Modeling and Reasoning about Information Quality Requirements in Business Processes

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 successful and efficient execution of any Business Process (BP). Despite this, most existing BP approaches either ignore IQ needs, or they deal with them as mere technical issues, without considering the social and organizational aspects that underlie such needs. In this paper, we propose a goal-oriented approach to capture IQ requirements (needs) and map these requirements into workflow net (WFA-net) that is a formal language for modeling and analyzing IQ requirements in BP. We illustrate our approach with an example concerning a stock market system.

Key words: Information Quality, Business process, Workflow, Requirements engineering

1 Introduction

A Business Process (BP) can be defined as a set of activities that has a clear structure describing their sequencing order and dependencies [1]. Traditionally, the BP literature has focused on a control-flow (activity flow) perspective of the process with less emphasis on information perspective. However, information related problems can be the main reason for different kinds of errors in BP [2]. Yet, in recent years some efforts have been devoted to information-aware process design (e.g., Sadiq et al. [3]; Sidorova et al. [4]; Trcka et al. [2]). However, the focus of attention in these works is combining information flow with activity flow, i.e., they are able to detect when an activity in BP rely on information that does not exist, but they say nothing about Information Quality (IQ) concerns.

IQ is a key success factor for most BPs since depending on low-quality information may result in undesirable outcome [5], or it might even prevent the BP from achieving its goals. In the literature, we can find several techniques for dealing with IQ (e.g., preventing, detecting and correcting IQ related issues [6]). Yet most of these techniques propose solutions that are able to address the technical aspects of IQ, and seem to be limited in addressing the social and organizational IQ related aspects. Such aspects are particularly important for BPs since they are mainly executed by social actors and not only machines [7]. More specifically, most BPs occur in a social context (e.g., socio-technical systems [8]), where humans and technical components are considered as an integral part of the BP. Thus, understanding the social/ organizational context where the BPs are executed is essential to detect different kinds of vulnerabilities.

For example, Fisher and Kingma [9] showed how existing IQ technical are not able to capture IQ needs in their social and organizational context, where different kinds of vulnerabilities might manifest themselves in the actors' interactions and dependencies. The Flash Crash (a main U.S market crash) is an example where the problem was not caused by a mere technical failure, but it was also due to several socio-technical IQ related vulnerabilities of the system [10]. This introduces the need of analyzing the social and organizational environment where the BP operates [11]. In this paper, we propose a goal-oriented approach for capturing IQ requirements of the social and organizational context where the BP is executed, and then introduce mechanisms for mapping these requirements into workflow net with actors (WFA-net). The paper is organized as follows; Section (§2) describes the research baseline, an example concerning a stock market system is presented in section (§3). We propose our approach in section (§4). The prototype is summarized in (§5). Related work is presented in Section (§6), we conclude and discuss the future work in Section (§7).

2 Research baseline

Our research baseline is based on three main areas; we briefly discuss each of them as follows:

(i) Goal-Oriented Requirements Engineering (GORE): several approaches that adopts GORE paradigm have been proposed in the literature (e.g., KAOS [12], *i** [13], secure Tropos [14]). Among the existing ones, we adopt an extended version of secure Tropos [15] as a baseline for our approach, which supports the basic modeling concepts offered by secure Tropos, and provides concepts for capturing IQ requirements. In particular, it introduces primitives for modeling actors of the system, which covers two concepts, a role, and agent. Goals that are used to represent actors' strategic interests, and they can be refined through AND/OR decomposition into finer sub-goals. While information is used to represent any informational entities, and it has volatility attribute that can be used to determine its timeliness (validity). An actor can be a legal owner of information item, which gave it a full control over its use. While goals may produce, need and send information. Finally, it adopts the notion of delegation to model the transfer of entitlements among actors, and it adopts the notion of trust and distrust to capture the actors' expectations of one another concerning their delegated entitlements and authorities.

(ii) Information Quality: IQ is a hierarchical multi-dimensional concept [16, 17], that can be characterized by several dimensions [18, 17], including: accuracy, completeness, timeliness, accessibility, trustworthiness, etc., where each of these dimensions can be used to represent a certain aspect of IQ. We focus on 3 main IQ dimensions that enable us to address the IQ related issues that we consider in this paper, namely: *Accessibility*: the extent to which information is available, or easily and quickly retrieved [16], we limit accessibility definition to information availability and having the required permission to perform a task at hand; *Accuracy*: means that information should be true or error free with respect to some known values [17]; *Timeliness*: can be defined as to which extent information is valid in term of time, i.e., sufficiently up-to-date [16].

(iii) Petri nets/ WF-nets/ WFD-nets: several workflow modeling languages have been proposed, yet we focus on Petri-nets-based languages (e.g., Petri nets, WF-nets, and WFD-nets). In particular, a petri net [19]is a graphical and formal language that can be used to model different kinds of BPs. Formally: A Petri net N = $\langle P, T, F \rangle$, where P is a finite set of places, T is a finite set of transitions, and $F \subseteq (PXT) \cup (TXP)$ is a set of arcs (flow relation). At any time, a place contains zero or more tokens, while a transition $t \in T$ is said to be *enabled*, **iff**, each input place p of t contains at least one token. An *enabled* transition t may *fire*, **iff**, a transition t consumes one token from each input place p of t, and it produces one token in each output place p of t. Furthermore, a marking of a Petri net is a multi-set of its places $M : P \longrightarrow \mathbb{N}$. Transitions are the active components in a Petri net, i.e., they change the state of the net. For example, given a Petri net N and a marking M_1, we say that $M_1 \xrightarrow{t} M_2$: if transition t is *enabled* at marking M_1, and firing t at M_1 results in M_2. While $M_1 \xrightarrow{\sigma} M_n$: $\sigma = t_1, t_2, \ldots, t_{n-1}$ is a firing sequence leading from M_1 to M_n. Finally, we say that a marking M_n is *reachable* from M_1, **iff**, there is a firing sequence $\sigma = t_1, t_2, \ldots, t_{n-1}$ such that $M_1 \xrightarrow{\sigma} M_n$. At the other hand, a workflow-net (WF-net) [20] is a Petri net with well-defined starting point (**start**) and a well-defined ending point (**end**), and every node (place or transition) is on a path from **start** to **end**, and transitions in a WF-net are called *tasks*. While workflow net with data (WFD-net) [4] is a workflow net with data elements in which tasks can read, write, or delete data elements. Moreover, a task can also have data dependent guards that block its execution when it is evaluated to false.

3 US Stock Market System

A stock market (equity market) system is the aggregation of investors, traders, trading markets, along with several firms that provide different kinds of financial services. Based on [21, 22], we can identify the main stakeholders of a stock market system, including: *stock investors* are individuals or companies, who have a main goal of making a 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*. *Stock markets* are places where *traders* gather and trade securities (e.g., NYSE, CME, NASDAQ, etc.). In particular, markets make a profit by facilitating security trading among traders, i.e., they receive, match, and perform trades from different *traders*. Moreover, they should guarantee a fair and stable trading environment for their traders.

Accounting firms can be defined as firms that provide accounting services to companies for a fee. While *auditing firms* are responsible for auditing the financial statements of legal entities (e.g., persons, companies), where a financial statement is a formal record of the financial activities of such entities. *Consulting firms* provide professional advice concerning financial securities for a fee to *traders* and *investors*. *Credit assessment rating firms* are specialized for providing assessments of the creditworthiness of companies' securities, i.e., they help traders in deciding how risky it is to invest money in a certain security.

4 Approach for Modeling and Reasoning about IQ Requirements in Business Process

In this section, we propose our approach for modeling and analyzing IQ requirements in BP. An overview of the methodological process that underlies our approach is shown in Figure 1, the process is composed of 3 main phases[1]:

(1) **Modeling phase**: in which we model the overall system, where the BP occur, i.e., the social, organizational and IQ requirements of the system where the BP occur are identified and modeled; this phase is composed of 5 main steps: (1.1) *Actors modeling*: aims to model the actors of the system in terms of agents and the role(s) the play; (1.2) *Goals modeling*: identify and model actors' top-level goals and refine them, if needed, through And/ Or-decomposition into leaf goals; (1.3) *Information modeling*: identify and model the legal owners of information items, which is essential to identify who has full control concerning information permissions. Moreover, we model the different relations between goals and information they use (e.g., produces, reads, modifies and sends); (1.4) *Social dependency modeling*: in which we model actors' dependencies for information provision, and the delegation of both authorities and entitlements, i.e., based on actors' capabilities some goals might be delegated to actors, who have the capabilities to achieve them; and based on actors' needs, information and permissions are provided/ delegated respectively. (1.5) *Trust modeling*: model trust/ distrust among actors concerning goals/ permissions delegation, based on their expectations in one another. When the modeling phase is complete, and if the model does not require any refinements, we proceed to the mapping phase.

(2) **Mapping phase**: in which, we map the requirements model, that has been produced in the previous phase, into workflow net with actors (WFA-net) that is a formal language we propose for modeling and analyzing BP control-flow, information flow, and IQ requirements.

(3) **Analysis phase**: aims to verify the correctness and consistency of the BP model. In particular, we define a set of properties to check the correctness and consistency of the BP control-flow, information flow along with IQ requirements, i.e., BP is correct and consistent, if all of these properties hold.

4.1 Modeling Phase

In order to model IQ requirements of the BP in their social and organizational context, we rely on an extended version of secure Tropos [15], which

[1] We discuss each of these phases in details in sections 4.1, 4.2, and 4.3 respectively

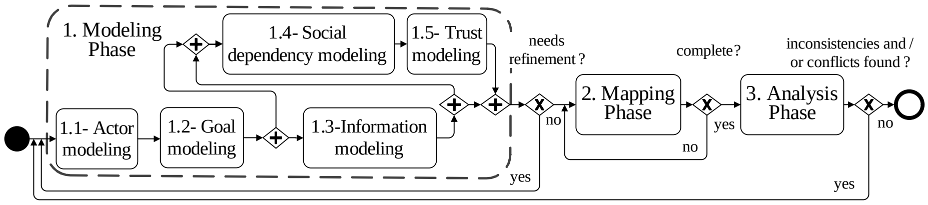

Fig. 1. The process for modeling and reasoning about IQ requirements in BP

provides concepts for capturing IQ requirements. Figure 2 shows a portion of a goal model concerning the stock market system represented in the extended secure Tropos modeling language to clarify its main concepts. For instance, `John` is an agent that plays `stock investor` role, which has a main goal of `G1. Make profit from trading securities` that is Or-decomposed into `G1.1 Trade securities by itself` and `G1.2 Delegate trading activities to a trader`. In the case of Or-decomposition, the parent goal is achieved, if any of its sub-goals is achieved, i.e., the goal `G1.` is achieved, if `G1.1` or `G1.2` is achieved. Moreover, `G1.1.1 Trade securities by itself` is And-decomposed into `G1.1.1.1 Produce and send orders` and `G1.1.1.2 Finalize the trade`. In the case of And-decomposition, the parent goal is achieved only if all its sub-goals are achieved.

Moreover, the goal `G1.1.1.1 (P)roduces` and `(S)ends investor's orders`, and the goal `G1.1.1.2 (R)eads trade settlement`. Each of the previously mentioned relations between goals and information are required for the achievement of the goals, i.e., if a goal could not use (e.g., reads, sends, or produces) information as intend, it will not be achieved (it will be prevented). Moreover, the `stock investor` provides (Integrity Provision (IP)[2]) `investor's orders` to `trader`, and it delegates the goal *G1.2 Delegate trading to a trader* to *stock trader*, and trust it for its achievement. Extended secure Tropos does not support modeling of permission, but secure Tropos does. Thus, we refine the modeling language by proposing 4 different types of permissions concerning the 4 types of information usage (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. For example, the `stock investor` is the `owner` of `investor's orders`, and it delegates (R)eads, (M)odifies and (S)ends permissions concerning it to the `trader`.

4.2 Mapping phase

In this section, we propose a workflow net with actors (WFA-net) that is a formal language, we propose, for modeling and analyzing IQ requirements for BP. In particular, WFA-net is able to model and analyze the control-flow, information flow, and IQ requirements of BP. Moreover, we discuss the mechanisms that are used for mapping IQ requirements model into WFA-net.

[2] IP provision preserves the integrity of the provided (transferred) information [23]

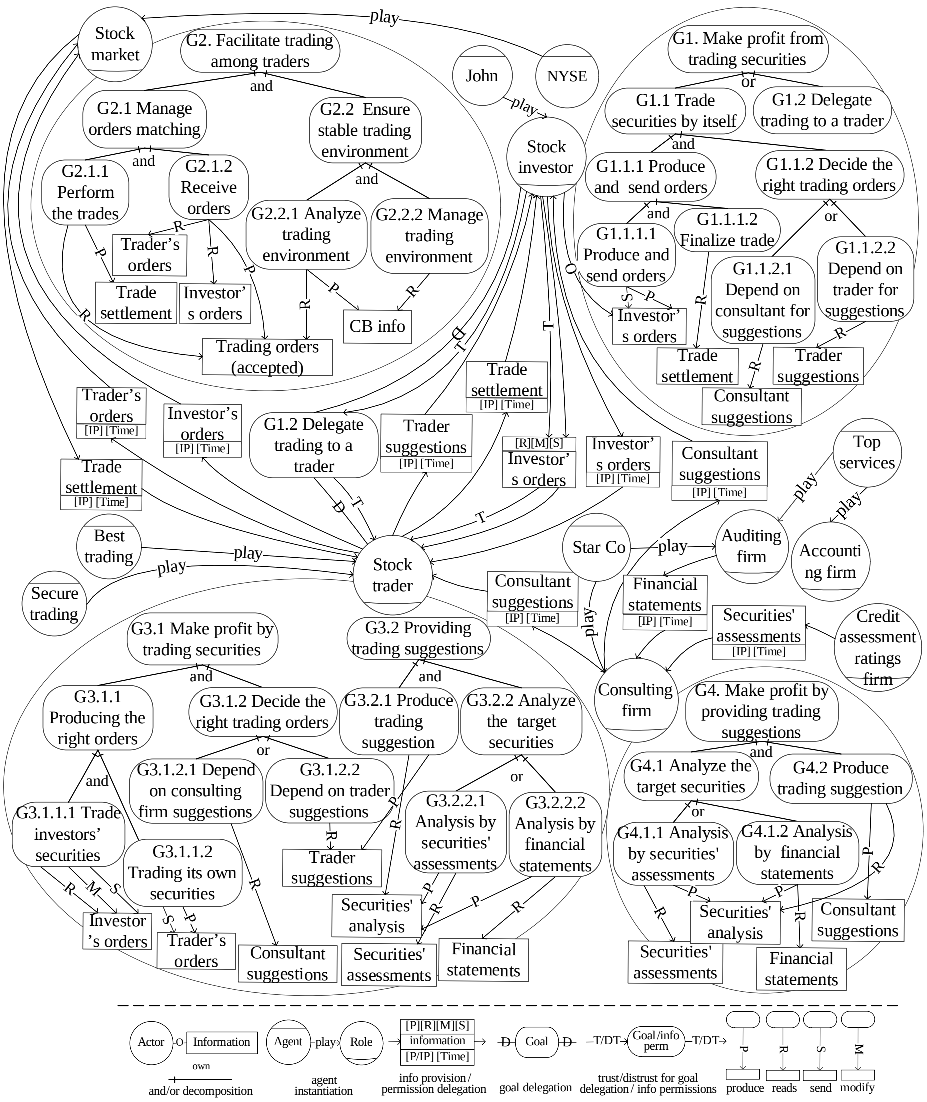

Fig. 2. A partial goal model concerning the stock market structure

Work flow net with actors (WFA-net) a workflow net with actors (WFA-net) adopts workflow net (WF-net) and extends it with the notion of social actor, and IQ related concerns. In WFA-net, activities (tasks) are assigned to social actors, and they may produce, read, modify, and send information items. In what follows, we define the semantics of WFA-nets. Let us consider a finite set of social actors $A = \{a_1, a_2, \ldots, a_n\}$, a finite set of information elements $I = \{i_1, i_2, \ldots, i_m\}$, and a finite set of time intervals $T = \{t_1, t_2, \ldots, t_m\}$, and we define $I_v \subseteq \{I \text{ X } T\}$ to describe information items along with their volatility values. Moreover, to capture information send relation, we define $S \subseteq \{A \text{ X } I \text{ X } T\}$ that describe the target (actor) of send information, information to be send, and the required send time. Further, we define a set of responsibility predicates $\mathit{\Pi}_A = \{\pi_{a_1}, \pi_{a_2}, \ldots, \pi_{a_k}\}$ to capture the relation between actors and activities they are responsible of (e.g., responsible(actor); a set of produces predicates $\mathit{\Pi}_P = \{\pi_{p_1}, \pi_{p_2}, \ldots, \pi_{p_j}\}$ to capture the relation between activities and information they read (e.g., produces(info)); a set of read predicates $\mathit{\Pi}_R = \{\pi_{r_1}, \pi_{r_2}, \ldots, \pi_{r_k}\}$ to capture the relation between activities and information they read (e.g., read(info)); a set of modify predicates $\mathit{\Pi}_M = \{\pi_{m_1}, \pi_{m_2}, \ldots, \pi_{m_l}\}$ to capture the relation between activities and information they modify (e.g., modify(info)); a set of send predicates $\mathit{\Pi}_S = \{\pi_{s_1}, \pi_{s_2}, \ldots, \pi_{s_l}\}$ to capture the relation between activities and information they send (e.g., send(actor, info, send_time)).

Furthermore, we define the following functions: $f_{\pi_a} = \mathit{\Pi}_A \longrightarrow \{A\}$, responsibility function that assign responsibility predicates with actors responsible of achieving the related activities; function $f_{\pi_p} = \mathit{\Pi}_P \longrightarrow 2^{I_v}$, production function that assign produce predicates with information items that activities produce; function $f_{\pi_r} = \mathit{\Pi}_R \longrightarrow 2^{I_v}$, reading function that assign read predicates with information items that activities read; function $f_{\pi_m} = \mathit{\Pi}_M \longrightarrow 2^{I_v}$, modify function that assign modify predicates with information items that activities modify; function $f_{\pi_s} = \mathit{\Pi}_S \longrightarrow 2^S$, send function that assign send predicates with information items that activities send.

Now, we define a WFA-net as a WF-net where every transition t is described with: an actor being assigned to perform the activity (res); a set of information items being produced (pd) by the activity when t fires; a set of information items being modified (md) by the activity when t fires; a set of information items being read (rd) by the activity when t fires; and a set of information items being send (sd) by the activity when t fires.

Definition 1 (WFA-net). *A workflow net with actors (WFA-net) $N = \langle P, T, F, res, pd, rd, md, sd \rangle$ consist of WF-net $N = \langle P, T, F \rangle$, an actor assigning function res: $T \longrightarrow A$, information producing function pd : $T \longrightarrow 2^{I_v}$, information reading function rd : $T \longrightarrow 2^{I_v}$, information modifying function md : $T \longrightarrow 2^{I_v}$, and information sending function sd: $T \longrightarrow 2^S$.*

Example 1. A WFA-net of a stock investor for trading securities is shown in Figure 3. Its actor set A = {`credit_firm`, `audit_firm`, `trader`, `investor`, `consulting_firm`, `stock market`}, and its information set I= {`securities_assessment`, `financial_statement`, `trader_suggestion`, `trading_orders`, `consultant _suggestion` , `investors_orders`, `trading_settlement`}. Considering the transition `Produce and send orders`, the responsibility function `res` (Produce and send orders) = {investor}, the production function `pd`(Produce and send orders) = {investor's order}, the sending function `sd`(Produce and send orders) = {(stock market, investor's order, time)}, the modify/ read functions `md`(Produce and send orders) = `rd`(Produce and send orders) = $\{\emptyset\}$.

To capture the work flow in WFA-net, we should be able to evaluate the activities related predicates either to true ($\top$) or to false ($\bot$) based on an already

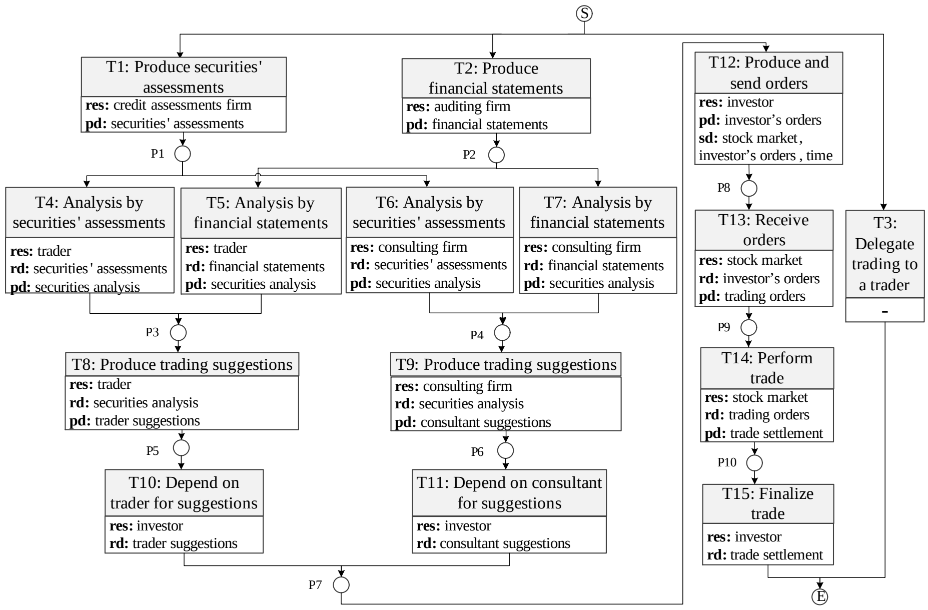

Fig. 3. A WFA-net of a stock investor for trading securities

defined criteria. Thus, we define the following functions, $\sigma_{\pi_a}\colon \varPi_A \longrightarrow \{\top, \bot\}$ assigns to each responsibility predicate either $\top$, when the responsible actor can achieve the activity, or it assigns $\bot$, when the responsible actor cannot achieve the activity. Similarly, we define $\sigma_{\pi_p}\colon \varPi_P \longrightarrow \{\top, \bot\}$, $\sigma_{\pi_r}\colon \varPi_R \longrightarrow \{\top, \bot\}$, $\sigma_{\pi_m}\colon \varPi_M \longrightarrow \{\top, \bot\}$, and $\sigma_{\pi_s}\colon \varPi_S \longrightarrow \{\top, \bot\}$ that assign to each produce/ read/ modify/ send predicate either $\top$, when information item i can be produce/ read/ modify/ send by the activity, or it assigns $\bot$ otherwise.

Finally, we define $\sigma_{\varPi}$ function that sums the values of the previously mentioned functions over their related predicates. $\sigma_{\varPi}\colon T \rightarrow (\top,\ \bot)$, where $\sigma_{\varPi} = \sigma_{\pi_a} \wedge \sigma_{\pi_p} \wedge \sigma_{\pi_r} \wedge \sigma_{\pi_m} \wedge \sigma_{\pi_s}$. Following WFD-net, we refer to a **state**[3] of WFA-net as a configuration, where a WFA-net configuration is a state that includes responsible actors along with the produce, read, modify, and send information that the activity perform. Moreover, a **state** can be represented as $\sigma_{\varPi}$, and the set of all **states** is denoted by $\varSigma$.

Definition 2 (Configuration). *Let $N = \langle\ P,\ T,\ F,\ res,\ pd,\ rd,\ md,\ sd\ \rangle$ be a WFA-net, let m be a* ***marking*** *of N, and let $\sigma_{\varPi}$ be as defined above. Then, $c = \langle m, \sigma_{\varPi}\rangle$ is a configuration of N. With $\varXi$ we denote the set of all configurations, and the start configuration of N is defined by $\langle [start],\ \sigma_{\varPi}\rangle,\ (\ I_v\ = \emptyset)\ \rangle$. While $C_e\ = \{\langle [end],\ \sigma_{\varPi}\rangle \mid \sigma_{\varPi} \in \varSigma\ \}$ defines the set of final configurations.*

In the initial configuration, only one place is marked [start], and the I_v set is initialized to the empty set. While a configuration is a final configuration, if

[3] A state (also called marking) of a Petri net is a distribution of tokens over its places

it contains a marking [end]. A transition t of a WFA-net N can be enabled at a configuration c $= \langle m, \sigma_{\Pi} \rangle$, IFF: (1) the transition t is enabled at marking m (activity flow), and (2) the predicates related to configuration c enabling (σ_{Π}) must be true, which guarantees that IQ requirements are met. When a transition t is enabled, it may fires, where firing of transition t changes the marking, as well as the related predicates, and information set I_v, i.e., the firing of t enable a set of successor configurations $\langle \mathrm{m}^{'}, \sigma_{\Pi}^{'} \rangle$, and changes predicates and information.

Definition 3 (Firing a transition for WFA-nets). *Let N =* $\langle$ *P, T, F, do, pd, md, nd, up* $\rangle$ *be a WFA-net. A transition* $t \in T$ *of N is enabled at a configuration* $c = \langle m, \sigma_{\Pi} \rangle$ *of N, if* $m \xrightarrow{t}$*,* σ_{Π} *is assigned true* ($\top$)*. Firing t enables a set of configurations* $C \subseteq \Xi$*, with* $C = \{ \langle \mathrm{m}^{'}, \sigma_{\Pi}{}^{'} \rangle \mid m \xrightarrow{t} m^{'} \wedge (\forall\, i \in pd(t) = \top \colon I = i \cap I) \}$ *and it is denoted by* $c \xrightarrow{t} C$*.*

Example 2. Consider transition `Receive orders` in Figure 3, and suppose there is a token in place `p8`. The transition is enabled if *stock market* (responsible actor) has the capability to achieve such activity, information `investors_orders` fits for read from the perspective of the *stock market*, i.e., information has been already produced, *stock market* has it, and it does not suffer from any IQ related issue (e.g., information is valid (timeliness), accurate, *stock market* has the read permissions over it, etc.). Firing this transition means that the token in `p8` is removed, and a token is produced in `p9`. Moreover, information `trading_orders` is produced by the *stock market.* Note that we only consider information producing when a transition fires, since it affects all the other information related operations (e.g., sends, modifies). While other IQ aspects can be captured with the help of the automated reasoning support (discussed in section 4.3).

Mapping IQ Requirements from the Goal model into WFA-nets in this section, we describe how IQ requirements model can be mapped into WFA-net. In particular, we define rules for identifying complete building blocks that are used to represent the extended secure Tropos constructs, which can be mapped into WFA-net activities. Moreover, we define several sets of constraints that should be followed during the mapping process to ensure the correctness of the mapping and the resulting WFA-net:

Building blocks: we define 3 rules for identifying building blocks: (a) a goal that is not And/ Or-decomposed of any other goal, and it is not composed into sub-goals as well, can be considered as a complete building, and it can be mapped into a WFA-net activity (task) taking into consideration the actor, who is responsible for its achievement, and information it relies on (if any); (b) goals that are And-decomposed from a parent goal are considered as a complete building block, and they can be mapped into a sequence of WFA-net activities, where each of these activities represents a sub-goal; (c) goals that are Or-decomposed from a parent goal are considered as a complete building block, and they can be mapped into parallel (alternatives) WFA-net activities, where each of these activities represents a sub-goal.

Consistency constraints we define 3 consistency constraints that can be used to ensure a correct mapping between the identified building blocks and WFA-net activities: (i) mapping is allowed for complete building blocks only, i.e., no goal is allowed to be mapped unless it can be considered as a complete building block; (ii) if the WFA-net is used to model a plan to achieve a top-level goal, the full plan to achieve the top-level goal should be considered in the WFA-net, and (iii) no information is allowed in the WFA-net unless its source (the goal that produces it) exist in the WFA-net.

Sequencing constraints we define 3 sequencing constraints that can be used to ensure the proper ordering of the WFA-net activities: (i) activities in WFA-net should be consistent with their sequencing order in their own building blocks; (ii) if an activity depend on another activity, it should appear after the activity it depends on in the WFA-net; (iii) if an activity depend on the outcome of another activity (e.g., information), it is desirable to appear after the activity it depends on its outcome.

Refinement constraints are constrains derived from the WFA-net semantics, and used to refine the WFA-net that results from the sequencing phase. We define two simple *refinement constraints*: (i) no two places (p_1, p_2) can appear in sequence without a transition (t) separating them; and (ii) no two transitions (t_1, t_2) can appear in sequence without a place (p) separating them.

Example 3. In this example, we show how investor process for trading securities shown in Figure 2 can be mapped into WFA-net shown in Figure 3. The investor aims to achieve the top-level goal `G1`, but it cannot be considered as a building block since it is or-decomposed into `G1.1` and `G1.2`. Thus, instead of `G1` we have `G1.1` and `G1.2` that can be mapped as parallel activities into WFA-net. `G1.2` can be mapped into `T3` activity, while `G1.1` is and-decomposed into `G1.1.1` and `G1.1.2`, which can be mapped into two sequential transactions. Moreover, `G1.1.1` is also and-decomposed into `G1.1.1.1` and `G1.1.1.2`, which can be mapped into two sequential transactions `T12` and `T15` respectively. While `G1.1.2` is or-decomposed into `G1.1.2.1` and `G1.1.2.2`, and they can be mapped into two parallel transactions `T11` and `T10` respectively.

Furthermore, transaction `T10` needs to read trader suggestion information, and transaction `T11` needs to read consultant suggestion information. Since no information is allowed to exist without its source, we add `T8` and `T9` transactions that produce trader suggestion and consultant suggestion respectively. However, `T8` requires to read securities analysis that is produced by the trader, which can be produced either by `T4` or `T5`. Similarly, `T9` requires to read securities analysis that is produced by a consultant, which can be produced either by `T6` or `T7`.

Moreover, `T4` `T6` needs to read securities assessment. Thus, `T1` is added since it is responsible for producing such information. Similarly, `T5` `T7` needs to read financial statement information. Thus, `T2` is added since it is responsible for producing such information. At the other hand, `T15` needs to read trade settlement information, thus, `T13-14` are also added. Finally, following the refinement constraints, we add some position between transactions when required.

Table 1. Properties of the design

Pro1	:- `position(end), not reached(end)`
Pro2	:- `read(A,I), information(I, T), not has(A,I, T)`
Pro3	:- `need_perm(P, A, I), not has_perm(P, A, I)`
Pro4	:- `dele_perm(P, A, B, I), not has_perm(P, A, I)`
Pro5	:- `has_perm(P, B, I), owner(A,I), not trust_perm_chain(P, A, B, I)`
Pro6	:- `read(G, I), not fits_read(G, I)`
Pro7	:- `send(T, G, B, I), not fits_send(T, G, B, I)`

4.3 Analysis Phase

In this section, we describe the automated reasoning support that our approach proposes to guarantee the correctness and consistency of information-flow, IQ requirements, and control-flow of a BP. In order to verify the correctness and consistency of BP model, we provide a Datalog [24] formalization of all the concepts that have been introduced in the paper, along with the reasoning axioms[4]. Moreover, we define a set of properties of the design (shown in Table 1) that can be used to verify the correctness and consistency of the BP model; in what follows we discuss each of these properties:

Pro1 states that an end position in BP should be reached. If this property holds, the BP reaches its end configuration, which verifies the correctness of the BP control-flow, information-flow, and IQ requirements. We can rely on this property to quickly verify the correctness and consistency of the BP.

Pro2 states that actors should have all information that is required for the transitions they are responsible of, i.e., it is used to verify information-flow related issues. For instance, consider transition `T4` in Figure 3, if trader did not have security analysis information (it is not created), the analysis will detect such situation and notify the analyst about it.

Pro3-5 are used to verify information permissions related properties. For instance, **Pro3** states that actors should have all permissions they require to achieve the transition they are responsible for. **Pro4** states that actors cannot delegate permissions they do not have. While **Pro5** states that BP should not include any actor who has permissions, and there is no trust/ trust chain between the actor and information owner concerning such permissions. This property enables information owners to guarantee that actors will not misuse permissions concerning information they own. For instance, consider transition `T14` in Figure 3, if the stock market does not have (R)ead permission concerning trading orders, or (P)roduce permission concerning trading settlement, the analysis will detect and notify the analyst about such situation. While if the stock market has (M)odify permission concerning investor's orders, and no trust relation hold between the market and the investor (information owner), the analysis will notify the analyst to solve such situation.

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

Pro6-7 are used to verify IQ related properties in the BP[5]. For instance, **Pro6** states that the model should not include any information that does not fit for the purpose of read (appropriate for read), where information should be accessible, accurate, and valid to be considered appropriate for read. In particular, *Pro6* is able to detect: (1) information accessibility: information is inaccessible to an actor, if the actor does not have read permissions; (2) information accuracy: information is inaccurate, if it is produced with no permissions, or modified intentionally/ intentionally during its transfer; (3) information timeliness (validity), where timeliness can be analyzed depending on information *currency* that is the time interval between information creation and its usage time [16], and information *volatility* that is the change rate of information value [16], i.e., information is not valid, if its currency is bigger than its volatility interval, otherwise it is valid. While **Pro7** states that the model should not include any information that does not fit for send, where information should be accurate, complete and valid at its intended destination to be considered appropriate for send. Information is complete at its destination if it was transferred through IP provision that guarantees its integrity. Finally, information is valid at its destination, if its transmission time is less than the required send time.

5 Prototype Implementation

Our proposed approach belongs to the design area. Thus, it can be evaluated by simulation method (experimental) [25], i.e., developing a prototype and test its applicability with artificial data. To this end, we developed a prototype implementation[6] to test the approach applicability, i.e., test its ability for modeling and analyzing IQ requirements in BPs. In what follows, we briefly describe the prototype and then discuss its applicability over scenarios abstracted from the Flash Crash case study[7].

Prototype implementation: our prototype has been developed depending on Eclipse integrated development environment (IDE), and it consists of 3 parts: (1) A graphical user interface (GUI)[8]: that support designers while designing BPs. In particular, it enable designer to model the overall system where the BP occur, and then map the requirements model into WFA-nets by drag-and-drop modeling elements from palettes; (2) Model-to-text transformation: supports the transformation of the graphical BP model into Datalog formal specifications depending on Acceleo[9]; and (3) automated reasoning support (DLV system[10]) that takes the Datalog specifications as an input, and then perform the required analysis that helps to verifies the correctness and completeness of the BP against the properties of the design.

5 Produce/ Modify related issues can be addressed by permissions properties (Pro3-5)
6 The prototype tool is available at `https://mohamadgharib.wordpress.com/iqbp/`
7 For more information about the case study refer to [26]
8 Developed by Sirius `https://projects.eclipse.org/projects/modeling.sirius`
9 https://projects.eclipse.org/projects/modeling.m2t.acceleo
10 http://www.dlvsystem.com/dlv/

Applicability: we tested the applicability of our approach by applying it to several scenarios abstracted from the Flash Crash case study. In particular, we modeled several BPs, each of them violates one or more of the properties of the design, and we transformed these BP models into Datalog specification, and then we run the automated analysis to test the analysis ability in discovering the violations to the properties of the design. The analysis was able to detect and notify the analyst about all the violation to the properties of the design.

6 Related Work

Traditionally, BP literature has focused on the control-flow perspective of the processes with less emphasis on the information perspective. However, in recent years, some efforts have been devoted to data-aware process design. For example, Trcka et al. [27] introduced data anti-patterns that represent undesirable data-flow behaviors in BPs, while [4] proposed WFD-nets that is able to address data-flow issues along with the control-flow of a BP. Moreover, data-related process analysis methods have also been proposed by [3]. Furthermore, Deutsch et al. [28] propose *TNest* that is a data-centric workflow modeling language, which allows for expressing data dependencies along with time constraints. However, all these approaches do not specifically consider IQ related issues.

At the other hand, combining Goal models and BPs is not new, for example, Cysneiros and Yu [29] discuss agents autonomy in modeling and supporting business processes (BPMN). While Koliadis et al. [11] propose a preliminary work for mapping $i*$ to BPMN. Lapouchnian et al. [30] propose a requirements-driven approach for BP design that uses requirements goal models to capture alternatives in process configuration. Still, to the best of our knowledge, there is no previous work in GORE that considers and map IQ requirements to BPs.

Several approaches for improving IQ by design have been proposed. For instance, Wang [31] proposes the Total Data Quality Management (TDQM) methodology for delivering high-quality information products (IP) to information consumers. Furthermore, Ballou et al. [32] presented an information manufacturing system that can be used to determine the data quality in terms of timeliness, quality, and cost. Moreover, Shankaranarayanan et al. [33] extend Ballou' work, and propose a formal modeling method for creating an IP-MAP.

7 Conclusions and Future Work

In this paper, we discussed the importance of capturing IQ requirements in BPs from the early design phase. Moreover, we introduced a goal-oriented approach to model and analyze IQ requirements in BPs from a socio-technical perspective. In particular, our approach is based on an extended version of secure Tropos that is able to model and analyze IQ requirements in their social and organizational context, and then map these requirements into workflow net with actors (WFA-net). Moreover, we provide detailed execution semantics for the WFA-nets, which enable to capture design flaws related to control-flow, information-flow, and IQ requirements. We illustrated the applicability of our framework by an example concerning a U.S stock market crash. For the future work, we intend to extend the IQ dimensions we considered, and we believe that the different interrelations among IQ dimensions need to be studied in more details. Finally, we aim to better validate our approach by applying it to more complex case studies that belong to different domains.

Acknowledgment

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

References

1. Aguilar-Saven, R.S.: Business process modelling: Review and framework. International Journal of production economics **90**(2) (2004) 129–149
2. Trčka, N., van der Aalst, W., Sidorova, N.: Data-flow anti-patterns: Discovering data-flow errors in workflows. In: Advanced Information Systems Engineering, Springer (2009) 425–439
3. Sadiq, S., Orlowska, M., Sadiq, W., Foulger, C.: Data flow and validation in workflow modelling. In: Proceedings of the 15th Australasian database conference-Volume 27, Australian Computer Society, Inc. (2004) 207–214
4. Sidorova, N., Stahl, C., Trčka, N.: Workflow soundness revisited: checking correctness in the presence of data while staying conceptual. In: Advanced Information Systems Engineering, Springer (2010) 530–544
5. Redman, T.: Improve data quality for competitive advantage. Sloan Management Review **36** (1995) 99–99
6. Hamming, R.: Error detecting and error correcting codes. Bell System technical journal **29**(2) (1950) 147–160
7. Yu, E., Mylopoulos, J.: From ER to "AR modelling strategic actor relationships for business process re-engineering. Entity-Relationship ApproachER'94 Business Modelling and Re-Engineering (1994) 548–565
8. Emery, F., Trist, E.: Socio-technical systems. management sciences, models and techniques. churchman CW et al (1960)
9. Fisher, C., Kingma, B.: Criticality of data quality as exemplified in two disasters. Information & Management **39**(2) (2001) 109–116
10. 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
11. Koliadis, G., Vranesevic, A., Bhuiyan, M., Krishna, A., Ghose, A.: Combined approach for supporting the business process model lifecycle. In: PACIS, Citeseer (2006) 76
12. Dardenne, A., Van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Science of computer programming **20**(1-2) (1993) 3–50
13. Yu, E.S.K.: Modelling strategic relationships for process reengineering. Ph.D. thesis, University of Toronto (1995)

14. 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
15. Gharib, M., Giorgini, P.: Modeling and reasoning about information quality requirements. In: Requirements Engineering: Foundation for Software Quality (REFSQ15), To appear. Springer (2015)
16. Pipino, L.L., Lee, Y.W., Wang, R.Y.: Data quality assessment. Communications of the ACM **45**(4) (2002) 211–218
17. 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
18. Wand, Y., Wang, R.: Anchoring data quality dimensions in ontological foundations. Communications of the ACM **39**(11) (1996) 86–95
19. Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE **77**(4) (1989) 541–580
20. van der Aalst, W.M.: The application of Petri nets to workflow management. Journal of circuits, systems, and computers **8**(01) (1998) 21–66
21. 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)
22. Mishkin, F.S.: Policy remedies for conflicts of interest in the financial system. In: Macroeconomics, Monetary Policy and Financial Stability Conference, Citeseer (2003)
23. Gharib, M., Giorgini, P.: Analysing information integrity requirements in safety critical systems. The 3rd International Workshop on Information Systems Security Engineering WISSE13. (2013)
24. Abiteboul, S., Hull, R., Vianu, V.: Foundations of databases. Citeseer (1995)
25. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS quarterly **28**(1) (2004) 75–105
26. Gharib, M., Giorgini, P.: Detecting conflicts in information quality requirements: the may 6, 2010 flash crash. Technical report, Università degli studi di Trento (2014)
27. Trcka, N., van der Aalst, W., Sidorova, N.: Analyzing control-flow and data-flow in workflow processes in a unified way. Computer Science Report (08-31) (2008)
28. Deutsch, A., Hull, R., Patrizi, F., Vianu, V.: Automatic verification of data-centric business processes. In: Proceedings of the 12th International Conference on Database Theory, ACM (2009) 252–267
29. Cysneiros, L.M., Yu, E.: Addressing agent autonomy in business process management-with case studies on the patient discharge process. In: Proc. 2004 IRMA Conference. (2004)
30. Lapouchnian, A., Yu, Y., Mylopoulos, J.: Requirements-driven design and configuration management of business processes. In: Business Process Management. Springer (2007) 246–261
31. Wang, R.: A product perspective on total data quality management. Communications of the ACM **41**(2) (1998) 58–65
32. 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
33. 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