

# Dealing with Information Quality Requirements

Mohamad Gharib and Paolo Giorgini

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

**Abstract** Information Quality (IQ) has been a growing concern for most organizations, since they depend on information for managing their daily tasks, make important decisions, etc., and relying on low quality information may negatively influence their overall performance. Despite this, most of the Requirements Engineering approaches either ignore or loosely define such requirements, i.e., they deal with them as generic non-functional requirements. In this paper, we propose a goal-oriented framework that is based on an extended version of secure Tropos methodology for modeling and reasoning about IQ requirements since the early phases of the system development, and refine these requirements until reaching their operational specifications. Moreover, the framework offers a methodological process along with several reasoning techniques to help designers during the different phases of the system design. We illustrate our framework with an example concerning a stock market crash.

**Key words:** Information Quality, Requirements Engineering, Modeling, Reasoning

## 1 Introduction

The importance of Information Quality (IQ) for organizations is out of discussion, since they depend on information for managing their daily tasks, make important decisions, etc., and relying on low-quality information may result in undesirable outcomes [1], or even disasters in the case of critical systems (e.g., Air Traffic Management). Despite this, most existing Requirements Engineering (RE) approaches either loosely define IQ requirements, or simply ignore them (e.g., UMLsec [2],  $i^*$  [3], etc.). In particular, like all non-functional (quality) requirements, IQ requirements use to be represented as generic qualitative properties of the system, without specific methods for their analysis [4].

Although there exists many technical solutions for dealing with IQ related concerns in storage, network and database systems related literature (e.g., integrity constraints), such solutions are not able to satisfy the needs of current complex systems, such as socio-technical systems [5]. Existing solutions are able to solve IQ related issues at the technical level, but seem to be limited in solving IQ issues that may rise at social or organizational levels [6]. The Flash Crash (a main U.S market crash) is an example about the limitation of such solutions for addressing IQ needs for socio-technical systems, where the crash was not

caused by a mere technical failure, but it was due to undetected vulnerabilities in socio-technical interactions [7].

In particular, the literature offers several RE approaches that are able to capture the social and organizational aspects of the system-to-be (e.g., secure Tropos [8], etc.), but their main focus is on the functionality of the system, and usually they ignore IQ needs. An integrated analysis of both functional and non-functional requirements is essential, since as highlighted in [4], some functional requirements might not be useful without their necessarily related non-functional requirements. Consider for example, an actor who wants to send an order to a stock market (functional requirements), and it requires its order to be send in an already defined period of time (non-functional requirements). If the system fails to satisfy time related aspects, requirements concerning the send of the order might not be achieved.

In [9], we have proposed a RE framework for modeling IQ requirements in terms of four IQ dimensions: accuracy, completeness, timeliness, and consistency. However, the framework does not provide a systematic process to guide the analysis of the four dimensions, and does not cover possible relations among them based on the actual information use. In this paper, we propose an extension to our previous framework [9] extending and refining its modeling concepts, and providing mechanisms for capturing IQ requirements, and then gradually refining them in terms of their different IQ dimensions until reaching their operational specifications. Moreover, the framework offers a methodological process along with several reasoning techniques to help designers during the different phases of the system design.

The paper is organized as follows; in Section (§2) we describe our motivating example concerning the Flash Crash scenario that is used to illustrate our framework. While in Section (§3), we propose our multi-dimensional model for analyzing IQ. Section (§4) introduce our proposed extensions for modeling IQ requirements, and in Section (§5), we briefly discuss the reasoning support that our framework offers. We implement and evaluate the proposed framework in Section (§6). Finally, we discuss the related work in Section (§7), and we conclude the paper and discuss future work in Section (§8).

## 2 Motivating Example

Our motivating example describes the May 6, 2010 Flash Crash, in which the Dow Jones Industrial Average (DJIA) dropped about 1000 points (9% of its value). Based on [10, 11], several stakeholders of system can be identified, including: *stock investors* are individuals or companies, who have a main goal of “making profit from trading securities”. While *stock traders* are persons 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*: facilitate trading on a particular security in the market, and they have the capability to trade very large number of securities; *High-Frequency Traders*

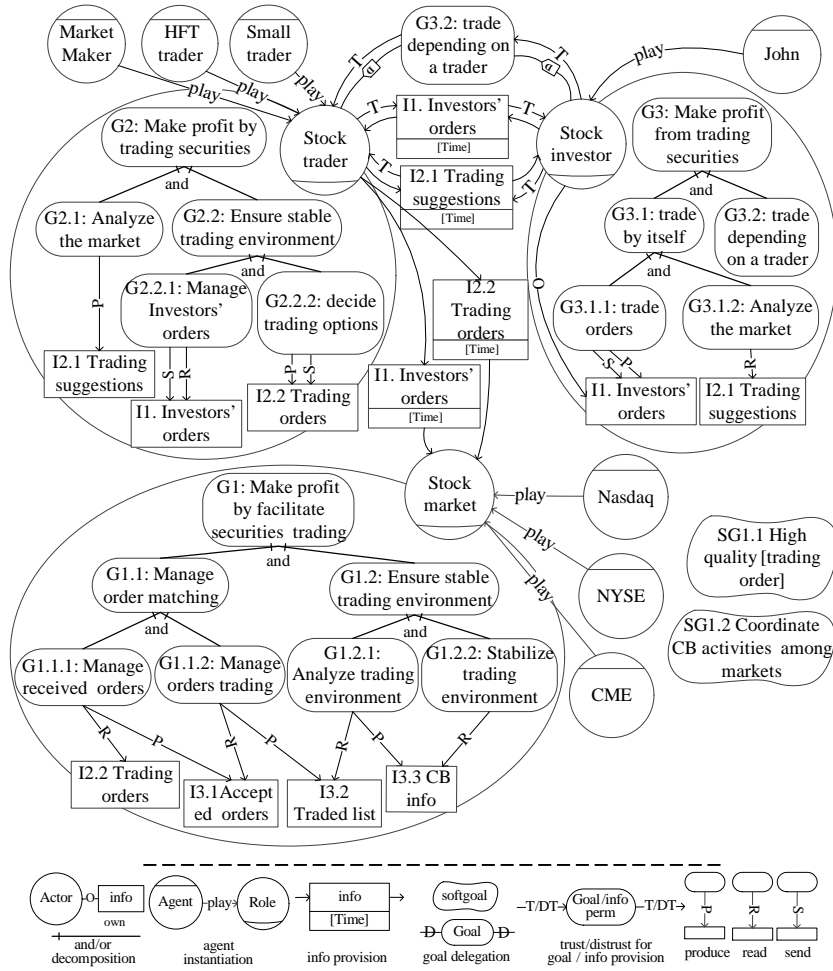


Fig. 1. A partial goal model concerning the Flash Crash scenario

(HFTs): are able to trade with very high trading frequency; and *small traders*: trade small amount of securities with very low trading frequency.

*Stock markets* are places where *traders* gather and trade securities, and they can be a physical trading place (e.g., New York Stock Exchange (NYSE), Chicago Mercantile Exchange (CME)), or electronic systems (e.g., NASDAQ). *Markets* have a main goal of “making profit by facilitating security trading”. Usually, they “manage traders’ order matching”, and they 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 activities to prevent a potential market crash.

Figure 1 shows a partial goal model of the Flash Crash scenario represented in the extended secure Tropos modeling language [9]. An actor covers two concepts

a role and an agent, where the first is an abstract characterization of the behavior of an actor (e.g., **Stock Market**), and the last is an actor within a concrete manifestations (e.g., **CME**). Moreover, an agent can *play* a role or more (e.g., **CME play a Stock Market**). Actors can have a set of goals, they intend to achieve (e.g., **G1: Make profit by facilitate security trading**). When a goal is too coarse to be achieved, it can be refined through AND/ OR-decompositions of a root goal into finer sub-goals (e.g., **G1:** is AND-decomposed into **G1.1:** and **G1.2:**), where an AND-decomposition means that in order to achieve the root goal, all of its sub-goals must be achieved, while for an OR, achieving only one of its sub-goals is enough.

Moreover, softgoals (e.g., **SG1.1**) are used to represent non-functional requirements, and they do not have clear-cut criteria for achievement [4]. An actor may own information, which gives it a full control concerning the usage of information it owns (e.g., **stock investor (O)wn I1**). A goal may (P)roduces, (R)eads, and (S)ends information (e.g., **G3.1.1: (P)roduces and (S)ends I1**). Actors may depend on one another for information to be provided, where information provision has a time attributes to describe the provision time, and they may trust/ distrusts one another for the provided information (e.g., **investor depend on trader for I2.1**, and trusts it for its provision). Finally, actors may delegate goals to one another, and trust/ distrusts one another for the achievement of the delegated goals.

### 3 Multi-dimensional Model for Analyzing IQ

There is a general consensus that IQ is a hierarchical multi-dimensional concept that can be characterized by different dimensions/ sub-dimensions (e.g., accuracy, completeness, consistency, etc. [12, 13, 14]). That is why deciding whether information is high or low quality is not an easy task, and it became harder for socio-technical systems, since intentional, social and organizational aspects might underlie some of these dimensions. Although there exist many models for analyzing IQ (e.g., [15, 13, 16]), yet most of them can be criticized by their ambiguity [14], inconsistency among the dimensions they consider (e.g., completeness is a sub-dimension of believability in [15], while it is a sub-dimension of integrity in [13]), and most of them were not designed to capture the needs of socio-technical systems, i.e., they do not consider intentional, social and organizational aspects of IQ.

Our multi-dimensional model (Figure 2) for analyzing IQ is based on 7 IQ dimensions: accessibility, accuracy, believability, trustworthiness, completeness, timeliness and consistency, and it consider the intentional, social and organizational aspects that might underlies these dimensions. We define and discuss each of these dimensions along with their interrelations as follows:

*Accessibility:* the extent to which information is available, or easily and quickly retrieved [12]. In this paper, accessibility is defined as having the required permission over information to perform a task at hand.

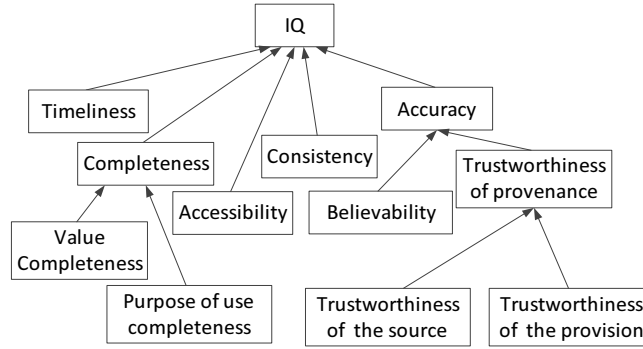


Fig. 2. IQ Model: the hierarchy of IQ dimensions

*Accuracy*: means that information should be true or error free with respect to some known, designated or measured value [13]. Accuracy is the most important and studied dimension, yet without clear standards, estimating accuracy is not an easy task. However, Dai et al. [17] stated that information accuracy is highly influenced by the *trustworthiness of its provenance*. While Wang and Strong [18] argued that accuracy can be analyzed based on several dimensions including *believability*. Thus, we analyze accuracy based on these two sub dimensions.

*Believability*: can be defined as the extent to which information is accepted or regarded as true and real [12, 13]. Concerning our motivating example, in order to fulfill their obligation (facilitate trading), Market Makers provide what is called “stub quotes”, which are orders with prices far away from the current market prices, i.e., such orders can be considered as fraud (inaccurate, falsified) orders. During the Flash Crash, over 98% of the trades were executed at prices within 10% of their values because of such orders [11]. However, such failure could be avoided, if *markets* apply a mechanism to verify the *believability* of the trading orders.

*Trustworthiness*: can be defined as the extent to which information is credible [16]. We rely on the *trustworthiness of the provenance* to analyze information trustworthiness, i.e., trustworthiness of information is analyzed depending on the (*trustworthiness of its source*), and the (*trustworthiness of its provision*) [17]. Concerning our example, some HFTs provide fraud/ falsified orders to affect the prices of some securities before starting their real trades (e.g., flickering quotes that are orders last very short time, which make them unavailable for most traders). If markets analyze the *trustworthiness of the provenance* of the trading orders, they will be able to detect such orders and apply the required mechanisms to mitigate their harmful effect.

*Completeness*: means that all parts of information should be available, and information should be complete for performing a task at hand [13]. 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 for performing a specific task should be available. Usually, the *purpose of use completeness* is harder to be analyzed and it requires a domain knowledge. For example, a main reason of the Flash Crash was the lack of coordination among the CBs of the trading markets. In particular, 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, such information can be considered as incomplete. For instance, during the Flash Crash CME employs its CB, but NYSE did not [19], since each of them depends only on its CB information.

*Timeliness*: means to which extent information is valid in term of time (e.g., sufficiently up-to-date) [12]. According to [20], information timeliness can be analyzed depending on information *currency (age)* that is the time interval between its creation (or update) to its usage time [18, 12]), and information *volatility* that is the change frequency of information value [18], i.e., information is not valid, if its currency is bigger than its volatility interval, otherwise it is valid. Note that timeliness can be subject to the needs of their stakeholders, i.e., stakeholders might define the timeliness of their own information. For example, stock investor can define the timeliness (validity time) of its sell/buy orders.

*Consistency*: means all multiple records of the same information should be the same across time and space [13]. In this paper, *consistency* is a time related aspect, i.e., the value of information among its different users might become inconsistent due to time related aspects (e.g., currency). For 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 Extended Concepts for Capturing IQ Requirements

Our previous framework [9] proposes concepts for modeling and analyzing 4 IQ dimensions, namely: accuracy, competence, timeliness, and consistency. However, it does not provide a systematic process that justifies why a certain IQ dimension should be considered or not for analyzing IQ, or how the considered IQ dimensions may contribute to one another. Thus, we extend our previous framework with mechanisms for capturing IQ requirements based on the actual purpose of use, and then gradually refining them in terms of 7 different IQ dimensions until reaching their operational specifications. In particular, we introduce two sets of modeling extensions: (I) *Basic IQ concepts*: that adopts and refine the concepts proposed in [9] for modeling 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 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 refined in terms of their different IQ dimensions, which can be modeled by the basic IQ concepts.

**(I) Basic IQ concepts**: first we extend and refine IQ modeling concepts proposed in [9] to accommodate the new IQ dimensions we consider along with

their interrelations. In particular, our previous framework introduces concepts for capturing IQ requirements in terms of their different dimensions such as accuracy, timeliness, consistency, etc. For instance, it introduces *trusted provision* concept that enables for capturing information accuracy, and it provides *information volatility*, *read timeliness* and *send timeliness* concepts for capturing information timeliness. Moreover, it proposes *interdependent readers* and *read-time* for capturing information consistency [21, 9]. However, some of the proposed concepts are at high abstraction level, and need to be refined to a level that enables for identifying detailed IQ specifications. For instance, it proposes *trusted provision* that helps in analyzing the accuracy of transferred information, yet we cannot rely on such concept to derive detailed IQ specifications. Moreover, we need to propose new constructs to accommodate the new IQ dimensions we consider (e.g., believability and accessibility). In what follows, we propose concepts to address these limitations.

*Accessibility*: can be influenced by the permissions that an actor has over information, which might enable or prevent it from using information as intended. However, our previous framework does not support the notion of permissions. Thus, we refine the modeling language by proposing 4 types of permissions concerning the 4 types of information usage that our framework supports (e.g., (P)roduces, (R)eads, (M)odifies and (S)ends). Moreover, we extend the language to model permission delegation among actors, and to model trust/ distrusts concerning the delegated permissions.

*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 against corruption or loss [22], i.e., the *value completeness* of information is guaranteed during its transfer; 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.

*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* that can be analyzed 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 (e.g., permissions and trust).

*Believability* is considered in both *read* and *produce*, since only these two relations can be influenced by information believability. Thus, we extend these two concepts to accommodate a *believability* check for read/produced information respectively, i.e., produced/read information is believable from the perspective of its producer/ reader, if the produce/ read operation apply a believability check.

*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 (e.g., fat finger mistakes), 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*

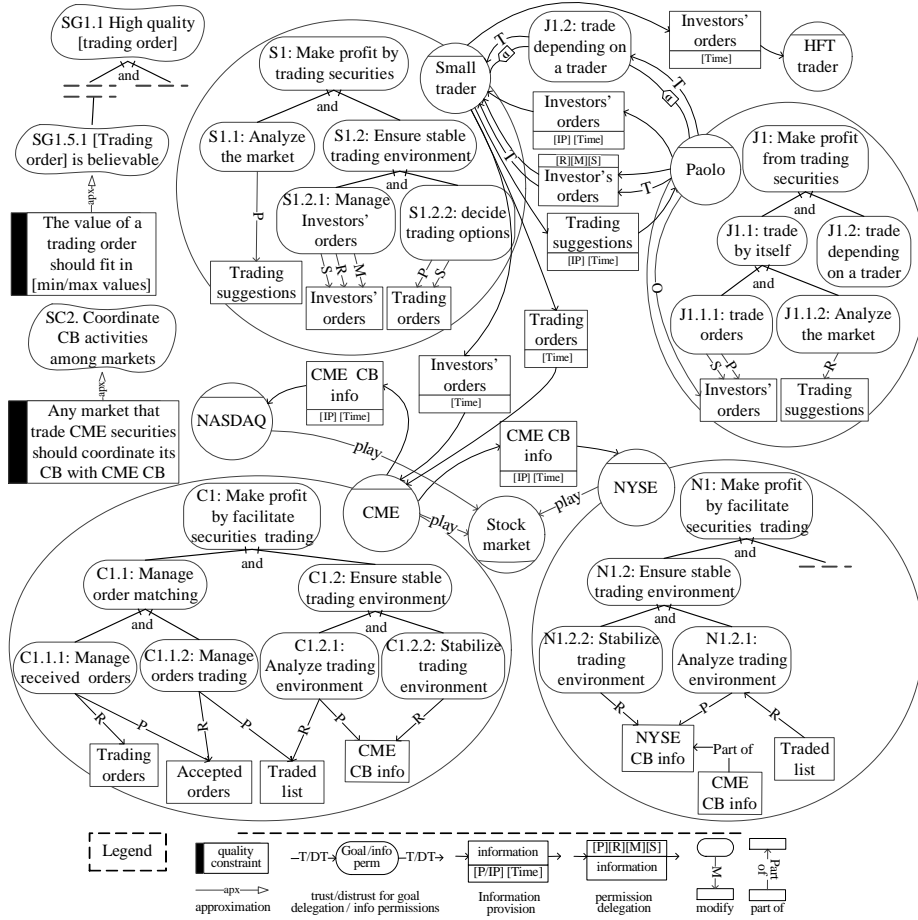


Fig. 3. A partial goal model of the Flash Crash extended with new IQ constructs

provision; and (3) *Accuracy of read information*: can be analyzed based on its *believability* and its *trustworthiness of the provenance*. Figure 3 shows an example of the graphical representation of the modeling concepts introduced above. A second set of modeling extensions are presented below to capture the rationale of IQ requirements.

**(II) High-level IQ concepts:** enable for capturing the stakeholders' high-level IQ requirements, and then gradually refining them until reaching their operational specifications.

*Top-level IQ softgoals:* are softgoal concerning IQ requirements, and they are used as a starting point for identifying the stakeholders' needs concerning information they use/own. For example, in Figure 1 we have two softgoals of the stock market **SG1.1: high quality [Trading order]** that is defined based on information usage, and we have **SG1.2: Coordinate CB activities among markets** that is defined based on information owner needs. In particular, in stock



market domain the same security can be traded in different markets, but it will always have only one primary listing market that is the main market for trading the security, such market has full authority over the way its listed security can be traded in other markets. In our motivating example, CME is the primary listing market and it requires that all CB activities related to its securities to be coordinated with its own activities, i.e., if CME turn to slow trading mode or stop trading, all markets trade the same security should do the same. However, at this point IQ softgoals are likely to be informal and imprecise, but they became more precise during the latter refinement activities.

*And-decomposition for IQ softgoals refinement:* a softgoal can be refined into more specific sub softgoals, if the joint satisfaction of these softgoals is considered equivalent to the satisfaction of the refined softgoal [23]. Usually, softgoals refinement into more specific sub softgoals can be done based on some related taxonomy. For example, Mylopoulos et al. [24] propose taxonomy for refining accuracy and performance softgoals. While Antón et al. [25] introduce taxonomy for refining privacy softgoals. The same can be applied to IQ softgoals, i.e., they can be refined based on different IQ dimensions (taxonomy proposed in Section §3). In particular, we introduce *and-decomposition* relation between an IQ softgoal and its sub IQ softgoals instead of *contribution links* proposed in [24], since such requirements will reach a point that enables for clearly deciding whether they can be achieved or not (operational specifications). For example, the IQ softgoal of stock market (SG1.1: **high quality** [Trading order]) can be *and-decomposed* into softgoals concerning accessibility, accuracy, completeness, timeliness, and consistency etc. based on the actual needs of the stock market. Note that IQ softgoals cannot be refined more than the leaf IQ softgoals, which are used to represent leaf IQ dimensions (shown in Figure 2), where a leaf IQ dimension is an IQ dimension that does not have sub dimensions (e.g., leaf IQ believability softgoal is used to represent believability (leaf IQ dimension)).

*Approximating leaf IQ Softgoals:* as previously mentioned, softgoals are difficult to be expressed in a measurable way. Yet Jureta et al. [23] introduce the *approximation* relation through which a softgoal can be satisfied by a Quality Constraint (QC), i.e., 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. To tackle this problem, we rely on Glinz [26] work<sup>1</sup>, to get better understanding of the nature/type of the leaf IQ softgoals, and to define the appropriate Information Quality Constraints (IQC)<sup>2</sup> for their approximation. Moreover, for the approximation to be consistent with the different types of leaf IQ softgoals, we define 3 different types of IQCs:

1. *Operational IQC:* are constraints that define the required actions to be performed in already determined situations. For example, IQ softgoal concerning

<sup>1</sup> Glinz classify requirements based on their kind, satisfaction and representation

<sup>2</sup> We use IQC to refer to QC, since no other type of constraints is 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

information *believability* can be approximated into operational IQC that define a min and max values to determine the believability of produced/ read information.

2. *Declarative IQC*: are constraints used to define properties of the system that should hold. For example, IQ softgoal concerning the *trustworthiness of provision* can be approximated into declarative IQC stated that information should be transferred only through IP provision.
3. *Quantitative IQC*: are constraints used to specify properties of the system that should hold, and can be measured on an ordinal scale. For example, IQ softgoal concerning *consistency* can be approximated into quantitative IQC stated that *interdependent readers* should rely on information that has the same currency (age).

Table 1 shows how leaf IQ softgoals can be classified, and how they can be 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 [23], where a quality space can be defined as a certain conceptual space that can be used to describe the quality value [27]. The main purpose of the quality space is providing a general consensus among the stakeholders of the system on how quality aspects (e.g., IQ dimensions) can be measured, which removes any ambiguity related to the verification of IQCs, i.e., determining whether a certain IQC is satisfied or not. For instance, both information timeliness and consistency are time related aspects. Thus, how time can be 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, milliseconds, etc.).

Our framework is equipped with an engineering methodological process that is based on the extended secure Tropos methodology [9], and extends it with the required activities to accommodate the new extensions. The process provides the required guidance to designers during the different phases of the system design, and it consists of 6 main steps: (1) *Actors modeling*: in which the stakeholders of the system are identified and modeled along with their top level goals, and then these goals might be refined through And/ Or-decompositions if required. Finally, based on the actors' capabilities, they may delegate goals to one another; (2) *Information modeling*: the different relations among goals and information are modeled, and then information provisions / permissions delegation among actors are modeled as well; (3) *Identifying top IQ softgoals*: IQ softgoals are

defined by the stakeholders based on their needs concerning information they use/own, and then they are refined through AND-decomposition until reaching their leaf IQ softgoals; (4) *Leaf IQ softgoals approximation*: leaf IQ softgoals are approximated into their corresponding IQC; (5) *Trust modeling*: trust among actors concerning goals/ permissions delegation and information producing are modeled; (6) *Analyzing and refining the model*: at this step the model is analyzed to verify whether all the stakeholders' requirements are achieved or not, if some requirements were not achieved, the analyst tries to find appropriate solutions.

## 5 Reasoning about Information Quality Requirements

We use Datalog [28] to formalize all the concepts along with the related axioms (reasoning rules) that have been introduced in this paper. Further, we define a set of properties of the design that can be used to verify the correctness and consistency of the IQ requirements model, i.e., such properties define constraints that designers should consider during the system design<sup>3</sup>. In particular, our framework offers several reasoning techniques that enable for detecting different design vulnerabilities (e.g., IQ related issues such as accuracy, competence, timeliness and consistency, etc.), and how such vulnerabilities may influence the achievement of the actors' requirements. Further, it enables to detect if social dependencies among actors (e.g., information provision) holds, and whether they satisfy the actors' needs or not. More specifically, the reasoning enables for checking whether stakeholders' IQ requirements are achieved or not, and identify the reason(s) preventing their achievement (if any). Further, it is able to identify which functional requirements is influenced if certain IQ needs were not achieved.

## 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. Hevner et al. [29] classify evaluation methods in design science under five categories: observational, analytical, experimental, testing, and descriptive. Since our framework belongs to the design science area, we evaluated its applicability and effectiveness depending on simulation method (experimental), i.e., execute artifact with artificial data. To this end, we developed a prototype implementation of our framework<sup>4</sup> to test the applicability and effectiveness of our framework 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.

<sup>3</sup> The formalization of the concepts, axioms and properties of the design are omitted due to space limitation

<sup>4</sup> <https://github.com/disi-unitn-RE-IQ/RE-IQ>

**Implementation:** our prototype consist of 3 main parts: (1) a graphical user interface (GUI) developed using Sirius<sup>5</sup>, which enable designers for drawing the requirements model; (2) model-to-text transformation that supports translating of the graphical models into Datalog formal specifications depending on Acceleo<sup>6</sup>; and (3) automated reasoning support (DLV system<sup>7</sup>) takes the Datalog specification that result from translating the graphical model into Datalog along with the reasoning axioms, and helps in verifying the correctness and completeness of the stakeholders' IQ requirements.

**Applicability and effectiveness:** we evaluated our framework by showing its applicability in capturing the IQ requirements along with its effectiveness to detect inconsistencies/ conflicts of the stakeholders' requirements 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. The analysis captured several inconsistencies in the design, including:

Incomplete information: the CB information of both *NYSE* and *NASDAQ* were identified as incomplete information, since they miss *CME* CB information that is considered as *part of* them (sub item), such information were not provided to them.

Inconsistent information: the incompleteness problem can be solved by providing "CME CB information" to both of *NYSE* and *NASDAQ*. However, "CME CB information" is provided to them with two different provision times. According to [30], provision time from *CME* to *NASDAQ* was 13 (ms), while provision time from *CME* to *NYSE* was 14.65 (ms). Thus, we face another problem that is inconsistency among *NYSE* and *NASDAQ* concerning "CME CB information", since they are *interdependent readers*.

Inaccurate information: *CME* market considers information received from *Market Marker* inaccurate information, since no believability check was applied to orders received from it. At the other hand, *CME* considers information received from *HFT trader* as inaccurate information, since no trust in information production holds between *CME* and *HFT trader*.

**Experiments on scalability:** to test the scalability of the reasoning technique, we expanded the model shown partially in Figure 3 by gradually increasing the number of its modeling elements from 97 to 3104 through 6 steps, and investigate the reasoning execution time at each step. The result is shown in Figure 4, 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 8.1, 64-bit.

<sup>5</sup> <https://projects.eclipse.org/projects/modeling.sirius>

<sup>6</sup> <https://projects.eclipse.org/projects/modeling.m2t.acceleo>

<sup>7</sup> <http://www.dlvsystem.com/dlv/>

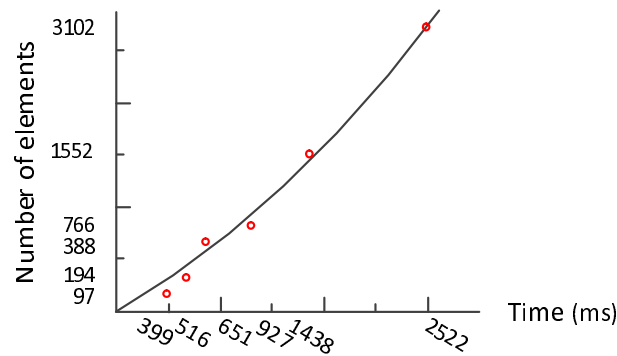


Fig. 4. Scalability results with increasing the number of modeling elements

## 7 Related Work

Requirements engineering community did not appropriately support capturing IQ requirements. For example, UMLsec [2] 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. Abuse frame [31] addresses integrity related issues (modification) by preventing unauthorized actors from modifying information, or prevent authorized actors from doing unauthorized modifications. Finally, secure Tropos [8] seems to be sufficient to capture the functional, security and trust requirements of system, yet it provides no primitives for capturing IQ needs.

At the other hand, we find several approaches for improving IQ by design, but they 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. For instance, Ballou et al. [20] 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. While Shankaranarayanan et al. [32] extend Ballou's work to develop a formal modeling method for creating an IP-MAP. While Scannapieco et al. [33] propose IP-UML approach that relies on the IP-MAP framework, which is a software engineering approach developed to improve information quality in a single organization.

## 8 Conclusion and Future Work

In this paper, we discussed the importance of capturing IQ requirements of the system starting from the early design phase. Moreover, we introduced a novel RE framework that proposes an extended language for modeling and reasoning about IQ requirements, and offers a systematic process for refining these requirements until reaching their operational specifications. We illustrated our framework by example concerning a U.S stock market crash, and showed how

many of the reasons that led to the crash could be avoided if the IQ requirements of the system were addressed properly during the system design. For the future work, we aim providing more expressive analysis for IQ related aspects rather than the binary one, which use only two values to evaluate IQ related concepts (e.g., accurate or inaccurate, believable or not believable, etc. ). Another track under investigation involves extending the relations between stakeholders and information they use, and enriching the proposed IQ model by considering other dimensions along with refining the already considered ones.

## References

1. Redman, T.: Improve data quality for competitive advantage. *Sloan Management Review* **36** (1995) 99–99
2. Jürjens, J.: *Secure systems development with UML*. Springer-Verlag New York Incorporated (2005)
3. Yu, E.S.K.: *Modelling strategic relationships for process reengineering*. PhD thesis, University of Toronto (1995)
4. Chung, L., do Prado Leite, J.: On non-functional requirements in software engineering. *Conceptual modeling: Foundations and applications* (2009) 363–379
5. Emery, F., Trist, E.: *Socio-technical systems. management sciences, models and techniques*. churchman cw et al (1960)
6. Fisher, C., Kingma, B.: Criticality of data quality as exemplified in two disasters. *Information & Management* **39**(2) (2001) 109–116
7. 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
8. 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
9. Gharib, M., Giorgini, P.: Modeling and reasoning about information quality requirements. In: *Requirements Engineering: Foundation for Software Quality*. Springer (2015) 49–64
10. 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)
11. 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)
12. Pipino, L.L., Lee, Y.W., Wang, R.Y.: Data quality assessment. *Communications of the ACM* **45**(4) (2002) 211–218
13. 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
14. Jiang, L.: *DATA QUALITY BY DESIGN: AGOAL-ORIENTED APPROACH*. PhD thesis, University of Toronto (2010)
15. Wang, R.Y., Reddy, M.P., Kon, H.B.: Toward quality data: An attribute-based approach. *Decision Support Systems* **13**(3) (1995) 349–372
16. Liu, L., Chi, L.: Evolutional data quality: A theory-specific view. In: *IQ*. (2002) 292–304

17. 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
18. Wang, R., Strong, D.: Beyond accuracy: What data quality means to data consumers. *Journal of management information systems* (1996) 5–33
19. Subrahmanyam, A.: Algorithmic trading, the flash crash, and coordinated circuit breakers. *Borsa Istanbul Review* **13**(3) (2013) 4–9
20. 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
21. Gharib, M., Giorgini, P.: Detecting conflicts in information quality requirements: the may 6, 2010 flash crash. Technical report, Università degli studi di Trento (2014)
22. Gharib, M., Giorgini, P.: Analysing information integrity requirements in safety critical systems. *The 3rd International Workshop on Information Systems Security Engineering WISSE13*. (2013)
23. 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
24. Mylopoulos, J., Chung, L., Nixon, B.: Representing and using nonfunctional requirements: A process-oriented approach. *IEEE Transactions on Software Engineering* (1992) 483–497
25. Antón, A.I., Earp, J.B., Reese, A.: Analyzing website privacy requirements using a privacy goal taxonomy. In: *Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on, IEEE* (2002) 23–31
26. Glinz, M.: Rethinking the notion of non-functional requirements. In: *Proc. Third World Congress for Software Quality. Volume 2.* (2005) 55–64
27. 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)
28. Abiteboul, S., Hull, R., Vianu, V.: *Foundations of databases*. Citeseer (1995)
29. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS quarterly* **28**(1) (2004) 75–105
30. Lewis, M.: *Flash boys: a Wall Street revolt*. WW Norton & Company (2014)
31. Lin, L., Nuseibeh, B., Ince, D., Jackson, M., Moffett, J.: Introducing abuse frames for analysing security requirements. In: *Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International, IEEE* (2003) 371–372
32. 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
33. 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