Department of
Information Engineering
and Computer Science
DISI

UNIVERSITY
OF TRENTO - Italy

DISI - Via Sommarive, 5 - 38123 POVO, Trento - Italy
http://disi.unitn.it

Detecting Conflicts in Information Quality Requirements: the May 6, 2010 Flash Crash

Mohamad Gharib and Paolo Giorgini

October 2014

Technical Report # DISI-14-016

Detecting Conflicts in Information Quality Requirements: the May 6, 2010 Flash Crash

Mohamad Gharib and Paolo Giorgini

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

Abstract

Information Quality (IQ) is a key success factor for the efficient performance of any system, and it becomes a vital issue for critical systems, where low-quality information may lead to disasters. Despite this, most of the Requirements Engineering frameworks loosely define, or simply ignore such requirements, which may lead to different conflicts among the stakeholders' IQ requirements. In this paper, we propose a novel conceptual framework for modeling and reasoning about IQ at requirements level. The proposed framework is based on the secure Tropos methodology and extends it with the required concepts for modeling and analyzing IQ requirements since the early phases of software development. A running example concerning a U.S stock market crash (the May 6, 2010 Flash Crash) is used throughout the paper.

Keywords

Requirements Engineering, Information Quality, Modeling, Reasoning

1 Introduction

Information Quality (IQ) is a key success factor for organizations, since depending on low-quality information may cause severe consequences [32], or

even disasters in the case of critical systems. Despite its importance, IQ is often loosely defined, or simply ignored [15]. In general, quality has been defined as "fitness for use" [18], or as in [33] the conformance to specifications, i.e., meeting or exceeding consumer expectations.

For example, consider a stock market investor who uses his laptop to trade some securities, the level of IQ required by him concerning his trades is not the same as the IQ level required by a main stock market (e.g., NYSE, NASDAQ) that is responsible of managing thousands of trades in milliseconds simultaneously. In the first case, low-quality information can be accepted to a certain level, while in the second case it may result in a financial disaster (e.g., stock market crash, or at least loses of millions of dollars).

Several techniques for dealing with IQ have been proposed in the literature (e.g., integrity constraints [27]). However, they mainly focus on technical aspects of IQ and do not solve problems that may rise at organizational or social levels. More specifically, these techniques do not satisfy the needs of complex systems these days, such as socio-technical systems [12], where humans and organizations are integral part of the system along with the technical elements such as software and hardware (e.g., healthcare systems, smart cities, etc.). In these cases, requirements about IQ should be extended to a socio-technical analysis.

For example, the Flash Crash was not caused by a mere technical failure, but it was due to undetected vulnerabilities that manifested themselves in the interactions of the stock market systems that led to a failure in overall socio-technical system [40]. In particular, several reasons contributed to the Flash Crash were caused by socio-technical IQ related issues. For instance, according to [20] some traders intentionally provide falsified information. Others continue trading during the crash by forwarding their orders to the markets that did not halt their trading activities due to lake of coordination among the markets, where the lack of coordination resulted also from IQ related vulnerabilities. More specifically, most of these issues resulted from conflicts among the IQ requirements of the stakeholders of the system. However, such failures could be avoided if the IQ requirements of the system-to-be were captured properly during the system design.

We advocate that answering "why" IQ related mechanisms and solutions are needed, and not just "what" mechanisms and solutions are needed to solve IQ related problems can provide a better understanding of stakeholders' needs that are beyond IQ requirements. Moreover, it enables for detecting any IQ requirements conflicts and resolving them at the early phases of the

system design.

The framework presented in this paper uses a Goal-Oriented Requirements Engineering (GORE) approach. Among the several GORE approaches offered in the literature (e.g., KAOS [9], i^* [47]), we adopted secure Tropos [28] as a baseline for our framework. Secure Tropos introduces primitives for modeling actors of the system along with their objectives, entitlements and capabilities. Goals are used to represent the strategic interest of actors, and can be refined through And/ Or decomposition of a root goal into sub-goals. Resources are used to represent both physical and informational entities that are needed/ produced for/by the achievement of goals. Moreover, secure Tropos provides the notion of delegation to model the transfer of responsibilities among actors of the system.

Finally, it adopts the notion of trust and distrust to capture the expectations of a trustor in the behavior of a trustee concerning a trustum. Our framework extends the conceptual framework of secure Tropos by providing the required concepts and constructs for modeling and reasoning about IQ requirements. It allows the analyst to identify clearly "why" a certain level of quality of a specific information is needed and not only "what" and "where" such information is needed.

The paper is organized as follows; Section (§2) describes our motivating example, while in Section (§3) we discuss the different problems related to capturing IQ. In Section (§4), we outline the limitation in secure Tropos for dealing with IQ, and then we propose the required extensions. In Section (§5), we present the reasoning techniques that our framework offers. Section (§6) implement and evaluates the proposed framework. Section (§7) presents the related work. Finally, we conclude and discuss the future work at Section (§8).

2 Motivating Example

Our motivating example concerns the May 6, 2010 U.S stock Flash Crash, in which the Dow Jones Industrial Average (DJIA) dropped about 1000 points (9% of its value) , and then it recovers those losses within minutes. This section is organized as follows, (1) we briefly describe the main stock market stakeholders along with their goals; (2) we list the 2010 Flash Crash chronology of events; then (3) we discuss the main theories about the reasons that led to the Flash Crash.

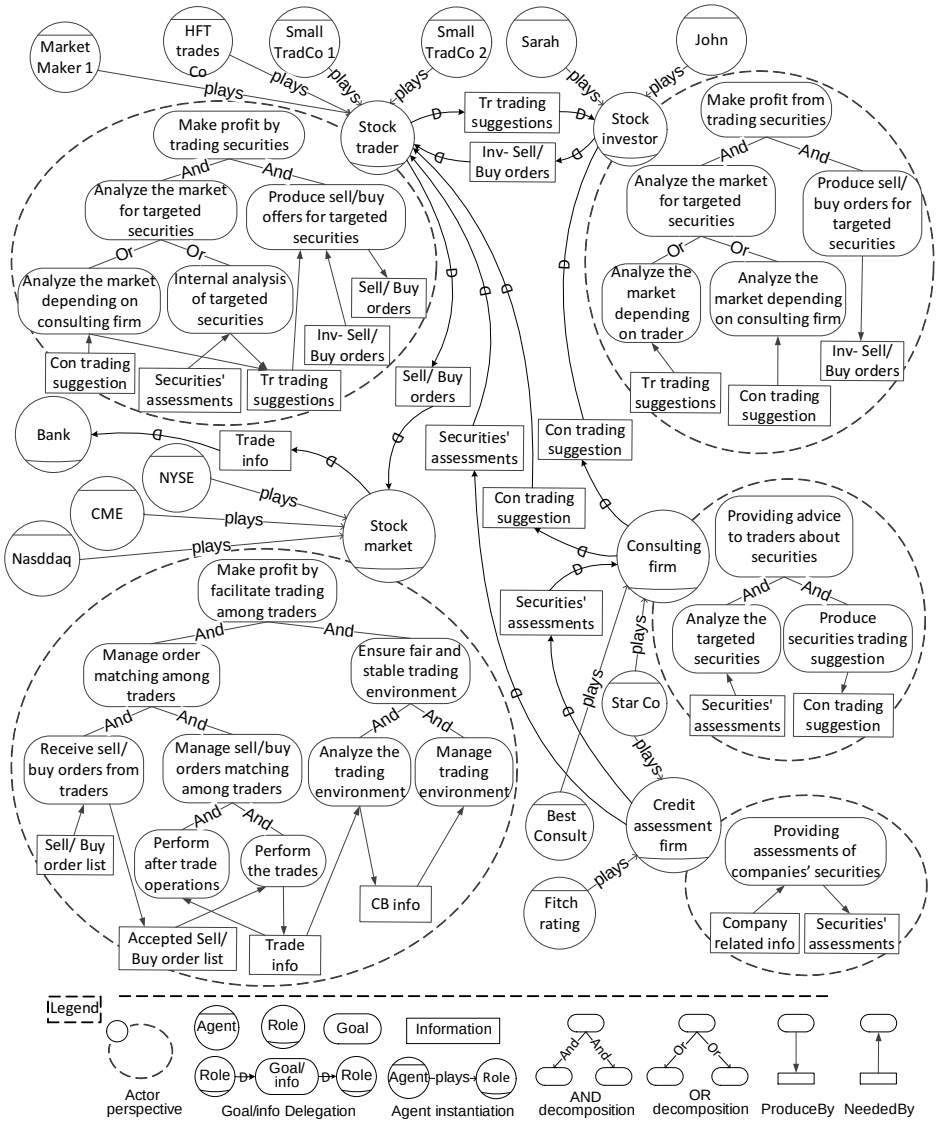

Figure 1: A partial goal model concerning the U.S stock market structure

2.1 The stock market system structure

Based on [20], we can identify several stakeholders including: *stock investors* are individuals or companies, who have a main goal of "making profit from trading securities", which is And decomposed into two goals "Produce sell/buy

orders for targeted securities" and "Analyze the market for targeted securities", where the first goal produces "Inv- Sell/ Buy orders". While the last goal is Or decomposed into two goals, "Analyze the market depending on trader" that needs to consume "Tr trading suggestions" (provided by a *trader*), and "Analyze the market depending on consulting firm" that needs to consume "Con trading suggestion" (provided by a *consulting firm*).

Stock traders are persons or companies involved in trading securities in *stock markets* with a main goal of "making profit by trading securities" either for their own sake or by trading on behalf of their *investors*. According to [20], *traders* can be classified under several categories, including: *Fundamental traders*: are able to either buy or sell a significant number of securities with a low trading frequency rate; *Market Makers*: facilitate trading on a particular security in the market, and they are able to trade large number of securities; *High-Frequency Traders (HFTs)*: are able to trade with very high trading frequency; *Small traders*: trade small amount of securities with very low trading frequency.

While *stock markets* are places where *traders* gather and trade securities, which have a main goal of "Make profit by facilitating the trades among stock traders" that is And decomposed into two sub goals "Manage order matching among traders" and "Ensure fair and stable trading environment", where the first intend to receive, match and perform orders from different *traders*, and the last is responsible of halting or slowing down the trading frequency in order to stabilize the trading environment when necessary. Moreover, *consulting firms* are firms specialized for providing professional advices concerning financial securities to *traders* and *investors* for a fee. Finally, *credit assessment ratings firms* are firms with a main objective of providing assessments of the credit worthiness of companies' securities, i.e., such firms help *traders* in deciding how risky it is to invest money in a certain security.

Figure 1 shows a portion of the secure Tropos representation of the stock market structure. Secure Tropos is able to capture the social/ organizational context of the system, but it does not offer primitives to model needs about IQ, i.e., it deals with information whether they are available or not and who is responsible about their delivery. For example, secure Tropos is able to model information provision between investors and traders, and between traders and markets. Yet, it does not provide concepts that enable to analyze the quality of the provided information (e.g., accurate, complete, consistent, etc.).

2.2 The Flash Crash: chronology of events

The following sequence of events is based on both the joint report of CFTC and SEC regarding the market events of May 6, 2010 [42, 36] and Nanex[1] Flash Crash summary report [1].

- On May 6, U.S. stock markets opened and trended down for most of the day on worries about the European debt crisis (Greece).
- By 2:30 p.m., the selling pressure had pushed the DJIA down about 2.5% of its value.
- At 2:32 p.m., a large fundamental trader initiated a sell program to sell a total of 75,000 E-Mini contracts (valued at approximately $4.1 billion). The sell was initially absorbed by HFTs and fundamental buyers. Usually, such big sell order may take more than 5 hours to execute. However, on May 6, it was executed extremely fast in only 20 minutes.
- Between 2:41 p.m. and 2:44 p.m., HFTs and other traders drove the price of the E-Mini down by more than 5%.
- At 2:45:28 p.m., trading on the E-Mini was paused for five seconds when the Chicago Mercantile Exchange (CME), CME Circuit Breakers (CB)[2] was triggered in order to prevent a cascade of further price declines. Yet NYSE did not halt trading [41].
- At 2:45:33 p.m., prices stabilized when trading resumed, and the E-Mini began to recover.

In summary, between 2:40 p.m. and 3:00 p.m., approximately 2 billion shares were traded with a total volume exceeding $56 billion. Over 98% of all shares were executed at prices within 10% of their 2:40 p.m. value.

[1]Nanex is a firm that offers streaming data on all market transactions and distributes the data in real time to clients and allows them to do analysis and visualization in real time

[2]CB is a technique that is used by markets to halt or slow trading in order to prevent potential market failure [16]

2.3 Main reasons of the Flash Crash

The Flash Crash has raised many questions concerning the efficiency of the information system supporting the stock market. A deep analysis of the Flash Crash shows that many of the reasons that led to the failure can be avoided if the IQ requirements of the system were captured properly during the early phases of the system design. Several researchers investigated specific cases of IQ and their effects on the overall performance of the stock market (e.g., [13, 13]),and different theories have been proposed to explain what happened, including:

- Fat-finger trade that is a human error caused by pressing a wrong key when using a computer to input data. However, this theory was quickly disproved after it was determined that the E-mini S&P 500 contracts (the trade that was under suspicion of triggering the Flash Crash) was not a result of a fat-finger trade [11, 11];

- The highly fragmented nature of the finical market along with the inefficient coordination mechanisms among the CBs of the trading markets also played a role in the Flash Crash [16, 41]. More specifically, trading markets should coordinate their CBs. Otherwise, HFTs will simply search for a market other than the closed ones and continue trading [24]. For instance, during the Flash Crash CME employs its CB but NYSE did not [41].

- The behavior of HFTs that effects the market prices and contributed to the Flash Crash[36, 7, 20].

- Fraud information (intentionally falsified information [23]) that have been used by some actors and compromised the overall system performance (e.g., HFTs' flickering quotes[3] [26], Market Makers' stub quotes[4], such orders can also be considered as falsified information; since they are orders were not intended to be performed [20], etc.).

- Sommerville et al. [40] argue that the failure(s) , which led to the Flash Crash was not caused by a mere software (technical) failure(s), but it was due to undetected vulnerabilities that manifested themselves in the

[3]Quotes that last very short time, which make them unavailable for most of traders
[4]Orders with prices far away from the current market prices

interactions of independently-managed software systems that led to a failure in the socio-technical systems in which the HFTs operate. In other words, the Flash Crash can be best understood as a failure in a large-scale complex socio-technical system [8].

3 Capturing Information Quality

The quality of information can be defined based on its "fitness for use", yet such definition do not explicitly capture the "fitness for use" for "what" and the "fitness for use" of "who", which is very important when information have several stakeholders, who may require different (might be conflicting) quality needs. In other words, existing definitions of IQ lake a clear semantics to capture IQ requirements taking into consideration the different needs of their stakeholders. Without having such semantics, it is hard to determine whether IQ "fits for use" or not.

Several IQ models and approaches have been propose [23, 31], yet most of them propose a holistic method for analyzing IQ (one size fits all), i.e., they consider a user-centric view [46] without taking into consideration the relation between information and its different purposes of usage. For example, in Figure 1 we can see a stock investor (e.g., John) who wants to send a sell/ buy order to a stock market through a stock trader. This simple scenario raises several questions: Do all the stakeholders (e.g., investor, trader, and stock market) have the same purpose of information usage? How we can define the quality of the buy/sell order based on the different purposes of usage? Should the stakeholders require the same quality of information? If not, how do their needs differ? Actually, the previous questions cannot be properly answered without defining a clear semantics among information, its quality, and the stakeholders' intended purposes of information usage.

Moreover, IQ can be characterized by different dimensions [43, 6] that can be used to analyze IQ, including: accuracy, completeness, consistency, timelines, accessibility, trustworthiness, etc. However, we only focus on 4 IQ dimensions, namely: accuracy, completeness, timeliness and consistency, since they are the main IQ dimensions, and they enable us to address the IQ related problems we consider in this paper. These dimensions can be defined as follows: **Accuracy**: means that information should be true or error free with respect to some known, designated or measured value[5]; **Completeness**: means that all parts of information should be available [5, 43];

Timeliness: means to which extent information is valid in term of time [31]; **Consistency**: means that multiple records of the same information should be the same across time [5].

After defining these dimensions, we need to ask several more questions, should the different stakeholders consider the same IQ dimensions for analyzing IQ? Do they analyze these dimensions by the same ways? For instance, can information validity be analyzed by an actor who requires to send information, and an actor who requires to receive (read) information by the same way? The same question can be asked about other dimensions. Moreover, most of the proposed IQ approaches ignore the social/ intentional aspects that underlie some of these IQ dimensions. Ignoring such aspects during the system design leaves the system open to different kinds of vulnerabilities that might lead to various kinds of failures (e.g., actors might intentionally provide falsified information).

4 Extending secure Tropos with IQ modeling concepts

In order to capture the stakeholders' requirements concerning IQ, secure Tropos modeling language needs to be able to provide the required concepts and constructs for capturing the stakeholders' different purposes of information usage, and the different relations among the purposes of usage and IQ in terms of its dimensions. From this perspective, we extend the conceptual model of secure Tropos to accommodate the following concepts:

Goal-Information interrelation: we need to provide the required concepts to capture the different relations between goals and information usage. Thus, we extend secure Tropos by introducing 3 different concepts that are able to capture such relations:

Produces : indicates that an information item can be created by achieving the goal that is responsible of its creation process;

Reads : indicates that a goal consume an information item. Reads relation can be strictly classified under, *Optional*: indicates that information is not required for the goal achievement, i.e., the goal can be achieved even such information has not been provided; *Required*: indicates that information is required for the goal achievement, i.e., the goal cannot be achieved without reading such information;

Sends : indicates that the goal achievement depends on transferring an information item under predefined criteria to a specific destination.

For instance, in Figure 2 achieving the goal "Perform the trades" produces "Trade information". While the goal "Receive sell/buy orders from traders" optionally reads the "Sell/ Buy orders", since the goal will be achieved regardless the number of the received sell/buy orders. While goal "Manage trading environment" requires to read "CME CB info". At the other hand, the goal "Perform after sale operations" needs to send "Trade info" to the bank that is responsible of finalizing the trade. These different relations are represented as edges labeled with *produce*, *send[destination][time]*, *read [R]* and *read [O]* to represent produces, sends, optionally read and required read respectively.

Information accuracy: we need to provide the required concepts that enable for deciding whether information is accurate or not from different perspectives of its stakeholders. In particular, information accuracy can be analyzed based on its production process, since information can be seen as product [3, 37], and many of the product quality concepts can be applied to it. In other words, the accuracy of information is highly affected by its source [10]. Moreover, actors might depend on one another for information to be provided, and the provision process might also affect the accuracy of the provided information. More specifically, the accuracy of information can be analyzed based on its sources along with its provision process.

We rely on the notion of trust that has been proposed in secure Tropos to analyze the accuracy of information based on its source (trusted/distrusted source) and provision process (trusted /distrusted provision). For instance, a *market* considers information it receives as accurate, if a trust relation holds between the *market* and information source (e.g., *trader*), and if information has been provided through a trusted provision. The same can be applied to information that is send, i.e., send information is accurate from the perspective of its sender, if a trusted provision holds between the sender and the final destination of information. Such relation is shown in Figure 2 as edges labeled with T concerning the provided information ("Inv sell/buy orders") between John (*investor* and Small marketCo1 (*stock market*).

Information completeness: we need to provide the required concepts to capture the relation between an information item and its sub-items (if any), which enables us to decide whether information is complete for achieving a specific goal or not. We rely on the *"part of"* concept that has been used

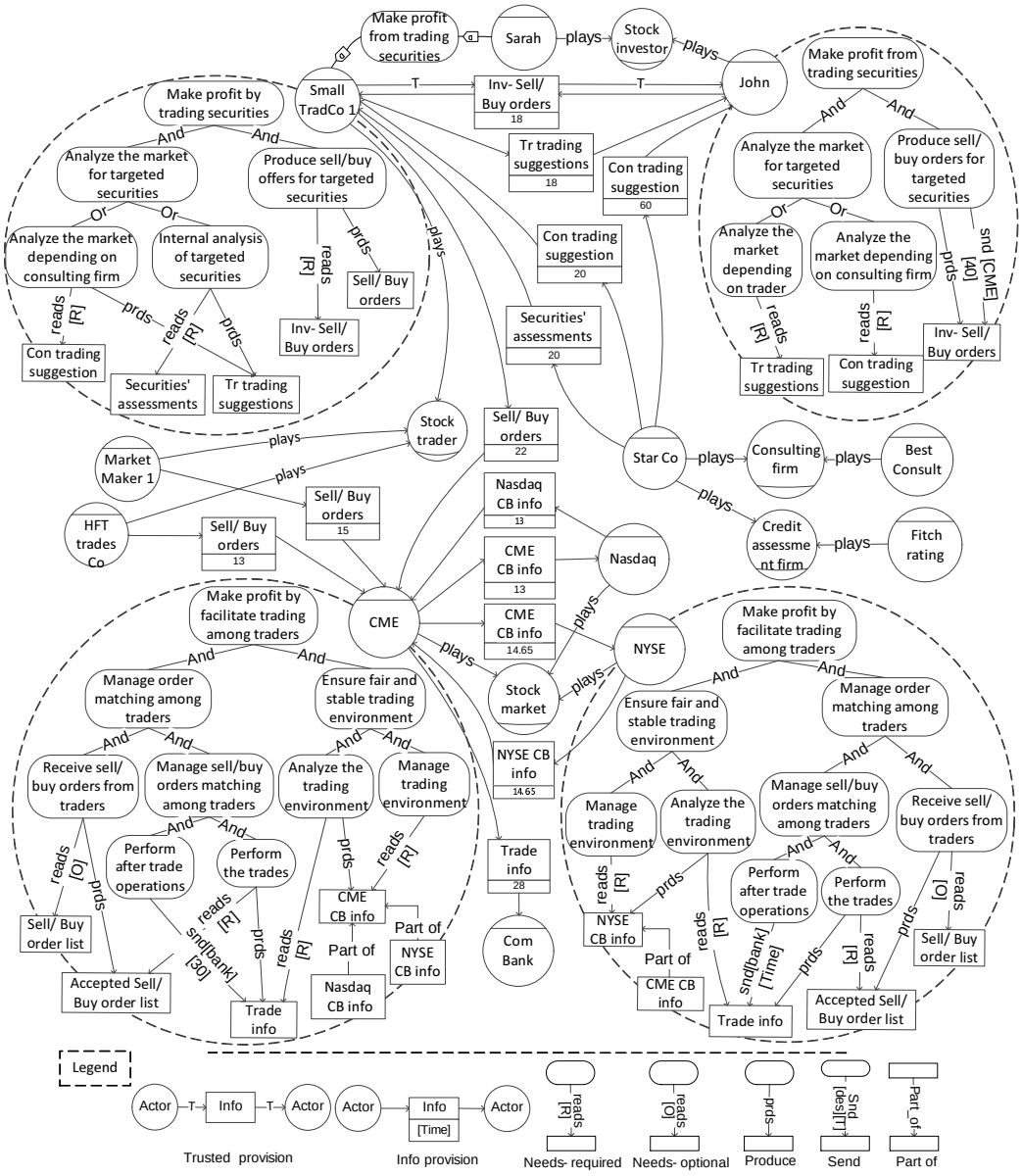

Figure 2: A partial goal model of the Flash Crash extended with IQ related constructs

in several areas (e.g., natural language, conceptual modeling, etc.) to model the relation between an information item and its sub parts. Moreover, we

provide the *purpose of use* along with*related to the purpose of use* concepts to capture information completeness for achieving a specific goal (information is complete for achieving a goal), where the first concept is used to capture the intended purpose of information usage, and the last is used to define all information sub parts related the defined *purpose of use*.

For example, one main reason of the Flash Crash was the effect of uncoordinated Circuit Breaker CBs among the *markets*. Such failure resulted due to depending on incomplete information by *markets* for their CBs. In particular, in stock market domain, the same security might be traded in different markets. Thus, in order to coordinate the CBs activities among the different markets that trade the same security, markets should be aware of one another activities concerning any change in the trading frequency. In other words, when a market halts or go into slow trading mode for a specific security, all markets trading the same security should do the same.

More specifically, information produced by the primary listing market is considered as *related to the purpose of use* for CB information of any market that trade the same security, which result in considering the CB information of the primary listing market (CME) as a sub part of the CB information of other markets (e.g., NYSE, Nasdaq). Similarly, the main listing market should be aware of the different activities performed by the markets that trade the same securities. Thus, NYSE and Nasdaq CB information is considered as sub parts of CB information that is used by the primary market (CME). Such relation is shown in Figure 2 as edges labeled with *part of* between "CME CB info" and both its sub-items "NYSE CB info" and "Nasdaq CB info".

Information timeliness: we need to provide the required concepts that enable for deciding whether information is valid in terms of time for its purpose of usage. Since we already defined two different relations between goals and information that can be affected by time aspects (e.g., reads and sends), we need to define validity that fits the needs of each of these relations:

Read timeliness : in order to ensure that information is valid for read, we need to ensure that its value in the system represents its value in the real world. Lack of timeliness leads to situations where the value of information in the system does not accurately reflects its value in the real world [43]. We rely on Ballou et al. [3] work to analyze the timeliness of read information depending on its *currency (age)*: the time interval between information creation (or update) to its usage

time [46, 31]) and its *volatility*: the change rate of information value [46], i.e., information is not valid, if its currency (age) is bigger than its volatility interval, otherwise it is valid.

Send timeliness : is used to capture the validity of information at its destination in terms of time. In particular, it defines the allowed amount of time for information to reach its destination, which should be defined based on the needs of information sender.

Referring to Figure 2, the achievement of the goal "Perform after trade operations" is subject to the validity of "Trade info" at its destination [bank], if information was not valid (delivered within the defined send [time]), the goal will not be achieved. While the achievement of the *investor's* goal "Analyze the market depending on trader" depends on the validity of "Tr trading suggestions" that is provided by the *trader*, in order for such information to be valid, it should be provided within a time interval that is less than its volatility change rate.

Information consistency: we need to provide the required concepts that enable for deciding whether information is consistent or not. Information consistency arises only when there are multiple records of the same information that are being used by several actors for *interdependent purposes (goals)*, and we call such actors as *interdependent readers*. While if actors use the same information for independent purposes, inconsistency will not be an issue since the actors' activities are independent. For example, CBs information should be consistent among all markets trade the same securities, since they depend on such information for controlling their trading environment (*interdependent purposes*). While the same information can be used by a *trader* for analyzing the market and make trading decision, yet inconsistency between information a *trader* use and the ones used by markets will not produce any problem, since such information is used for independent purposes.

Moreover, consistency in our work is a time related aspect [5], i.e., the value of information among its different *interdependent readers* might became inconsistent due to time related aspects. In particular, to ensure consistency among the different *interdependent readers*, we need to ensure that these readers depend on the same information value in term of time. Thus, we define *read-time* that indicates the actual read time by information *reader*,

[5]In [46] consistency was used to refer to "representational consistency" of information

and by ensuring that all *interdependent readers* have the same *read-time*, we can ensure the consistency of such information. Considering our example, to ensure the consistency of "CME CB info" among all markets that trade the same security (*interdependent readers*), all of these markets (e.g., NYSE, Nasdaq) should have the same *read-time*, i.e., such information should be provided to them in a way that ensure all of them have the same *read-time*.

Actor's social interactions and IQ: actors' interactions might affect IQ. Thus, we need to provide the required concepts to capture how such interactions might affect IQ in terms of its different dimensions. To get better understanding of actors interactions and IQ, we depend on what is called *information provenances* [38], which enable us to capture any information that helps in determining the history of information, starting from its source and the process by which it has been delivered to its destination [35]. In particular, information accuracy can be influenced by the trustworthiness of information production along with its provision process (discussed earlier). At the other hand, information validity can also be affected by actors' interactions. More specifically, *information provision time* [6] might influence information read and send timeliness, or even information consistency, if there are *interdependent readers* of the provided information.

All new concepts along with the basic constructs of secure Tropos modeling language are structured in terms of a meta-model shown in Figure 3, where we identify: an actor that covers two concepts (role and agent) and it may have a set of goals, it aims for. Further, an actor may have the related capabilities for the achievement of goals. Actors can be interdependent readers concerning an information item. Moreover, actors may delegate goals to one another, and they may have information, and provides it to one another, where provision has a provision time. Goals can be and / or-decomposed, and they may produce, read, or send information; yet read can be descried by its type (e.g., optional or required), and its purpose of use that is used to address both information competence and consistency in the case of interdependent readers. While send can be described by its time attribute. Information has volatility rate that is used to determine its validity. Further, information can be composed of several information items (*part of*). Finally, actors may trust one another for goal achievement / information provision.

Finally, in order to allow for the systematic design of the system-to-be, we

[6]The amount of time information transmission requires from source to destination (referred to as the transmission time in networks [14])

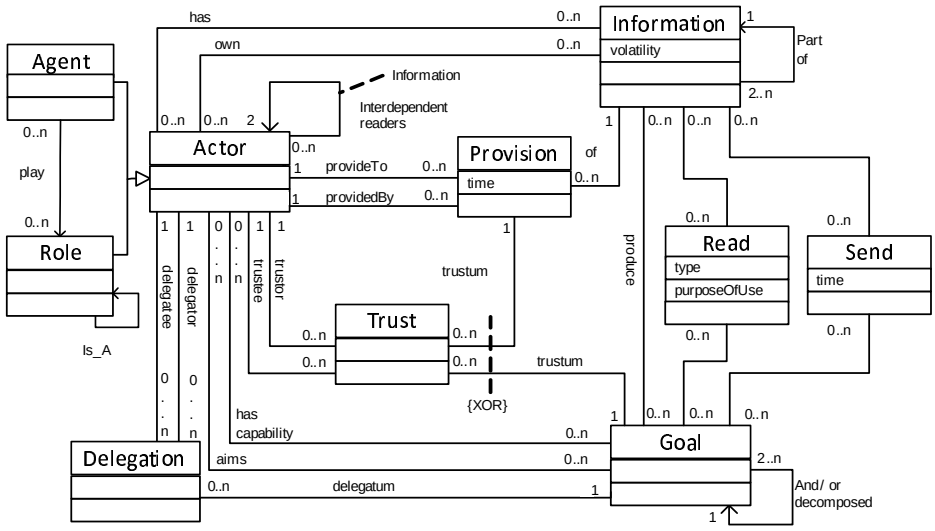

Figure 3: Meta-model shows the extended version of secure Tropos

propose an engineering methodology that underlies our extended framework. The process consists of several steps that should be followed by designers during the system design; each of these steps is described as follows: (1) *Actors modeling*: in which the stockholders of the system are identified and modeled along with their objectives, entitlements and capabilities; (2) *Goals modeling*: the stockholders' goals are identified and refined through And/ Or-decomposition, and based on the actors capabilities some goals might be delegated; (3) *Goals-information relations*: the different relations among goals and information are identified and modeled along with their IQ needs; (4) *Information modeling*: information is modeled, the structure of composed information is identified, and then information provisions are modeled; (5) *Trust modeling*: trust among actors concerning goal delegation, information producing and provisions are modeled; (6) *Analyzing the model*: at this step the model is analyzed to verify whether all the stakeholders' requirements are achieved or not; (7) *Refining the model*: during the model analysis, if some of the stockholders' requirements were not achieved, the analysis try to find solution for such issues at this step.

Table 1: General predicate

Type Predicates	
actor(Actor:a) role(Role:r) info(Info:i, Volatility:v)	agent(Agent:x) goal(Goal:g)
Actor's Relations	
is_a(Role:r_1, Role:r_2) conflicting_roles(Role:r_1, Role:r_2)	plays(Agent:a, Role:r)
Actor's Properties	
aims(Actor:a, Goal:g) producer(Actor:a, Info:i, Time:t) sender(Time:t, Actor:a, Actor:b, Info:i) has(Actor:a, Info:i, Time:t) can_achieve(Actor:a, Goal:g) achieved(Actor:a, Goal:g) fits_reader(Actor:a, Info:i)	objective(Actor:a, Goal:g) reader(Type:t, Purpose:pou, Actor:a, Info:i) is_responsible(Actor:a, Goal:g) can_provide(Actor:a, Info:i) achieve(Actor:a, Goal:g) not_achieved(Actor:a, Goal:g)
Goals' Properties	
produces(Goal:g, Info:i, Time:t) send(Time:t, Goal:g, Actor:a, Info:i) read_dependent(Goal:g, Info:i) fits_send(Time:t, Goal:g, Actor:a, Info:i) prevented(Goal:g) read_prevented(Goal:g, Info:i)	read(Type:t, Purpose:pou, Goal:g, Info:i) dependent(Goal:g) send_dependent(Goal:g, Info:i) fits_read(Type:t, Purpose:p, Goal:g, Info:i) send_prevented(Goal:g, Info:i)
Goal Analysis	
andDecomposition(Goal:g, Goal:g_1)	orDecomposition(Goal:g, Goal:g_1)
Information Quality Analysis	
accurate_read(Actor:a, Info:i)	inaccurate(Actor:a, Info:i)
complete_read(Actor:a, Info:i) partOf(Info:i, Info:i_1) composedOfOne(Info: i) numOfParts(Info: i, Number: n) relatedToPurpose(PoU, I1)	incomplete_read(Actor:a, Info:i) composed(Info:i) composedOfTwo(Info: i) used_for(I, PoU)
valid_read(Actor:a, Info:i) read_time(Time:t, Actor:a, Info:i)	invalid_read(Actor:a, Info:i)
consistent_read(Actor:a, Info:i) numOfReaders(Info:i, Number:n) interdependent_readers(Actor:a, Actor:b, Info:i)	inconsistent_reader(Actor:a, Info:i) only_reader(Actor:a, Info:i)
Actors' Social Relations	
provide(Time:t , Actor:a, Actor:b, Info:i) delegate(Actor:a, Actor:b, Goal:g) trust(Actor:a, Actor:b, Type:t, Info:i)	prvChain(Time:t, Actor:a, Actor:b, Info:i) deleChain(Actor:a, Actor:b, Goal:g) trustChain(Actor:a, Actor:b, Type:t, Info:i)

5 Reasoning about Information Quality requirements

We use Datalog [2] to formalize the concepts that have been introduced, along with the required axioms. Table 1 introduces the general predicates. While Table 2 lists the actors' objectives, entitlements and capabilities related axioms. For example, O1 states that if an actor aims for a goal, it became an objective for the actor. E1 states that an actor became responsible of a goal

Table 2: Actors Objectives, Entitlements and Capabilities Axioms

Actor's objectives	
O1	`objective(A, G) :- aims(A, G).`
O2	`objective(A, G) :- deleChain(B, A, G), objective(B, G).`
O3	`objective(A, G1) :- andDecomposition(G, G1), objective(A, G).`
O4	`objective(A, G1) :- orDecomposition(G, G1), objective(A, G).`
Actor's entitlements	
E1	`is_responsible(A, G) :- objective(A, G), can_achieve(A, G), not not_leaf(G).`
E2	`reader(Type, PoU, A, I) :- is_responsible(A, G), read(Type, PoU, G, I).`
E3	`sender(T, A, B, I) :- is_responsible(A, G), send(T, G, B, I).`
E4	`not_leaf(G) :- andDecomposition(G, G1).`
E5	`not_leaf(G) :- orDecomposition(G, G1).`
Actor's capabilities	
C1	`producer(A, I, T) :- achieve(A, G), produces(G, I, T).`
C2	`has(A, I, T) :- producer(A, I, T).`
C3	`can_provide(A, I) :- has(A, I, T).`
C4	`has(A, I, T) :- prvChain(T, B, A, I), can_provide(B, I).`
C5	`can_achieve(A, G) :- play(A, R), can_achieve(R, G).`
C6	`can_achieve(A, G) :- deleChain(A, B, G), can_achieve(B, G).`
C7	`can_achieve(A, G) :- orDecomposition(G, G1), can_achieve(A, G1).`
C8	`can_achieve(A, G) :- andDecomposition(G, G1), andDecomposition(G, G2), can_achieve(A, G1), can_achieve(A, G2), G1 != G2.`

achievement, if the goal is an objective of the actor and the actor has the capabilities to achieve it. While C1 states that an actor became information producer, if the actor achieves the goal that is responsible of information production.

Table 3 lists IQ related axioms. For example, IQ1 states information fits for send for its sender, if a valid provision chain holds between its sender and its destination with a provision time less than the time required by its sender. IQ3 states that information fits for read from the perspective of its reader if it was accurate, complete, valid and consistent. While IQ4-26 provide the axioms required to analyze information accuracy, completeness, validity and consistency.

Table 4 lists axioms concerning the different relations among goals/ information. For example, G1-2 state that a goal is dependent if it was information read dependent or information send dependent. In table 5 lists the actors social relations concerning information provision/ producing (S1, S2), goals delegation (S3, S4), along with trust relations among the actors (S5, S6). Finally, table 6 lists axioms used to identify whether a goal is achieved or not from the perspective of the actor, who aims for it. For example, A1 states

Table 3: Information Quality Axioms

IQ1	fits_send(T,G,B,I):- is_responsible(A,G), prvChain(Tr,A,B,I), trustChain(A,B,provide,I), #int(T), #int(Tr), Tr<T .
IQ2	fits_read(T,PoU,G,I):- is_responsible(A,G), reader(T,PoU,A,I), fits_reader(A,I).
IQ3	fits_reader(A,I):- accurate_read(A,I), complete_read(A,I), valid_read(A,I), consistent_read(A,I).
IQ4	accurate_read(A,I):- reader(T,PoU,A,I), not inaccurate(A,I).
IQ5	inaccurate(A,I):- reader(T,PoU,A,I), producer(B,I,T), not trust(A,B,produce,I).
IQ6	inaccurate(A,I):- reader(T,PoU,A,I), producer(B,I,T), not trustChain(A,B,provide,I).
IQ7	complete_read(A,I):- reader(T,PoU,A,I), has(A,I,_), not composed(I).
IQ8	complete_read(A,I):- reader(T,PoU,A,I), has(A,I,_), composedOfOne(I), partOf(I,I1), has(A,I1,_).
IQ9	complete_read(A,I):- reader(T,PoU,A,I), has(A,I,_), composedOfTwo(I), partOf(I,I1), partOf(I,I2),has(A,I1,_), has(A,I2,_), I1 != I2.
IQ10	composedOfOne(I):- numOfParts(I,1).
IQ11	composedOfTwo(I):- numOfParts(I,2).
IQ12	composed(I):- composedOfOne(I).
IQ13	composed(I):- composedOfTwo(I).
IQ14	numOfParts(I, X):- partOf(I,I1),#countZ:partOf(I,Z)=X.
IQ15	used_for(I, PoU):- read(T, PoU, A, I).
IQ16	partOf(I, I1):- used_for(I, PoU), relatedToPurpose(PoU, I1).
IQ17	incomplete_read(A,I):- reader(T,PoU,A,I), not complete_read(A,I).
IQ18	valid_read(A,I):- reader(T,PoU,A,I), read_time(T,A,I), info(I,V), #int(T), #int(V), V >T.
IQ19	invalid_read(A,I):- reader(T,PoU,A,I), not valid_read(A,I).
IQ20	consistent_read(A,I):- only_reader(A,I).
IQ21	only_reader(A,I):- reader(_,_,A,I), numOfReaders(X,I), X=1.
IQ22	numOfReaders(X,I):- reader(_,_,A,I), #countZ:reader(_,_,Z,I) =X.
IQ23	consistent_read(A,I):- reader(_,_,A,I), not only_reader(A,I) ,not inconsistent_reader(A,I).
IQ24	inconsistent_reader(A,I):- interdependent_readers(A,B,I), read_time(X,A,I), read_time(Y,B,I),#int(X), #int(Y), X != Y, A!=B.
IQ25	read_time(T,A,I):- reader(_,_,A,I), has(A,I,T).
IQ26	interdependent_readers(A,B,I):- reader(_,PoU,A,I), reader(_,PoU,B,I), A!=B.

that a goal is achieved for an actor, if the goal is not information dependent and the actor took the responsibility of achieving it by itself.

Further, we define a set of properties (shown in Table 7) that are used

Table 4: Goal-Information Axioms

G1	dependent(G):- read_dependent(G,I).
G2	dependent(G):- send_dependent(G,I).
G3	read_dependent(G,I):- read(_,PoU,G,I).
G4	send_dependent(G,I):- send(T,G,B,I).
G5	prevented(G):- read_prevented(G,I).
G6	prevented(G):- send_prevented(G,I).
G7	send_prevented(G,I):- send(T,G,B,I), not fits_send(T,G,B,I).
G8	read_prevented(G,I):- read(r,PoU,G,I), not fits_read(r,PoU,G,I).

Table 5: Social Relations Axioms

```
S1  prvChain(T, A, B, I) :- provide(T, A, B, I).
S2  prvChain(Z, A, C, I) :- provide(X, A, B, I), prvChain(Y, B, C, I),
    #int(X),#int(Y), #int(Z), Z = X + Y.
S3  deleChain(A, B, G) :- delegate(A, B, G).
S4  deleChain(A, C, G) :- delegate(A, B, G), deleChain(B, C, G).
S5  trustChain(A, B, O, S) :- trust(A, B, O, S).
S6  trustChain (A, C, O, S) :- trust(A, B, O, S), trustChain(B, C, O, S).
```

Table 6: Goals Achievement Axioms

```
A1  achieve(A, G) :- is_responsible(A, G), not dependent(G).
A2  achieve(A, G) :- is_responsible(A, G), dependent(G), not prevented(G).
A3  achieved(A, G) :- achieve(A, G).
A4  achieved(A, G) :- deleChain(A, B, G), trustChain(A, B, achieve, G), achieve(B, G).
A5  achieved(A, G) :- andDecomposition(G, G1), andDecomposition(G, G2),achieved(A, G1),
    achieved(A, G2), G1 != G2 .
A6  achieved(A, G) :- orDecomposition(G, G1), achieved(A, G1).
A7  not_achieved(A, G) :- aims(A, G), not achieved(A, G).
```

to verify the correctness and consistency of the requirements model. These properties define constraints that the designers should consider during the system design.

Pro1 states that the model should not include any goal that is not *achieved* from the perspective of the actor, who has it within its objectives. Goal might not be achieved due to several reasons (e.g., delegating the goal with no trust chain, missing required information, IQ related issues, etc.). For example, in Figure 2 *Sarah* delegates the goal "making profit by trading securities" with no trust chain to *Small tradCo 1*. This leaves *Sarah* with no guarantee that its goal will be achieved.

Pro2-3 state that the model should not include any information unavailability related issues, i.e., senders / required readers should have the information they intend to send/ read. Note that capturing information availability is not a trivial task. For example, in Figure 2 if the goal "Perform the trades" was not achieved, information "Trade info" will not be produced, and both goals "Perform after trades operations" and "Analyzing the trading environment" will not be achieved as well, since both of them require to read "Trade info". Similarly, the effect of not achieving these goals might be propagated to other goals.

Pro4-5 state that the model should not include any inaccurate information from the perspectives of their readers, i.e., there is no guarantee that

information is accurate for read, if it was not produced by a trusted source (**Pro4**), and provided by a trusted provision (**Pro5**). Intentionally falsified information (inaccurate from the reader's perspective) was a main reason that led to the Flash Crash. In particular, some HFTs were accused of providing orders that last very short time, which make them unavailable to most traders, in order to affect the prices of some securities before starting their real trades. Moreover, Market Makers and in order to fulfill their obligations concerning providing sell / buy orders in the market, provide what is called "stub quotes" (falsified information). During the Flash Crash, over 98% of all trades were executed at prices within 10% of their values before the crash because of "stub quotes" [20]. In particular, if orders that have been provided by both HFTs and Market Makers were not considered accurate for granted, such crash might be avoided.

Pro6 states that the model should not include information that is not complete from the perspective of its reader. For example, after considering "CME CB info" as a part of "NYSE CBs information", Pro6 is able to detect and notify the designer, if *NYSE* does not has "CME CB info".

Pro7 states that the model should not include any invalid information from the perspective of their readers. For example, a *Small Tradco 1* provides *John* with "Tr trading suggestions". Yet, the delivery time should not exceed the information volatility rate to be considered as valid. Otherwise, *John* may make wrong trading decisions based on invalid (old) information.

Pro8 states that the model should not include any *interdependent reader* that depend on inconsistent information. Considering our example, *NYSE* and *Nasdaq* are *interdependent readers* concerning "CME CB info". Pro8 is able to detect and notify the designer, if "CME CB info" is not consistent between both of them, i.e., they do not have the same *read-time*.

Pro9 states that the model should not include inaccurate information at their destination from the perspective of their senders, i.e., a trusted provision chain should hold between the sender and its intended destination. While **Pro10** states that the model should not include invalid information at their destination from the perspective of their senders. For example, *stock traders* (e.g., *Small TradCo 1*) have different quality of services, including the time that orders require to reach the market (milliseconds might be very important). If a *Small TradCo 1* is not able to provide the time to market that *John* requires, his orders will not be considered as valid from his perspectives.

Pro11 states that the model should not include any agent that plays

Table 7: Properties of the design

Pro1	:- objective(A,G), not achieved(A,G)
Pro2	:- sender(T,A,B,I), not has(A,I,Z)
Pro3	:- reader(required,P,A,I), not has(A,I,Z)
Pro4	:- reader(T,P,A,I), producer(B,I), prvChain(T,B,A,I), not trust(A,B,produce,I)
Pro5	:- reader(T,P,A,I), producer(B,I), prvChain(T,B,A,I), not trustChain(B,A,provide,I)
Pro6	:- reader(T,P,A,I), not complete(A, I)
Pro7	:- reader(T,P,A,I), prvChain(T,B,A,I), producer(B,I), info(I,V), not T<V
Pro8	:- reader(T,P,A,I), interdependent_reader(A,I), not consistent(A,I)
Pro9	:- sender(T,A,B,I), prvChain(T,A,B,I), not trustChain(A,B,provide,I)
Pro10	:- sender(T,A,B,I), prvChain(Tr,A,B,I), not Tr<T
Pro11	:- plays(A,R1), plays(A,R2), conflicting_roles(R1,R2)

conflicting roles. In particular, it is used to ensure that the model manage separation of duties among its actors to avoid any conflict of interest that leaves the system open to various kinds of vulnerability.

In Figure 2, we can see that *Star Co* is playing both roles "Credit assessment firm" and "Consulting firm". Such situation should be avoided, since we cannot trust a company for providing accurate consulting information considering the securities of a company that they get paid to perform their credit assessment. Pro11 can be used to capture similar situations, such as firms that provide accounting services along with auditing services to the same company (e.g., The Enron scandal [30]).

6 Implementation and evaluation

Evaluation is an important aspect of any research proposal; it aims to demonstrate the utility, quality, and efficacy of a design artifact. Our framework belongs to the design science area. Hevner et al. [17] classify evaluation methods in design science under five categories: observational, analytical, experimental, testing, and descriptive. We aim to evaluate the applicability and effectiveness of our framework depending on simulation method (experimental), i.e., execute artifact with artificial data. To this end, we developed a prototype implementation of our framework[7] (Figure 4) to test its applicability and effectiveness for modeling and reasoning about IQ requirements. In what follows, we briefly describe the prototype, discuss its applicability and effectiveness over the Flash Crash scenario, and then test the scalability of its reasoning support.

[7]http://mohamadgharib.wordpress.com/

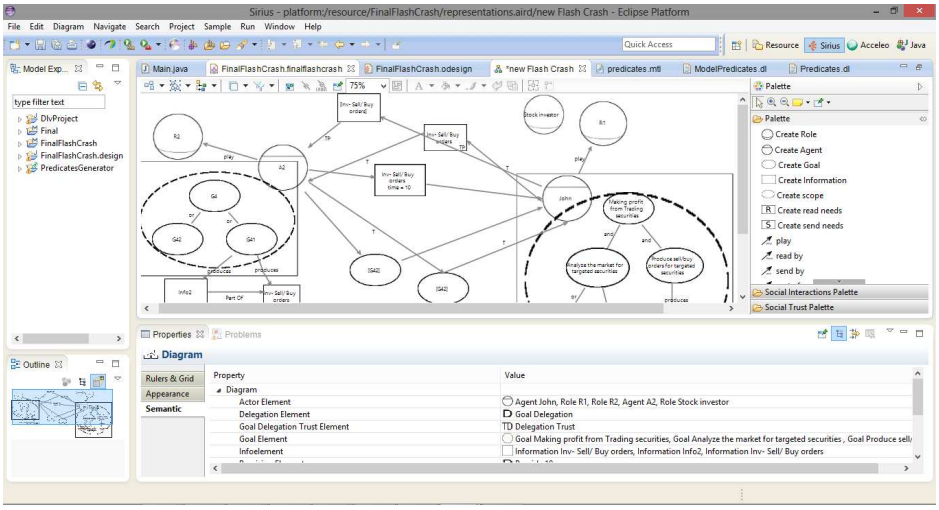

Figure 4: Screenshot of the Eclipse-based tool

Implementation: our prototype consist of 3 main parts: (1) a graphical user interface (GUI) developed using Sirius[8], which enable designers for drawing the model diagram by drag-and-drop modeling elements from palettes, and enables for specifying the properties of these elements along with their interrelations; (2) model-to-text transformation that supports the translating of the graphical models into Datalog formal specifications depending on Acceleo[9]; (3) automated reasoning support (DLV system[10]) takes the Datalog specification that resulted from translating the graphical model along with the reasoning axioms, and then verifies the correctness and completeness of the requirements model against the properties of the design.

Applicability and effectiveness: We evaluate our framework by showing its applicability in capturing the IQ requirements along with its effectiveness in capturing the violation of the properties of the design by applying it to the Flash Crash motivating example. We used our extended modeling language to model the Flash Crash motivating example, and then we translate the requirements diagram into Datalog formal language. Finally, we depend on the reasoning support technique that our framework provides to check whether the requirements model is correct and consistent, i.e., all the prop-

[8]https://projects.eclipse.org/projects/modeling.sirius
[9]https://projects.eclipse.org/projects/modeling.m2t.acceleo
[10]http://www.dlvsystem.com/dlv/

erties of the design should hold. The analysis captured several violations of the properties of the design including:

Inaccurate information : *CME* market considers information received from both *Market Marker 1* and *HFT trades Co* as inaccurate information, since no trust in information production holds between them at one hand, and *CME* at the other. Note that how such situation should be handled is left to the stakeholders decision. For example, *CME* may apply some mechanism to verify the accuracy of information received from *Market Marker 1*, i.e., it can define a min/max order value for accepting the orders received from textitMarket Marker 1, in order to reject "stub quotes". Other mechanisms can be applied for orders received from *HFT trades Co*, in order to reject their flickering quotes.

Inaccurate information due to playing conflicting roles : *Star Co* is playing both "Credit assessment firm" and "Consulting firm" roles. However, we cannot trust a company for providing accurate consulting information considering the securities of a company that they get paid to perform their credit assessment. Thus, information produced by *Star Co* is considered as inaccurate.

Incomplete information : "CME CB info", " NYSE CB info" and "Nasdaq CB info" are identified as incomplete information from the perspectives of their readers, since they miss some sub parts related to the purpose they are used for (goal they aims to achieve). However, this can be solved by providing these markets with the missed information sub items, i.e., providing *CME* with both "NYSE CB info" and "Nasdaq CB info", and providing both NYSE and Nasdaq with "CME CB info".

Inconsistent information : both of *NYSE* and *Nasdaq* are *interdependent readers* concerning "CME CB info". However, "CME CB info" is provided to them with different provision times. According to [21], provision time from *CME* to *Nasdaq* was 13 (ms), while provision time from *CME* to *NYSE* was 14.65 (ms), which leads to different *read-times* between these two markets, and resulted in inconsistency between them.

Experiments on scalability: to test the scalability of the reasoning technique, we expanded the model shown in Figure 2 by increasing the num-

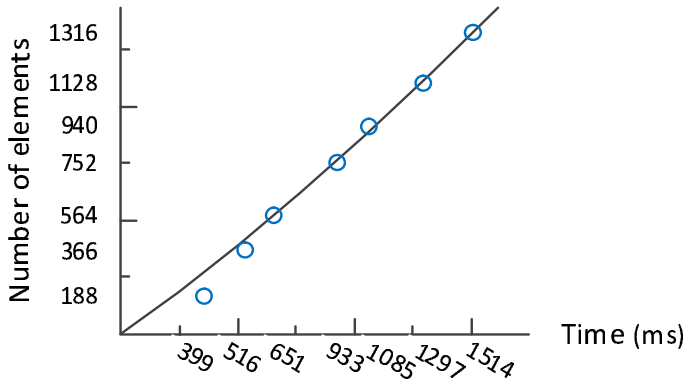

Figure 5: Scalability results with increasing the number of modeling elements

ber of its modeling elements from 188 to 1316 through 7 steps, and investigate the reasoning execution time at each step by repeating the reasoning execution 7 times, discarding the fastest and slowest ones, and then computed the average execution time of the rest. The result is shown in Figure 5, and it is easy to note 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, 4GB RAM, OS Window 8, 64-bit.

Note that the model can be extremely big. According to [29], the numbers of market participation for the E-mini securities at the Flash Crash day were: 1268 fundamental buyers; 1276 fundamental sellers; 176 market makers; 16 HFTs; 6880 small traders; 5808 opportunistic traders, and the number of investor is much greater than that.

7 Related Work

A large body of literature has focused on IQ. For instance, Wand and Wang [43] propose a theoretical approach to define information quality. While Wang and Strong [45] propose the Total Data Quality Management (TDQM) methodology, with a main purpose of delivering high quality information products (IP) to information consumers. TDQM was build based on the same idea proposed in [44]. Ballou et al. [3] presented an information manufacturing system that can be used to determine the data quality in terms of timeliness, quality, cost, and value of information products. In their work

the IQ is a customer driven, i.e., the quality of the information products is determined by its customer.

Moreover, Shankaranarayanan et al. [37] extend the information manufacturing system model proposed by Ballou to develop a formal modeling method for creating an IP-MAP. While Scannapieco et al. [34] propose IP-UML approach that relies on the IP-MAP framework. IP-UML uses the Data Quality profile as the modeling language; which consists of three different models, namely: the Data Analysis Model, the Quality Analysis Model, and the Quality Design Model. However, all the previously mentioned approaches were not designed to capture neither the organizational nor the social aspects of the system-to-be, which are very important aspects in current complex systems.

At the other hand, requirements engineering community did not appropriately support modeling nor analyzing IQ requirements of the system-to-be. For example, Abuse cases [25] and misuse case [39] both did not provide primitives for modeling IQ needs. While UMLsec [19] propose concepts for modeling information integrity (IQ related aspect) as a constraint, which can restrict unwanted modifications of information, yet IQ still can be compromised in several other ways. SecureUML [4] does not consider IQ at all, since it was mainly developed to model access control policies. Abuse frame [22] addresses integrity related issues (modification) by preventing unauthorized actors from modifying information, or prevent authorized actors from doing unauthorized modifications. Finally, secure Tropos [28] seems to be sufficient to capture the functional, privacy and trust requirements of system-to-be in its organizational context, yet it provides no primitives for capturing IQ needs. In particular, secure Tropos cares about information in terms of its availability and who is responsible of providing it.

8 Conclusions and Future Work

In this paper, we highlighted the importance of capturing IQ needs of the system-to-be starting from design phase. We argued that IQ is not only a technical issue, but it is also an organizational and social issue. Thus, any solution for IQ should be considered at the organizational level. Moreover, we showed how IQ can be analyzed through its different dimensions. At the other hand, we discussed the required extensions for secure Tropos for capturing IQ requirements, and then we extended its conceptual model with

the required concepts and constructs for modeling and analyzing IQ requirements. In particular, our framework enables system designers to capture IQ requirements in terms of its different dimensions; taking into consideration the information intended purpose of usage. Further, it provide the required analysis techniques to verify whether the stakeholders' requirements are met or not, and to detect any conflict among the IQ requirements of the stakeholders of the system.

For the future work, we intend to extend the IQ dimensions we considered. Further, the different interrelations among IQ dimensions need to be studied in more details. Furthermore, information production process needs more investigation, since information might be produced depending on other information item(s), and in such cases, the quality of the produced information is highly influenced by the quality of the information item(s) that has/have been used in the production process.

References

[1] Nanex flash crash summary report. `http://www.nanex.net/FlashCrashFinal/FlashCrashSummary.html`, 2010. Accessed: 2014-05-30.

[2] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of databases*. Citeseer, 1995.

[3] Donald Ballou, Richard Wang, Harold Pazer, and Giri Kumar Tayi. Modeling information manufacturing systems to determine information product quality. *Management Science*, 44(4):462–484, 1998.

[4] D. Basin, J. Doser, and T. Lodderstedt. Model driven security: From uml models to access control infrastructures. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 15(1):39–91, 2006.

[5] M. Bovee, R.P. Srivastava, and B. Mak. A conceptual framework and belief-function approach to assessing overall information quality. *International journal of intelligent systems*, 18(1):51–74, 2003.

[6] Matthew Bovee, Rajendra P Srivastava, and Brenda Mak. A conceptual framework and belief-function approach to assessing overall information quality. *International journal of intelligent systems*, 18(1):51–74, 2003.

[7] Graham Bowley. Lone $4.1 billion sale led to flash crashin may. *The New York Times*, 1, 2010.

[8] Dave Cliff. The flash crash of may 6th 2010: Wtf. Technical report, mimeo, University of Bristol, 2011.

[9] A. Dardenne, A. Van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Science of computer programming*, 20(1-2):3–50, 1993.

[10] A.F. Dragoni. A model for belief revision in a multi-agent environment. *Decentralized AI*, 3:103–112, 1992.

[11] David Easley, MM Lopez De Prado, and Maureen OHara. The microstructure of the flash crash: Flow toxicity, liquidity crashes and the probability of informed trading. *Journal of Portfolio Management*, 37(2):118–128, 2011.

[12] FE Emery and EL Trist. Socio-technical systems. management sciences, models and techniques. churchman cw et al, 1960.

[13] Mark Flood, HV Jagadish, Albert Kyle, Frank Olken, and Louiqa Raschid. Using data for systemic financial risk management. In *CIDR*, pages 144–147, 2011.

[14] A Behrouz Forouzan. *Data Communications & Networking*. Tata McGraw-Hill Education, 2006.

[15] Christopher Fox, Anany Levitin, and Thomas Redman. The notion of data and its quality dimensions. *Information processing & management*, 30(1):9–19, 1994.

[16] Peter Gomber, Martin Haferkorn, Marco Lutat, and Kai Zimmermann. The effect of single-stock circuit breakers on the quality of fragmented markets. In *FinanceCom*, pages 71–87, 2012.

[17] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS quarterly*, 28(1):75–105, 2004.

[18] JM Juran, FM Gryna, and RS Bingham. *Quality control handbook*. New York [etc.]: McGraw-Hill, 1979.

[19] Jan Jürjens. *Secure systems development with UML*. Springer-Verlag New York Incorporated, 2005.

[20] Andrei Kirilenko, Albert S Kyle, Mehrdad Samadi, and Tugkan Tuzun. The flash crash: The impact of high frequency trading on an electronic market. *Manuscript, U of Maryland*, 2011.

[21] Michael Lewis. *Flash boys: a Wall Street revolt*. WW Norton & Company, 2014.

[22] L. Lin, B. Nuseibeh, D. Ince, M. Jackson, and J. Moffett. Introducing abuse frames for analysing security requirements. 2003.

[23] Liping Liu and Lauren Chi. Evolutional data quality: A theory-specific view. In *IQ*, pages 292–304, 2002.

[24] Ananth Madhavan. Exchange-traded funds, market structure and the flash crash. *SSRN Electronic Journal*, pages 1–33, 2011.

[25] J. McDermott and C. Fox. Using abuse case models for security requirements analysis. In *Computer Security Applications Conference, 1999.(ACSAC'99) Proceedings. 15th Annual*, pages 55–64. IEEE, 1999.

[26] Thomas McInish and James Upson. Strategic liquidity supply in a market with fast and slow traders. *Available at SSRN 1924991*, 2012.

[27] A. Motro. Integrity= validity+ completeness. *ACM Transactions on Database Systems (TODS)*, 14(4):480–502, 1989.

[28] H. Mouratidis and P. Giorgini. Secure tropos: A security-oriented extension of the tropos methodology. *International Journal of Software Engineering and Knowledge Engineering*, 17(2):285–309, 2007.

[29] Mark Paddrik, Roy Hayes, Andrew Todd, Steve Yang, Peter Beling, and William Scherer. An agent based model of the e-mini s&p 500 applied to flash crash analysis. In *Computational Intelligence for Financial Engineering & Economics (CIFEr), 2012 IEEE Conference on*, pages 1–8. IEEE, 2012.

[30] Joseph A Petrick and Robert F Scherer. The enron scandal and the neglect of management integrity capacity. *American Journal of Business*, 18(1):37–50, 2003.

[31] Leo L Pipino, Yang W Lee, and Richard Y Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, 2002.

[32] T.C. Redman. Improve data quality for competitive advantage. *Sloan Management Review*, 36:99–99, 1995.

[33] Carol A Reeves and David A Bednar. Defining quality: alternatives and implications. *Academy of management Review*, 19(3):419–445, 1994.

[34] M. Scannapieco, B. Pernici, and E. Pierce. Ip-uml: Towards a methodology for quality improvement based on the ip-map framework. In *7th Intl Conf. on Information Quality (ICIQ-02)*, pages 8–10, 2002.

[35] Laura Sebastian-Coleman. *Measuring Data Quality for Ongoing Improvement: A Data Quality Assessment Framework*. Newnes, 2012.

[36] Securities, Exchange Commission, et al. Findings regarding the market events of may 6, 2010. *Report of the Staffs of the CFTC and SEC to the Joint Advisory Committee on Emerging Regulatory Issues*, 2010.

[37] G. Shankaranarayanan, R.Y. Wang, and M. Ziad. Ip-map: Representing the manufacture of an information product. In *Proceedings of the 2000 Conference on Information Quality*, pages 1–16, 2000.

[38] Yogesh L Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance in e-science. *ACM Sigmod Record*, 34(3):31–36, 2005.

[39] G. Sindre and A.L. Opdahl. Eliciting security requirements by misuse cases. In *Technology of Object-Oriented Languages and Systems, 2000. TOOLS-Pacific 2000. Proceedings. 37th International Conference on*, pages 120–131. IEEE, 2000.

[40] Ian Sommerville, Dave Cliff, Radu Calinescu, Justin Keen, Tim Kelly, Marta Kwiatkowska, John Mcdermid, and Richard Paige. Large-scale complex it systems. *Communications of the ACM*, 55(7):71–77, 2012.

[41] Avanidhar Subrahmanyam. Algorithmic trading, the flash crash, and coordinated circuit breakers. *Borsa Istanbul Review*, 13(3):4–9, 2013.

[42] US Commodity Futures Trading. Preliminary findings regarding the market events of may 6, 2010. 2010.

[43] Y. Wand and R.Y. Wang. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11):86–95, 1996.

[44] Richard Y Wang, Veda C Storey, and Christopher P Firth. A framework for analysis of data quality research. *Knowledge and Data Engineering, IEEE Transactions on*, 7(4):623–640, 1995.

[45] R.Y. Wang. A product perspective on total data quality management. *Communications of the ACM*, 41(2):58–65, 1998.

[46] R.Y. Wang and D.M. Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, pages 5–33, 1996.

[47] Eric Siu-Kwong Yu. *Modelling strategic relationships for process reengineering*. PhD thesis, University of Toronto, 1995.