Modeling and Reasoning about Information Quality Requirements

Mohamad Gharib and Paolo Giorgini

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

Abstract. [**Context and motivation**] 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. [**Question/problem**] Despite this, most of the Requirements Engineering frameworks focus on "what" and "where" information is required, but not on the *intention* behind its use, which is essential to define the required level of quality that information should meets. [**Principal ideas/results**] In this paper, we propose a novel conceptual framework for modeling and reasoning about IQ at requirements level. [**Contribution**] 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

Information Quality, Requirements Engineering, Modeling, Reasoning

1 Introduction

Information Quality (IQ) is a key success factor for organizations, since depending on low-quality information may cause severe consequences [1], or even disasters in the case of critical systems. Despite its importance, IQ is often loosely defined, or simply ignored [2]. In general, quality has been defined as "fitness for use" [3], or as in [4] 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). 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 [5], 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 [6]. In particular, several reasons contributed to the Flash Crash were caused by socio-technical IQ related issues. For instance, according to [7] 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. 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. 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 [8], *i** [9]), we adopted secure Tropos [10] as a baseline for our framework. Secure Tropos introduces primitives for modeling actors of the system along with their goals that can be refined through And/ Or decompositions. Resources are used to represent both physical and informational entities that are needed/ produced for/by the achievement of goals[1].

Moreover, it provides the notion of delegation to model the transfer of responsibilities among actors, and it adopts the notion of trust and distrust to capture the expectations of actors in one another. 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 IQ 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).

---

[1] NeededBy/ producedBy have been proposed in SI* [11], which is an extension of secure Tropos

2 Motivating Example

Our motivating example concerns the May 6, 2010 U.S stock Flash Crash. Based on [7], 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 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 [7], *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 for a fee to *traders* and *investors*. 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., information is not falsified).

3 The Problem of Capturing Information Quality

The quality of information can be defined based on its "fitness for use", yet such definition does not explicitly capture the "fitness for use" for "what" and the

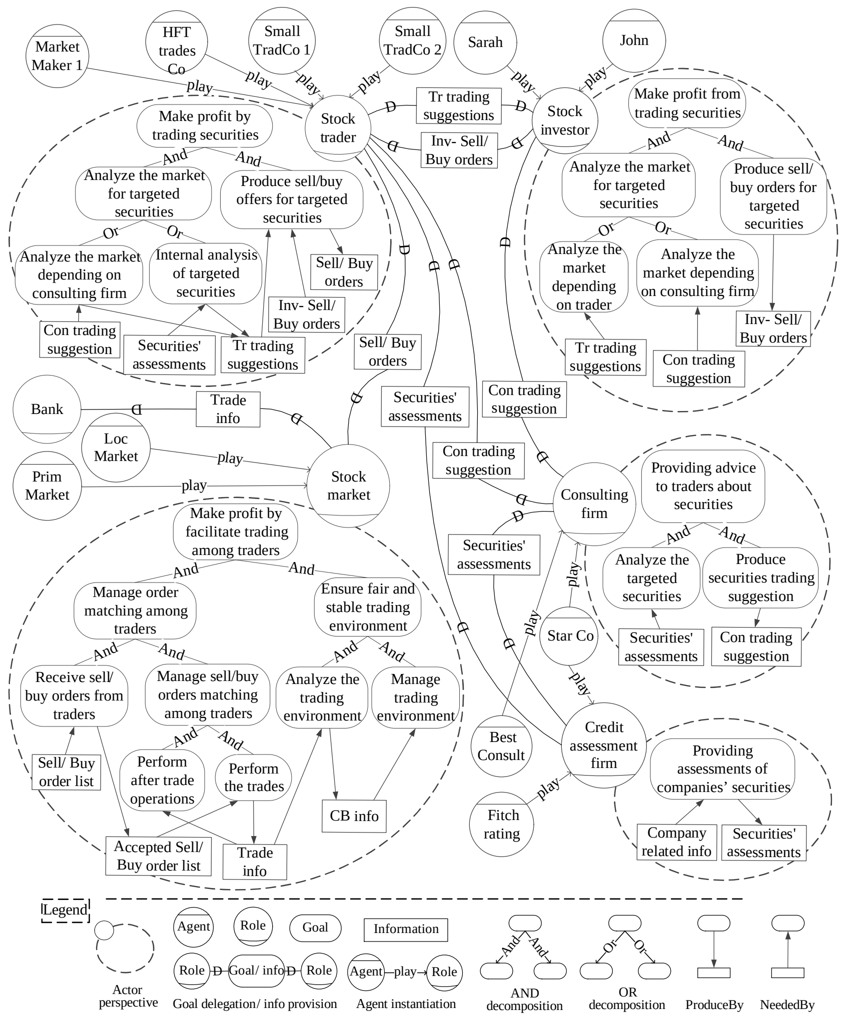

Fig. 1. A partial goal model concerning the U.S stock market structure

"fitness for use" of "who", which is very important when information has several stakeholders, who may require different (might be conflicting) quality needs. In other words, existing definitions of IQ miss the 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 [12,13], yet most of them propose holistic methods for analyzing IQ (one size fits all), i.e., they consider a user-centric view [14] 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 [15,16] 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 enable us to address the main IQ related problems that 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[16]; **Completeness**: means that all parts of information should be available [15,16]; **Timeliness**: means to which extent information is valid in term of time [13]; **Consistency**: means that multiple records of the same information should be the same across time [16].

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 "Prim (CB) information". 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 shown in Figure 2 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 [17,18], 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 [19]. 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 market Co1 (*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 or not. Thus, we rely on the *"part of"* concept that has been used in several areas (e.g., natural language, conceptual modeling, etc.) to model such relation. For example, one main reason of the Flash Crash was the effect of uncoordinated Circuit Breaker (CBs) [2] among the *markets.* Such failure resulted due to depending on incomplete information by *markets* for their CBs.

[2] Techniques used to slow or halt trading to prevent a potential market crash [20]

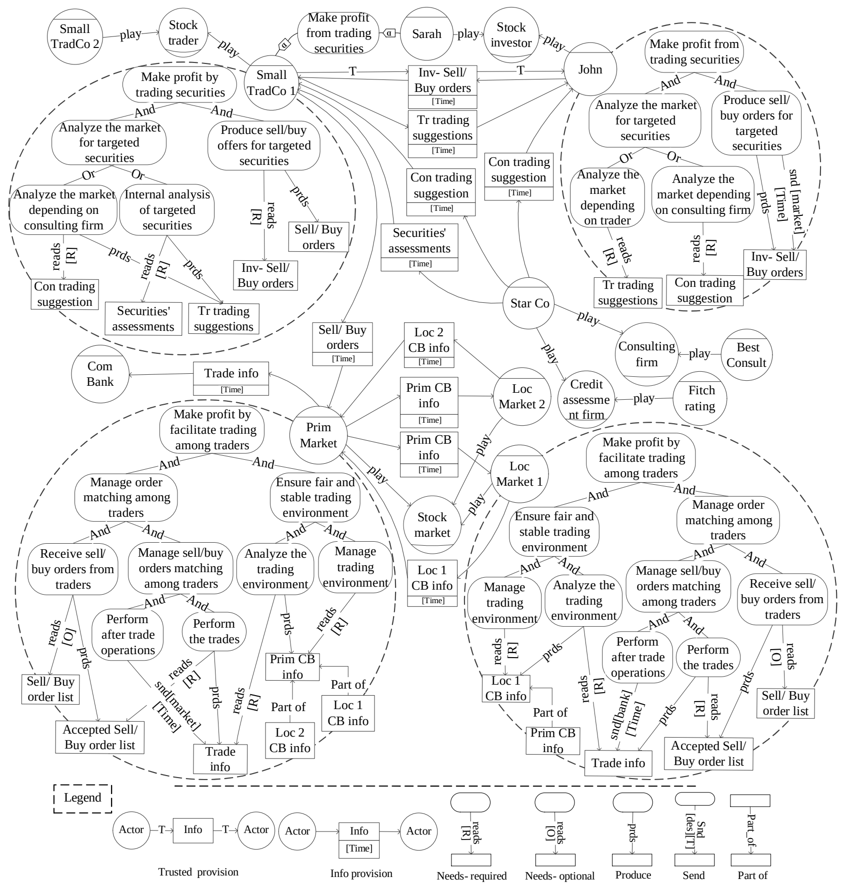

Fig. 2. A partial goal model of the Flash Crash extended with IQ related constructs

In particular, in stock market domain, the same securities might be traded in different markets. Thus, in order to coordinate the CBs between the different markets that trade the same security, markets should be aware of one another's 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. This can be solved, if we consider the CB information that is used by any market is composed of the local CB information along with the CB information produced by the primary listing market (the main market for trading the security) to guarantee that all markets who trade the same securities will coordinate properly. Similarly, the main listing market should be aware of the different activities performed by the markets that trade the same securities. Such relation is shown in Figure 2 as edges labeled with *part of* between "Prim CB info" and both its sub-items "Loc 1 CB info" and "Loc 2 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 [15]. We rely on Ballou et al. [17] 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 [14,13]) and its *volatility*: the change rate of information value [14], 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 [3], 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*, and by ensuring that all *interdependent readers* have the same *read-time*, we can ensure the consistency of

[3] In [14] consistency was used to refer to "representational consistency" of information

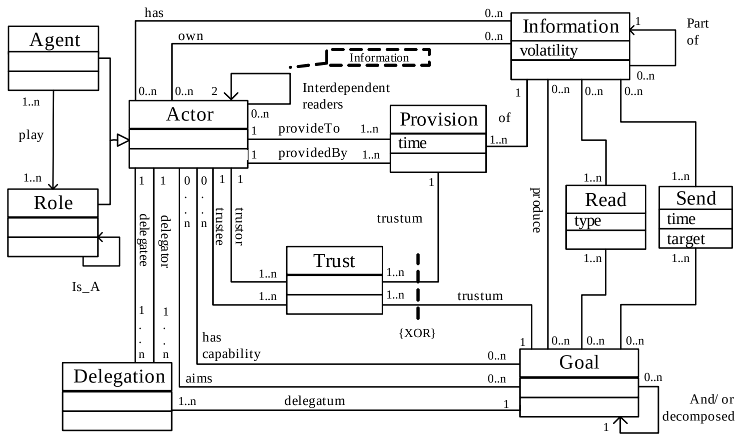

Fig. 3. Meta-model shows the extended version of secure Tropos

such information. Considering our example, to ensure the consistency of "Prime CB info" among all markets that trade the same security (*interdependent readers*), all of them 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* [21], 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 [22]. 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* [4] 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), while send can be described by its both time and target attributes.

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

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 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.

5 Reasoning about Information Quality requirements

We use Datalog [23] to formalize the concepts that have been introduced, along with the required axioms[5]. Further, we define a set of properties (shown in Table 1) that are used 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 tradCom 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.

[5] The formalization of the concepts and axioms is omitted due to space limitation, yet they can be found in [24]

Table 1. 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	:- play(A, $R1$), play(A, $R2$), conflicting_roles($R1$, $R2$)

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", which are orders with prices far away from the current market values. Such orders can also be considered as falsified information; since they are orders were not intended to be performed. 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" [7]. 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 "Prim CB info" as a part of "loc 1 CBs information", Pro6 is able to detect and notify the designer, if *Loc market 1* does not has "Prim CB info". While **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, *Loc Market 1* and *Loc Market 2* are *interdependent readers* concerning "Prim CB info". Pro8 is able to detect and notify the designer, if "Prim CB info" is not consistent between them.

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

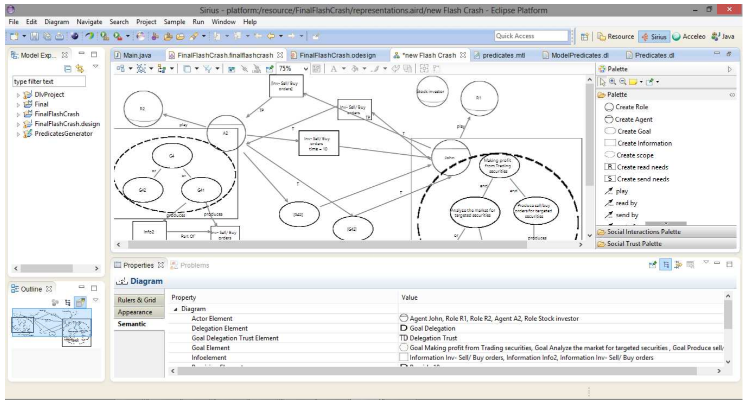

Fig. 4. Screenshot of the Eclipse-based tool

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 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 [25]).

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. [26] 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[6] (Figure 4) to test its applicability and effectiveness for modeling and reasoning about IQ requirements. In what follows, we briefly

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

describe the prototype, discuss its applicability and effectiveness over the Flash Crash scenario, and then test the scalability of its reasoning support.

Implementation: our prototype consist of 3 main parts: (1) a graphical user interface (GUI) developed using Sirius[7], 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[8]; (3) automated reasoning support (DLV system[9]) 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: is reported in [24], where the framework was applied to a big-size Flash Crash scenario. In particular, the Crash was not due to an attack or illegal activities, but some actors exploit undetected vulnerability in the system organizational structure, i.e., the design of the system allows for such failure. The framework was able to identify these vulnerabilities along with other vulnerability that manifested themselves in actors' interactions, or resulted from their conflict of interests. For example, a *stock 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 the *market* at the other. Moreover, information produced by *Star Co* is considered as inaccurate, since it plays two conflicting roles ("Credit assessment firm" and "Consulting firm"), i.e., 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.

At the other hand, "Prim CB info", "Loc 1 CBs info" and "Loc 2 CBs info" were identified as incomplete information from the perspectives of their readers, since they miss some sub parts related to the purpose of their use. Finally, it was able to detect the inconsistency concerning "Prim CB info" to both "Local market 1" and "Local market 1".

Experiments on scalability: to test the scalability of the reasoning technique, we expanded the model shown in Figure 2 by increasing the number 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. We have performed the experiment on laptop computer, Intel(R) core(TM) i3- 3227U CPU@ 190 GHz, 4GB RAM, OS Window 8, 64-bit. The result is shown in Figure 5, and it is easy to note that the relation between the size of the model (the number of its nodes) and the execution time is not exponential, i.e., the reasoning techniques should work fine with real world scenarios, where there sizes probably will not exceed the sizes we considered.

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

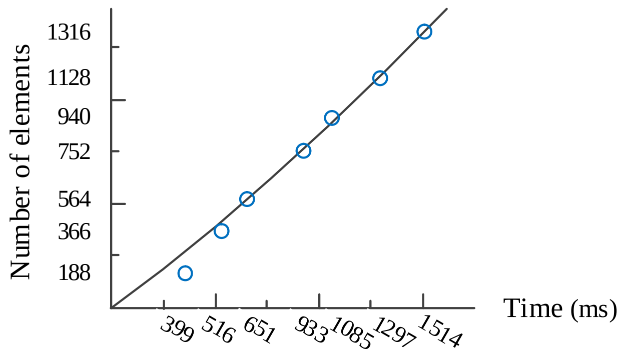

Fig. 5. Scalability results with increasing the number of modeling elements

7 Related Work

A large body of literature has focused on IQ. For instance, Wand and Wang [15] propose a theoretical approach to define information quality. While Wang and Strong [27] introduce the Total Data Quality Management (TDQM) methodology, with a main purpose of delivering high quality information products (IP) to information consumers. Ballou et al. [17] presented the Information Manufacturing System (IMS), which can be used to determine data quality in terms of timeliness, quality, etc. Moreover, Shankaranarayanan et al. [18] propose Information Product Map (IP-MAP) that extends IMS and offers a formal modeling method for creating Information Product (IP). Relying on the IP-MAP framework, Scannapieco et al. [28] introduce IP-UML approach that combines both data analysis and process analysis in order to assess the quality of data. 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, RE community did not appropriately support modeling nor analyzing IQ requirements (e.g., [8,9]). For example, abuse frame [29] addresses integrity (IQ related aspect) related issues (modification) by preventing unauthorized actors from modifying information, or prevent authorized actors from doing unauthorized modifications. While, UMLsec [30] proposes concepts for modeling information integrity as a constraint, which can restrict unwanted modifications of information, but IQ can still be compromised in several other ways. Finally, secure Tropos [10] / SI* [11] seem to be sufficient to capture the functional, privacy and trust requirements of system-to-be, yet they provide no primitives for explicitly capturing IQ requirements.

8 Conclusions and Future Work

In this paper, we highlighted the importance of capturing IQ needs from the early phase of system development. Moreover, we argued that IQ is not only a technical problem, but it is also an organizational and social issue, and we showed how IQ can be analyzed depending on its different dimensions. Furthermore, we proposed framework that enables system designers to capture IQ requirements in terms of their different dimensions; taking into consideration the intended purposes of information usage. Further, it provides the required analysis techniques to verify whether the stakeholders' IQ requirements are met or not, and it enables designers to refine the system design until such requirements are met.

For the future work, we intend to extend the considered IQ dimensions (e.g., trustworthiness, believability, etc.), and investigate in more details the different interrelations among them. Further, information production process needs more investigation, since information might be produced depending on other information item(s), and the quality of the produced information might be influenced by the quality of the information item(s) that has/have been used in the production process. Moreover, we aim to enrich the trust analysis that is used to assess information accuracy by relying on actors' internal structure (their intentions, desires, etc.), which allows to clearly identify "why" an actor should trust/ distrust another one for information accuracy. Finally, we plan to provide IQ policy specification language, which can be used to clearly identify the permitted, forbidden and obligated action to be carried out by the actors of the systems.

Acknowledgment

This research was partially supported by the ERC advanced grant 267856, "Lucretius: Foundations for Software Evolution", `http://www.lucretius.eu/`.

References

1. Redman, T.: Improve data quality for competitive advantage. Sloan Management Review **36** (1995) 99–99
2. Fox, C., Levitin, A., Redman, T.: The notion of data and its quality dimensions. Information processing & management **30**(1) (1994) 9–19
3. Juran, J., Gryna, F., Bingham, R.: Quality control handbook. New York [etc.]: McGraw-Hill (1979)
4. Reeves, C.A., Bednar, D.A.: Defining quality: alternatives and implications. Academy of management Review **19**(3) (1994) 419–445
5. Emery, F., Trist, E.: Socio-technical systems. management sciences, models and techniques. churchman cw et al (1960)
6. Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., Mcdermid, J., Paige, R.: Large-scale complex it systems. Communications of the ACM **55**(7) (2012) 71–77
7. Kirilenko, A., Kyle, A.S., Samadi, M., Tuzun, T.: The flash crash: The impact of high frequency trading on an electronic market. Manuscript, U of Maryland (2011)
8. Dardenne, A., Van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Science of computer programming **20**(1-2) (1993) 3–50
9. Yu, E.S.K.: Modelling strategic relationships for process reengineering. PhD thesis, University of Toronto (1995)

10. Mouratidis, H., Giorgini, P.: Secure tropos: A security-oriented extension of the tropos methodology. International Journal of Software Engineering and Knowledge Engineering **17**(2) (2007) 285–309
11. Zannone, N.: A requirements engineering methodology for trust, security, and privacy. PhD thesis, PhD thesis, University of Trento (2006)
12. Liu, L., Chi, L.: Evolutional data quality: A theory-specific view. In: IQ. (2002) 292–304
13. Pipino, L.L., Lee, Y.W., Wang, R.Y.: Data quality assessment. Communications of the ACM **45**(4) (2002) 211–218
14. Wang, R., Strong, D.: Beyond accuracy: What data quality means to data consumers. Journal of management information systems (1996) 5–33
15. Wand, Y., Wang, R.: Anchoring data quality dimensions in ontological foundations. Communications of the ACM **39**(11) (1996) 86–95
16. Bovee, M., Srivastava, R.P., Mak, B.: A conceptual framework and belief-function approach to assessing overall information quality. International journal of intelligent systems **18**(1) (2003) 51–74
17. Ballou, D., Wang, R., Pazer, H., Tayi, G.K.: Modeling information manufacturing systems to determine information product quality. Management Science **44**(4) (1998) 462–484
18. Shankaranarayanan, G., Wang, R., Ziad, M.: Ip-map: Representing the manufacture of an information product. In: Proceedings of the 2000 Conference on Information Quality. (2000) 1–16
19. Dragoni, A.: A model for belief revision in a multi-agent environment. Decentralized AI **3** (1992) 103–112
20. Gomber, P., Haferkorn, M., Lutat, M., Zimmermann, K.: The effect of single-stock circuit breakers on the quality of fragmented markets. In: FinanceCom. (2012) 71–87
21. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. ACM Sigmod Record **34**(3) (2005) 31–36
22. Sebastian-Coleman, L.: Measuring Data Quality for Ongoing Improvement: A Data Quality Assessment Framework. Newnes (2012)
23. Abiteboul, S., Hull, R., Vianu, V.: Foundations of databases. Citeseer (1995)
24. Gharib, M., Giorgini, P.: Detecting conflicts in information quality requirements: the may 6, 2010 flash crash. (2014)
25. Petrick, J.A., Scherer, R.F.: The enron scandal and the neglect of management integrity capacity. American Journal of Business **18**(1) (2003) 37–50
26. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS quarterly **28**(1) (2004) 75–105
27. Wang, R.: A product perspective on total data quality management. Communications of the ACM **41**(2) (1998) 58–65
28. Scannapieco, M., Pernici, B., Pierce, E.: Ip-uml: Towards a methodology for quality improvement based on the ip-map framework. In: 7th Intl Conf. on Information Quality (ICIQ-02). (2002) 8–10
29. Lin, L., Nuseibeh, B., Ince, D., Jackson, M., Moffett, J.: Introducing abuse frames for analysing security requirements. (2003)
30. Jürjens, J.: Secure systems development with UML. Springer-Verlag New York Incorporated (2005)