (eds.) REFSQ 2015. LNCS, vol. 9013, pp. 49–64. Springer, Heidelberg (2015)
11. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Filling the gap between requirements engineering and public key/trust management infrastructures. In: Katsikas, S.K., Gritzalis, S., López, J. (eds.) EuroPKI 2004. LNCS, vol. 3093, pp. 98–111. Springer, Heidelberg (2004)
12. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Modeling social and individual trust in requirements engineering methodologies. In: Herrmann, P., Issarny, V., Shiu, S.C.K. (eds.) iTrust 2005. LNCS, vol. 3477, pp. 161–176. Springer, Heidelberg (2005)
13. Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R.: Reasoning with goal models. Conceptual Model. ER **2002**, 167–181 (2003)
14. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quart. **28**(1), 75–105 (2004)
15. Kirilenko, A., Kyle, A.S., Samadi, M., Tuzun, T.: The Flash Crash: The impact of high frequency trading on an electronic market. Manuscript, University of Maryland (2011)

16. Moody, D.L.: The physics of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Trans. Softw. Eng. **35**(6), 756–779 (2009)
17. Mouratidis, H., Giorgini, P.: Secure tropos: a security-oriented extension of the tropos methodology. Int. J. Softw. Eng. Knowl. Eng. **17**(2), 285–309 (2007)
18. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture, pp. 473–484. Kaufmann publishers, San Mateo (1991)
19. Sabater, J., Sierra, C.: Review on computational trust and reputation models. Artif. Intell. Rev. **24**(1), 33–60 (2005)
20. Schillo, M., Funk, P., Rovatsos, M.: Using trust for detecting deceitful agents in artificial societies. Appl. Artif. Intell. **14**(8), 825–848 (2000)
21. Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., Mcdermid, J., Paige, R.: Large-scale complex IT systems. Commun. ACM **55**(7), 71–77 (2012)
22. Stephen, M.: Formalising trust as a computational concept. Ph.D. dissertation. University of Stirling, Scotland (1994)
23. Van Lamsweerde, A.: Elaborating security requirements by construction of intentional anti-models. In: Proceedings of the 26th International Conference on Software Engineering, pp. 148–157. IEEE Computer Society (2004)
24. Zannone, N.: A requirements engineering methodology for trust, security, and privacy. Ph.D. thesis, University of Trento (2006)