Modeling and analyzing Information Quality Requirements for Socio-technical Systems: Experience Report

Mohamad Gharib and Paolo Giorgini

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

Abstract. Information Quality (IQ) is particularly important for the efficient performance of any system. Despite this, most of the Requirements Engineering (RE) frameworks either ignore IQ needs, or they deal with them as mere technical issues, i.e., they do not consider the social and organizational aspects that underlie such needs. This paper summarizes the experience of the authors in modeling and analyzing IQ requirements for socio-technical systems. In particular, it summarizes the authors effort to propose an integrated RE framework that provides concepts for modeling and reasoning about IQ requirements since the early phases of the socio-technical system development.

Keywords

Information Quality, Requirements Engineering, Modeling, Reasoning

1 Introduction

Nowadays, Information Quality (IQ) is a growing concern for most organizations, since they depend on information for managing their daily operations, taking important decisions, etc., and relying on low quality information may negatively influence their overall performance, or it might even leads to disasters in the case of critical systems (e.g., air traffic management systems, etc.). Despite this, most existing Requirements Engineering (RE) frameworks and approaches either loosely define, or simply ignore IQ requirements. Several techniques for dealing with IQ have been proposed in the literature (e.g., integrity constraints). However, most of them focus on the technical aspects, and do not solve problems that may arise at organizational or social levels. More specifically, these techniques do not satisfy the needs of current complex systems, such as socio-technical systems [1], where humans and their interactions are considered as an integral part of the system along with the technical elements (e.g., smart cities, etc.).

Fisher and Kingma [2] highlighted the limitation of existing IQ techniques for addressing IQ related issues that might arise at the social or organizational level, where different kinds of vulnerabilities might manifest themselves in the actors' interactions and dependencies. For instance, the Flash Crash (a main stock market crash) is an example where the problem was not caused by a mere technical failure, but it was due to several socio-technical related vulnerabilities of the system [3]. In particular, several reasons contributed to the Flash Crash were caused by socio-technical IQ related issues. For example, some traders intentionally provide inaccurate information (e.g., fraud, falsified, etc.). Others continue trading during the crash by forwarding their orders to the markets that did not halt their trading activities due to lack of coordination among the markets, where the lack of coordination resulted also from IQ related vulnerabilities. We advocate that such failures could be avoided if the IQ requirements of the system were captured properly during the system design.

In addition, most of these approaches provide ad-hoc techniques to deal with IQ related vulnerabilities, instead of solving the main reason of such vulnerabilities by considering them during the early phase of the system development (e.g., requirements level). In particular, a RE framework that enables for modeling and reasoning about IQ requirements is still missed. To this end, we proposed a novel RE framework that adopts the Goal-Oriented Requirements Engineering (GORE) paradigm. Among the several GORE approaches offered in the literature (e.g., KAOS, *i**, etc.), we adopted Secure Tropos [4] as a baseline for our framework. Secure Tropos introduces primitives for modeling actors of the system along with their objectives, entitlements and capabilities. Moreover, it allows for modeling the social and organizational environment where the system will be eventually implemented, which is the main reason for choosing it as baseline for our proposed framework. In particular, our proposed framework extends the conceptual model of Secure Tropos by providing concepts and constructs for modeling and reasoning about IQ requirements. In what follows, we list and discuss the main objectives of this research.

Objectives of the research : as previously discussed, a RE framework for capturing IQ requirements is still missing. This paper summarizes our effort to solve this problem by proposing a framework that is able to model and reason about IQ requirements. In particular, through this paper we summarize our work trying to answer the following research questions:

RQ1: How can we analyze IQ in socio-technical systems? in other words, which IQ dimensions should be considered for analyzing IQ from socio-technical perspective. We summaries our effort to answer this question in section 3;

RQ2: How can we model IQ requirements? we summaries our effort to answer this question in section 4 by proposing concepts and constructs for modeling IQ requirements;

RQ3: How can we verify the correctness of the IQ requirements model? we summaries our effort to answer this question in section 5, by discussing a set of properties of the design that can be used to verify the correctness and consistency of the IQ requirements model.

RQ4: How can we support system designers in constructing the system? we summaries our effort to answer this question in section 6 by proposing a methodological process to be followed by system designers during the different phases of the system design.

RQ5: How well does the framework perform when applied to realistic settings? we summaries our effort to answer this question in section 7 by performing a set of experiments to verify whether the framework can efficiently perform the tasks it has been developed to do.

This experience report is structured as follows; Section (§2) describes our motivating example, while in Section (§3) we propose our multi-dimensional model for analyzing IQ. In Section (§4), we highlight the limitation in Secure Tropos for dealing with IQ requirements, and then we propose the required modeling concepts and constructs. Section (§5) presents the reasoning techniques that our framework offers; and in Section (§6) we introduce the methodological process that underlies our framework. Section (§7) implements and evaluates the proposed framework. Finally, in Section (§8) we present our ongoing work, and we conclude the report at Section (§9).

2 Motivating Example

Our motivating example concerns the May 6, 2010 Flash Crash, in which the Dow Jones Industrial Average (DJIA) dropped about 1000 points (9% of its value). Based on [5,6], we can identify several main stakeholders of system, including: *stock investors* are individuals or companies, who have a main goal of making profit from trading securities. While *stock traders* are individuals or companies involved in trading securities in *stock markets* either for their own sake or on behalf of their *investors* with a main goal of making profit by trading securities. *Traders* can be classified under several main categories, including: *Market Makers* are traders who have the capability to trade large number of particular securities, and they facilitate trading on such securities in the market; *High-Frequency Traders (HFTs)* are traders who have the capability to trade with very high trading frequency; and *Small traders*: trade small amount of securities with low trading frequency.

Stock markets are the places where *traders* gather and trade securities (e.g., NYSE, Chicago Mercantile Exchange (CME), NASDAQ). *Markets* have a main goal of making profit by facilitating security trading. Usually, they manage traders' order matching and should ensure stable trading environment, which can be done by depending on their Circuit Breakers (CBs), where a CB is a technique that is used to slow down or halt trading to prevent a potential market crash. Furthermore, *accounting firms* are specialized for performing accounting activities, i.e., provide companies with clear and reliable information about their economic activities and the status of their assets. While *auditing firms* are specialized for providing efficient monitoring of the quality of information produced by companies concerning their financial statements. 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.

A deep analysis of the Flash Crash shows that several reasons that led to the failure were caused by IQ related vulnerabilities. In what follows, we list and discuss the main theories that have been proposed to explain the Crash: (1) Fat-finger trade that is a human error caused by pressing a wrong key when using a computer to input data; (2) The suspicious behavior of some HFTs that negatively effects the market prices and contributed to the Flash Crash [5,6]; (3) Fraud/ falsified information that have been used by some actors and negatively influence the overall system performance. E.g., HFTs' flickering quotes that last very short time, which make them unavailable for most of traders, and Market Makers' stub quotes that are orders with prices far away from the current market prices[6]; (4) Undetected vulnerabilities in the socio-technical system design that led to a failure in the overall system.

3 A Multi-dimensional Model for IQ Analysis

IQ is a hierarchical multi-dimensional concept that can be characterized by different dimensions/ sub-dimensions (e.g., accuracy, completeness, consistency, etc. [7,8]). Although there exist several models for analyzing IQ (e.g., [9,8,10]), yet they were not designed to capture the needs of socio-technical systems, i.e., they do not consider the social and organizational aspects that might underlie some IQ dimensions.

In [11], we analyzed IQ based on four of its dimensions (highlighted within a box in Figure 1), namely: accuracy, completeness, timeliness and consistency. While in [12], we considered an extended model (shown in Figure 1) for analyzing IQ based on seven IQ dimensions: accessibility, accuracy, believability, trustworthiness, completeness, timeliness and consistency. In addition, it considers the intentional, social and organizational aspects that might underlie these dimensions. We define and discuss each of these dimensions along with their interrelations and how they can be analyzed as follows:

Accessibility: the extent to which information is available, or easily and quickly retrieved [7]. We limit accessibility definition to having the required permission over information to perform a task at hand.

Believability: can be defined as to which extent information is accepted or regarded as true [7,8]. Concerning our motivating example, if *markets* apply mechanisms to analyze the *believability* of the trading orders, they will be able to detect "stub quotes" that have been provided by some Market Makers, and in turn, they can mitigate their negative influence.

Trustworthiness: can be defined as the extent to which information is credible [10]. We relied on the *trustworthiness of the provenance* to analyze trustworthiness, which enables for capturing any information that helps in determining the trustworthiness of information based on its source (*trustworthiness of the source*), and the process by which it has been delivered to its destination (*trust-*

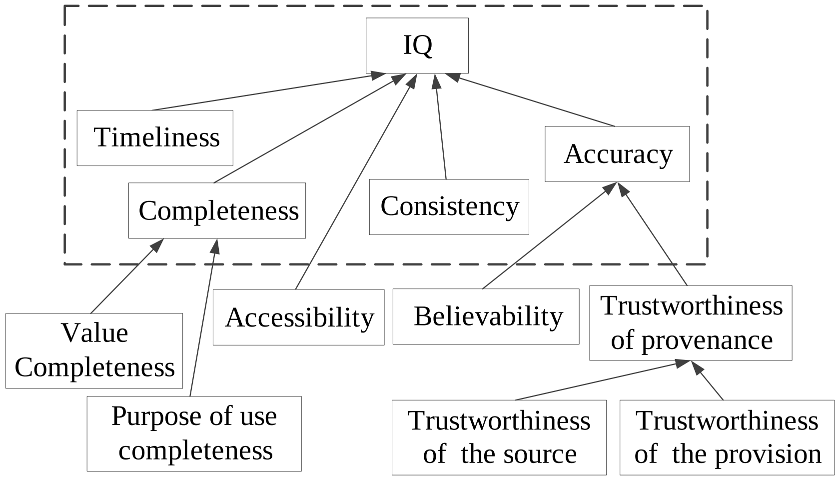

Fig. 1. Multi-dimensional model for analyzing IQ for socio-technical systems

worthiness of the provision). Concerning our example, if *markets* apply a mechanism to analyze the *trustworthiness* of the trading orders, they will be able to detect "flickering quotes" that have been provided by some HFTs, and apply the required mechanisms to mitigate their harmful effect.

Accuracy: means that information should be true or error free with respect to some known or measured value [8]. Accuracy is the most important and studied dimension, yet without clear standards, estimating accuracy is not an easy task. However, Dai et al. [13] stated that information accuracy is highly influenced by information *trustworthiness*. While Wang and Strong [14] argued that accuracy can be analyzed based on several dimensions including *believability*. Thus, we analyzed accuracy based on these two sub dimensions.

Completeness: means that all parts of information should be available, and information should be complete for performing a task at hand [8]. Thus, completeness can be analyzed depending on two sub dimensions: *Value Completeness*: information is preserved against corruption or lost that might endanger its integrity (e.g., during its storage/ transfer); and *Purpose of use completeness*: information is complete for performing a task at hand, i.e., all the required information items for performing a specific task should be available. Concerning our example, markets depend only on their own CBs information to stabilize their trading environment. However, such information is enough for each market alone, but when it comes to coordinate the CBs activities among all the markets, it can be considered as incomplete information.

Timeliness: means to which extent information is valid in term of time (e.g., sufficiently up-to-date) [7]. According to [15], information timeliness can be analyzed depending on information *currency* that is the time interval between its creation or update to its usage time [14,7]), and information *volatility* that is

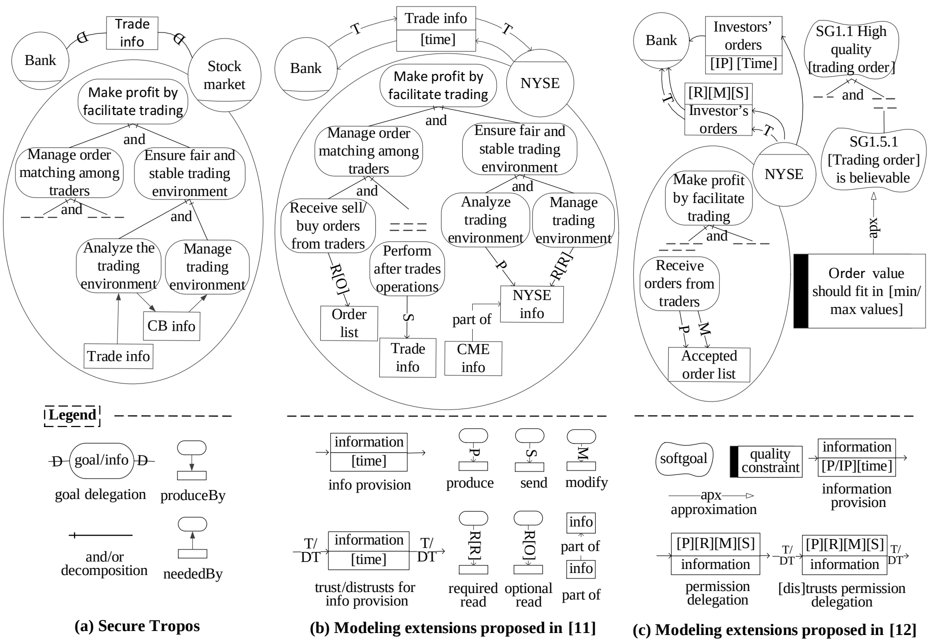

Fig. 2. A partial goal model of the stock market system represented in secure Tropos, and its extended versions

the change frequency of information value [14], i.e., information is not valid, if its currency is bigger than its volatility interval, otherwise it is valid.

Consistency: means all multiple records of the same information should be the same across time and space [8]. Concerning our example, the lack of coordination among CB activities of the trading markets will not be resolved unless markets depend on consistent information for their CB activities.

4 Extending Secure Tropos with IQ modeling concepts

Secure Tropos (Figure 2 (a)) is able to capture the social and organizational aspects of the system-to-be, but it does not offers primitives for capturing IQ requirements, i.e., it just deals with information whether they are available or not and who is responsible about their provision. To tackle this problem, we proposed a framework [11] that extend the conceptual model of Secure Tropos with concepts for modeling and analyzing four IQ dimensions (Figure 2 (b)). However, the framework does not provide a systematic process that justify why a certain IQ dimension should be considered or not for analyzing IQ. Thus, in [12] (Figure 2 (c)), we extended our previous framework with a mechanism for capturing IQ requirements based on the actual purpose of information usage, and then gradually refining them in terms of seven different IQ dimensions.

In particular, our modeling extensions can be classified under the following two sets: *(I) Basic IQ concepts*: that provide constructs for modeling seven IQ dimensions, and *(II) Top-level IQ concepts*: that are used to capture the IQ requirements of the stakeholders based on the actual information usage, and then gradually refining them until reaching their operational specifications. More specifically, this set is used to identify how top-level IQ requirements can be captured and then refined in terms of their different IQ dimensions, which can be modeled by the basic IQ concepts.

(I) Basic IQ concepts: in what follows, we summarize the concepts that we have proposed in both [11,12] for modeling IQ requirements:

Goal-Information relations: we refined Secure Tropos goal-information relations by introducing four different concepts, *(P)roduce* indicates that an information item can be created by achieving the goal that is responsible of it producing; *(R)ead* indicates that a goal consumes an information item , and it can be either *optional* in which a goal can be achieved even if information was not consumed, or *required* in which information is required for the goal achievement; *(M)odify*: indicates that the goal achievement depends on modifying an information item; and *(S)end*: indicates that the goal achievement depends on transferring an information item to a specific destination under predefined criteria.

Accessibility: can be influenced by the permissions that an actor has over information, which might enables or prevents it from using information as intended. Thus, we proposed four types of permissions concerning the four types of information usage that our framework supports (e.g., (P)roduce, (R)ead, (M)odify and (S)end). In addition, we extend the language to model permission delegation among actors, and to model trust/ distrusts concerning such permissions.

Believability: we extended *read* and *produce* concepts to accommodate *believability* check, since they are the only relations that can be influenced by information believability.

Trustworthiness: is subject to (1) *trustworthiness of the source* that can be captured by *trust/distrust produce* relations between information consumer and its producer concerning the produced information; and the (2) *trustworthiness of the provision.* In [11], we proposed trusted/ distrusted provision to analyze information trustworthiness. However, we noticed that such concepts are at high abstraction level, and seem to be inappropriate for identifying detailed IQ specifications. Thus, we refined these two concepts in [12], and we analyzed the *trustworthiness of the provision* based on the way information arrives to its destination (e.g., P/IP provision), and the operations (e.g., modify) that have been applied to it taking into consideration if such operations were authorized or not.

Accuracy: can be analyzed based on : (1) *Accuracy of produced information* that can be analyzed based on its *believability*, which enables to avoid producing unintended information, and its *trustworthiness of the production process* if the producing goal has been delegated; (2) *Accuracy of provided information* can be analyzed based on the *trustworthiness of the provision*; and (3) *Accuracy of read information*: can be analyzed based on information *believability* and its *trustworthiness of the provenance.*

Completeness: completeness can be subject to (1) *value completeness* for which we rely on *Integrity Preserving provision* (IP-provision) that preserves the integrity of the provided information; and (2) *purpose of use completeness* we rely on the "Part of" concept to model the relation between an information item and its sub-items.

Timeliness (validity): we extended information concept with a *volatility* attribute to represent the change rate of information value [14], and we proposed *read time*, and *send time* to analyze the validity of read and send information respectively. *Read time*[1] captures the actual *usage time* of information, and it enables to determine information *currency (age)*, which can be used to analyze the timeliness of read information, i.e., information is valid, if its currency is smaller than its volatility, otherwise it is not invalid. While *send time* represents the allowed amount of time for information to reach its final destination. The timeliness of send information at its destination can be analyzed depending on the *send time* and the *read time* at its destination, i.e., information is valid, if *read time* is smaller than the *send time*. Finally, we extended *information provision* with a time aspect that represent the transmission time.

Consistency: we extend the *read* relation between a goal and information with *Purpose Of Use* (POU) attribute that captures the intended purpose of information usage, which enables to identify *interdependent readers* that are actors who read the same information for the same *POU*. To this end, consistency can analyzed among *interdependent readers* based on their *read times*, i.e., information is consistent among its *interdependent readers*, if all of them have the same *read time*, otherwise it is inconsistent.

(II) High-level IQ concepts: in what follows, we summarize the concepts that we have proposed in [12] for modeling high-level IQ requirements:

Top-level IQ softgoals: are softgoal concerning IQ requirements, and they are used as a starting point for identifying the stakeholders' needs concerning IQ.

And-decomposition for IQ softgoals refinement: since a softgoal can be refined into more specific sub softgoals, if the joint satisfaction of these softgoals is considered equivalent to satisfying the refined softgoal [16]. We introduced *And-decomposition* relation between an IQ softgoal and its sub IQ softgoals.

Approximating leaf IQ Softgoals: we adopted the *approximation* relation proposed by Jureta et al. [16] through which a softgoal can be satisfied by a Quality Constraint (QC), where a QC can provide clear-cut criteria for the satisfaction of a softgoal. However, leaf IQ softgoals are used to capture different IQ dimensions (e.g., accuracy, completeness, etc.), i.e., each of them is used to describe different aspects of IQ. Thus, leaf IQ softgoals might not have the same nature/type, and in turn, they may need to be approximated in different ways.

Thus, to get better understanding of leaf IQ softgoals nature/type, and to define the appropriate Information Quality Constraints (IQC)[2] for their approximation, we relied on Glinz [17] work that classifies requirements based on their

[1] Read time can be derived by analyzing the model, i.e., there is no specialized construct or attribute to represent it

[2] We use IQC to refer to QC, since no other types of constraints are used in this paper

Table 1. IQ softgoal classification & approximation into IQC

Leaf IQ softgoals	Kind	Satisfaction	Representation	Approximated into IQC
Believability	Functional	Hard	Operational	Operational IQC
Trustworthiness	Constraint	Hard	Declarative	Declarative IQC
Completeness	Constraint	Hard	Declarative	Declarative IQC
Timeliness	Performance	Hard	Quantitative	Quantitative IQC
Consistency	Performance	Hard	Quantitative	Quantitative IQC

kind, satisfaction and representation, and we classify leaf IQ softgoals under three main categories.

In addition, for the approximation to be consistent with the different types of leaf IQ softgoals, we defined three different types of IQCs: (1) *Operational IQC:* are constraints that define the required actions to be performed in already determined situations; (2) *Declarative IQC:* are constraints used to define properties of the system that should hold; and (3) *Quantitative IQC:* are constraints used to specify properties of the system that should hold, and can be measured on an ordinal scale. Table 1 shows how leaf IQ softgoals can be classified and approximated into the appropriate IQCs.

Finally, in order for the approximation relation between IQ softgoal and its related IQC to hold, a well-defined quality space should exist [16], where a quality space can be defined as a certain conceptual space that can be used to describe the quality value [18]. The main purpose of the quality space is removing any ambiguity related to the verification of IQCs, i.e., determining whether a certain IQC is satisfied or not. Consider timeliness for example; how time is represented and measured should be clear to all stakeholders of the system, i.e., the allowed number of digits along with the value they represent (e.g., seconds, etc.).

5 Automated Analysis Support

We used Disjunctive Datalog [19] to formalize all the concepts along with the related axioms (reasoning rules) that have been introduced in the two frameworks [11,12]. In addition, for each of them we defined a set of properties of the design that can be used to verify the IQ requirements model. Table 2 lists some properties of the design; in what follows we discuss each of them:

Pro1 states that the model should not include any top-level goal that is not *achieved* from the perspective of the actor, who aims for it. This property can be used to quickly verify the IQ requirements model, i.e., if this property holds for all top-level goals, we can infer that the requirements model is correct and consistent. **Pro2** states that the model should not include any goal delegation chain, if there is no trust chain holds between the delegator and the delegatee, since delegation with no trust leaves the delegator with no guarantee about the achievement of its goal. **Pro3** states that actors should have all information that is required for the achievement of the goals they are responsible of.

Table 2. Some properties of the design

Goal properties	
Pro1	`:- aims(A,G), not achieved(A,G)`
Pro2	`:- deleChain(A, B, G), not trustChain(A, B, G)`
Information availability property	
Pro3	`:- reader(_, _, A, I), information(I, T), not has(A,I,T), #int(T)`
Permissions properties	
Pro4	`:- need_perm(P, A, I), not has_perm(P, A, I)`
Pro5	`:- dele_perm(P, A, B, I), not has_perm(P, A, I)`
Pro6	`:- has_perm(P, B, I), own(A,I), not trust_perm_chain(P, A, B, I)`
IQ properties	
Pro7	`:- producer(A, I, T), not fits_produce(A,I)`
Pro8	`:- sender(T, A, B, I), not fits_send(T, A, B, I)`
Pro9	`:- reader(T, P, A, I), has(A, I, T), not fits_read(A, I)`
Conflict of interest properties	
Pro10	`:- play(A, R1), play(A, R2), conflict_roles(R1, R2, produce, I), producer(A, I, T)`
Pro11	`:- play(A, R1), play(A, R2), conflict_roles(R1, R2, read, I), reader(T, P, A, I)`

Pro4-6 are used to verify information permission related properties. For instance, **Pro4** states that actors should have all the permissions they require to achieve their objectives. While **Pro5** states that the model should not include actors who delegate permissions that they do not have. **Pro6** states that the model should not include actors who have permissions, and there is no trust/ trust chain between such actors and information owner.

Pro7-9 are used to verify IQ related properties, i.e., the model should not include any information that does not fits for the purpose of use (e.g., produce, read, and send) from the perspectives of their user (e.g., producer (**Pro7**), reader (**Pro8**), and sender (**Pro9**)). Note that each of these properties covers several sub-properties. Consider **Pro8** for example, information fits for read if it is accessible, accurate, complete, valid, and consistent.

Pro10-11 are used to ensure that the model manage separation of duties among its actors to avoid any conflict of interest that may leads to different kinds of vulnerability, i.e., such properties allow to identify situations at the instance level that might endanger the system performance. In particular, they state that the model should not include any agent that plays conflicting roles in terms of producing and/or reading information. The Enron scandal [20] is a famous example of producing inaccurate (intentionally biased) information due to playing conflicting roles. Defining conflicting roles in terms of producing/reading information is not an easy task, and it is done with the help of domain experts.

These properties allow for the automated analysis of the IQ requirements model, i.e., the analysis relay on them to detect any design violation and notify the designer about it. In other words, they enable for checking whether stakeholders' IQ requirements are achieved or not, and identify the reason(s) preventing their achievement (if any). In what follows, we list the main violations to the properties of the design concerning the Flash Crash scenario that the automated analysis was able to capture:

Inaccurate information: markets (e.g., *CME*) consider information received from any trader that plays the role of *HFT trader* as inaccurate, since no trust in information production holds between them. In particular, HFTs has the capability (trade large amount of securities in very fast rate) of providing inaccurate information that is able to destabilize the trading environment. According to [21], there was 16 HFTs for the security that was under suspicion of triggering the crash.

Inaccurate information due to playing conflicting roles: some companies are playing two conflicting roles "accounting firm" and "auditing firm" concern producing "financial statement". Since we cannot trust a company for producing accurate "financial statement" concerning the companies they get paid from to perform their accounting services, such information is considered as inaccurate.

Unauthorized read due to playing conflicting roles: some companies are playing two conflicting roles "auditing firm" and "consulting firm" concern reading "security assessment", since they might use such information for providing paid consulting services. Thus, such companies should not has read permissions concerning "security assessment" information.

Incomplete information: " NYSE CB info" and "NASDAQ CB info" that are used to stabilize the trading environment in NYSE and NASDAQ markets respectively, are identified as incomplete information from the perspectives of their readers, since they miss a sub part related to the purpose of use. However, this can be solved by providing NYSE and NASDAQ markets with the missed information sub items "CME CB info".

Inconsistent information: *CME CB info* is provided to *NASDAQ* and *NYSE* with two different provision times 13 ms and 14.65 ms respectively [22], which leads to two different *read times* between them, and in turn, results in inconsistency since they are *interdependent readers.*

Following Secure Tropos, our framework allows for a clear distinction between organizational (e.g., roles) and instance (e.g., agents) levels, and most of these properties apply to both of these levels. However, to avoid conflicts between the organizational and instance levels that is a main issue in Secure Tropos, we are planning to model actors' interactions only at instance level.

6 Methodological Process

Our framework [11] is equipped with an engineering methodological process (shown in Figure 3) that consists of seven steps, which should be followed by designers during the system design; each of these steps is described as follows: (1) *Actors modeling*: aims to model the actors (e.g., roles and agents) of the system along with their objectives, entitlements and capabilities; (2) *Goals modeling*: aims to refine the actors' top-level goals, if needed, through and/ or decomposition until reaching their leaf goals; (3) *Information modeling*: the different relations between goals and information are identified and modeled along with their IQ needs, and the structure of composed information are modeled as well.

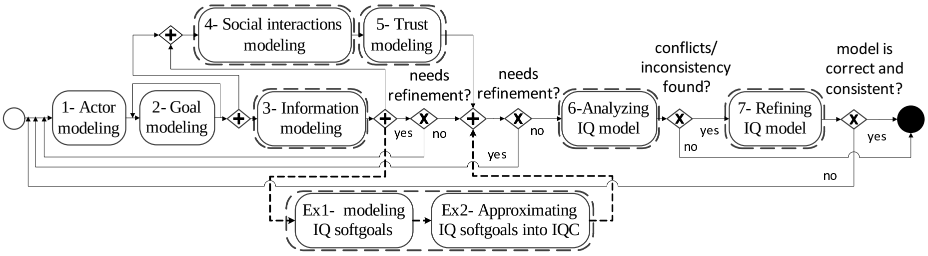

Fig. 3. The Methodological process for modeling and reasoning about IQ requirements

(4) *Social interactions modeling*: aims to identify and model the different social dependencies among actors of the system concerning information provision, goal and permission delegation. More specifically, based on actors' capabilities some goals might be delegated to other actors, who have the capabilities to achieve them; and based on actors' needs, information/ permission is provided/ delegated to them; (5) *Trust modeling*: aims to model trust/ distrusts among actors concerning information producing, goal and permission delegation; (6) *Analyzing IQ model*: at this step the model is analyzed to verify whether all the stakeholders' requirements are achieved or not; and (7) *Refining IQ model*: if some of the stockholders' requirements were not achieved during the model analysis step, the analyst try to find solutions for such issues at this step.

In [12], we refined some steps of the previously discussed methodology to accommodate the proposed modeling extensions (surrounded by dashed lines in Figure 3). For instance, step (4) *Social interactions modeling* is extended to model permissions delegation among actors, while step (5) *Trust modeling* is extended to model trust/ distrusts concerning the delegated permissions. Moreover, we have added two more steps Ex1 and Ex2 to the methodological process, where the first aims to identify and model top-level IQ softgoals, and refine them until reaching their leaf IQ softgoals, while the last aims to approximate leaf IQ softgoals into their corresponding IQC.

7 Implementation and evaluation

We evaluated the applicability and effectiveness of our framework and its extended version depending on simulation method (experimental), i.e., execute artifact with artificial data. To this end, we developed a prototype implementation of our framework (ST-IQ Tool)[3] to test it for modeling and reasoning about IQ requirements. In what follows, we briefly describe our 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 4 main parts (ST-IQ Tool structure is shown in Figure 4): (1) Control component (JAVA based-program) that

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

controls and coordinates the three other components; (2) a graphical user interface (GUI) developed using Sirius[4], which enables the system designers for drawing the IQ requirements model diagram by dragging-and-dropping modeling elements from palettes, and allows for specifying the properties of these elements along with their interrelations; (3) model-to-text transformation mechanism that supports the translating of the graphical requirements models into Disjunctive Datalog formal specifications depending on Acceleo[5]; and finally (4) automated reasoning support (DLV system[6]) that takes the Disjunctive Datalog specification, which 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 evaluated our framework and its extended version by showing their applicability in capturing the IQ requirements along with its effectiveness in identifying any violation to the properties of the design by applying it to the Flash Crash case study. In particular, we used our modeling language to model the Flash Crash, and then we translated the requirements diagram into Disjunctive Datalog formal language. Finally, we depend on the automated reasoning support to check whether the requirements model is correct and consistent, i.e., whether all the properties of the design hold. The analysis captured all violations of the properties of the design that we consider including inaccurate, incomplete, inconsistent information, etc.

Experiments on scalability: to test the scalability of the automated reasoning support, we investigated the reasoning execution time with respect to the model size. In particular, we proposed a model and then expanded it by increasing the number of its modeling elements through several steps. At each step, we performed an analysis test and we calculated the execution time. The result showed that the relation between the size of the model and the execution time is linear (not exponential).

8 Ongoing work

We briefly discuss several ongoing works to improve our proposed framework.

Social trust analysis: all RE approaches including ours mainly focus on trust as social relations, without proposing any specific modeling technique to relate them with the internal requirements of the system's components. In other words, existing approaches are able to model trust requirements, but offer no analysis mechanisms to verify their consistency with the different actors' competencies and motivations toward the trustum. We are currently working on extending our framework by proposing the required concepts and constructs for modeling and analyzing trust among actors of the systems based on sets of beliefs related to the actors' competencies (can do) and motivations (will do), which can be used to clearly identify "why" an actor can trust/ distrusts another one.

[4] https://projects.eclipse.org/projects/modeling.sirius
[5] https://projects.eclipse.org/projects/modeling.m2t.acceleo
[6] http://www.dlvsystem.com/dlv/

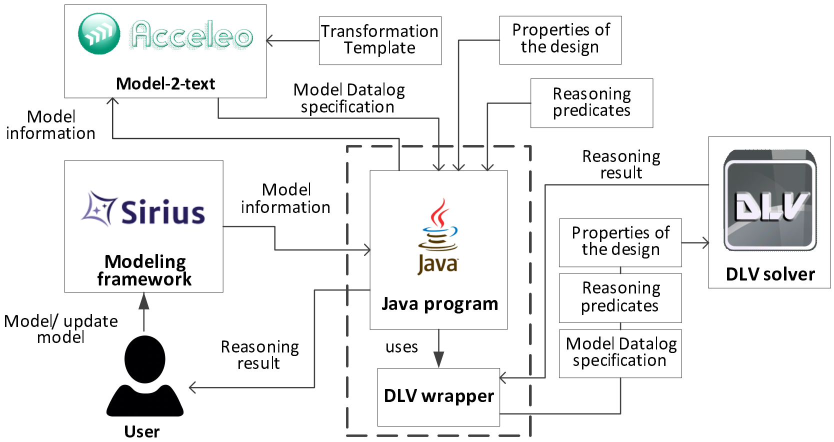

Fig. 4. ST-IQ Tool architecture

Automated specification of IQ policies: organizations rely on different kinds of policies to define the permitted/ forbidden actors' activities toward information in order to deal with IQ related concerns. Yet most of the proposed solutions focus on the technical aspects of the system, which make them inadequate for socio-technical systems. Thus, we extended our framework to support the automatic derivation of the final IQ policies from the IQ requirements model [23]. In particular, we proposed an IQ policy specification language that is able to clearly represent the final IQ specifications in terms of permitted, forbidden and obligated actors' activities toward information. In addition, we introduced a set of rules that enables for the automatic derivation of such policies from the requirements model. Currently, we are working on refining the IQ policy specification language to make it more expressive, and we aim to extend the derivations rules to cover more IQ policies.

Modeling and analyzing IQ requirements in Business Processes (BP): driven by the needs of some system designers, who put more emphasis on modeling BPs within the system rather than modeling the whole system, we proposed an extension to our framework [24] that offers mechanisms for modeling and analyzing IQ requirements in BPs. In particular, we proposed a detailed approach for capturing IQ requirements of the overall system where the BP is executed, and then we introduced mechanisms for mapping these requirements into workflow net with actors (WFA-net) that is a graphical language for modeling and analyzing IQ requirements in BP.

9 Conclusions

In this paper, we summarized our experience in modeling and analyzing IQ requirements for socio-technical systems. First, 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 consider the social and organizational context where the system will operate. Moreover, we highlighted the importance of capturing IQ requirements of the system from the early development phases.

Second, we showed how IQ can be analyzed through its different dimensions, and we proposed a multi-dimensional model for analyzing IQ from socio-technical perspective. Third, we justify our choice in considering Secure Tropos as a baseline for our proposed framework, and we discussed its limitations for modeling and reasoning about IQ requirements. In addition, we discussed the required extensions of Secure Tropos for capturing IQ requirements, and then we extended its conceptual model with the required concepts. In particular, our proposed framework enables system designers to capture IQ requirements in terms of its different dimensions, and it provide the required analysis techniques to verify whether the IQ requirements are met or not. In addition, we discussed our methodological process that supports designers during the different phases of the system design.

We illustrated the utility of our framework by applying it to a stock market crash case study. More specifically, we evaluated our framework by showing its ability in modeling and analyzing IQ requirements along with its effectiveness in detecting any violations to the properties of the design. Moreover, we evaluated the scalability of its reasoning techniques by calculating the relation between the model sizes and the reasoning execution time, and the result proves that such relation is linear (not exponential). Finally, we discussed our ongoing research related to IQ requirements.

Acknowledgment

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

References

1. Emery, F., Trist, E.: Socio-technical systems. management sciences, models and techniques. churchman cw et al (1960)
2. Fisher, C., Kingma, B.: Criticality of data quality as exemplified in two disasters. Information & Management **39**(2) (2001) 109–116
3. 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
4. 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
5. Securities, Commission, E., 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)
6. 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)

7. Pipino, L.L., Lee, Y.W., Wang, R.Y.: Data quality assessment. Communications of the ACM **45**(4) (2002) 211–218
8. 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
9. Wang, R.Y., Reddy, M.P., Kon, H.B.: Toward quality data: An attribute-based approach. Decision Support Systems **13**(3) (1995) 349–372
10. Liu, L., Chi, L.: Evolutional data quality: A theory-specific view. In: IQ. (2002) 292–304
11. Gharib, M., Giorgini, P.: Modeling and reasoning about information quality requirements. In: Requirements Engineering: Foundation for Software Quality - 21st International Working Conference, REFSQ 2015, Essen, Germany, March 23-26, 2015. Proceedings. (2015) 49–64
12. Gharib, M., Giorgini, P.: Dealing with information quality requirements. In: Enterprise, Business-Process and Information Systems Modeling (EMMSADŠ15), to appear. Springer (2015)
13. Dai, C., Lin, D., Bertino, E., Kantarcioglu, M.: An approach to evaluate data trustworthiness based on data provenance. In: Secure Data Management. Springer (2008) 82–98
14. Wang, R., Strong, D.: Beyond accuracy: What data quality means to data consumers. Journal of management information systems (1996) 5–33
15. 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
16. Jureta, I.J., Mylopoulos, J., Faulkner, S.: Revisiting the core ontology and problem in requirements engineering. In: International Requirements Engineering, 2008. RE'08. 16th IEEE, IEEE (2008) 71–80
17. Glinz, M.: Rethinking the notion of non-functional requirements. In: Proc. Third World Congress for Software Quality. Volume 2. (2005) 55–64
18. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Schneider, L.: Dolce: a descriptive ontology for linguistic and cognitive engineering. WonderWeb Project, Deliverable D17 v2 **1** (2003)
19. Bihlmeyer, R., Faber, W., Ielpa, G., Lio, V., Pfeifer, G.: Dlv-user manual. The DLV Project (2009)
20. 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
21. Paddrik, M., Hayes, R., Todd, A., Yang, S., Beling, P., Scherer, W.: 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, IEEE (2012) 1–8
22. Lewis, M.: Flash boys: a Wall Street revolt. WW Norton & Company (2014)
23. Gharib, M., Giorgini, P.: A goal-based approach for automated specification of information quality policies. In: Research Challenges in Information Science (RCIS), 2015 IEEE Ninth International Conference, to appear, IEEE (2015)
24. Gharib, M., Giorgini, P.: Modeling and reasoning about information quality requirements in business processes. In: Enterprise, Business-Process and Information Systems Modeling (BPMDSŠ15), to appear. Springer (2015)