Satisfaction of Control Objectives by Control Processes[*]

Daniela Marino[1], Fabio Massacci[2], Andrea Micheletti[1], Nataliya Rassadko[2], and Stephan Neuhaus[2]

[1] Fondazione Centro San Raffaele del Monte Tabor
e-Services for Life & Health Unit, Via Olgettina 60 - 20132 - Milano, Italy
surname.name@hsr.it
[2] Dipartimento di Ingegneria e Scienze dell'Informazione
Università degli Studi di Trento, via Sommarive 14 - 38100 Trento, Italy
name.surname@disi.unitn.it

Abstract. Showing that business processes comply with regulatory requirements is not easy. We investigate this compliance problem in the case that the requirements are expressed as a directed, acyclic graph, with high-level requirements (called *control objectives*) at the top and with low-level requirements (called *control activities*) at the bottom. These control activities are then implemented by *control processes*. We introduce two algorithms: the first identifies whether a given set of control activities is sufficient to satisfy the top-level control objectives; the second identifies those steps of control processes that contribute to the satisfaction of top-level control objectives. We illustrate these concepts and the algorithms by examples taken from a large healthcare provider.

1 Introduction

Processes – no matter whether executed by people or by machines – are often governed by desirable or prescribed features of their execution. For example, if an Italian hospital dispenses drugs to a patient, the identity of the person requesting the dispensation must appear in an audit log, according to Legislative Decree no. 196 of 30 June 2003 "personal data protection code", "Computerized Authentication System", clauses 1, 2, 3 and 6 [23]. Processes that have these features are called *compliant*.

However, designers of such processes are faced with a dilemma: the desirable features are usually listed as high-level control objectives or goals, such as "Processing operations may only be performed by persons in charge of the processing that act under the direct authority of either the data controller or the data processor by complying with the instructions received.", but the actions to which such control objectives pertain happen only at a much lower level, such as "look up the user's ID and check authorization". In order to know that the action is influenced by the control objective, that objective must be successively decomposed until it is clear to which process steps it pertains. For example, a sub-objective of "Personal data undergoing processing shall be kept [...] in such a way as to minimise [...] the risk of their destruction or loss," [23, Section 31] could be "patient records may only be deleted after authorization by at least two authorized persons".

[*] Research partly supported by the EU under the project EU-IST-IP-MASTER (FP7-216917)

Sometimes, it may not be feasible to implement *all* the actions that are prescribed by control objectives. In this case, we want to know whether the subset that we *have* implemented is sufficient to guarantee the satisfaction of the high-level objective. For example, we could prescribe that patient records are anonymized even as they are assembled for sending to the local health administration. But failing to implement this anonymization would not be fatal if the records are anonymized during the sending process.

In this paper, we consider this problem on three levels:

- on the *design level*, we consider the decomposition of control objectives into sub-objectives. The objectives then become successively more specific on refinement until we consider them to be atomic. These atomic objectives can then be either implemented or not. We ask: "given a decomposition of objectives into sub-objectives and atomic objectives, and given that certain atomic objectives are implemented and others not, are the top-level objectives satisfied?" This allows us to claim compliance at the design level, when we plan to satisfy certain atomic control objectives, but don't have a concrete implementation yet.
- on the *implementation level*, we first consider the implementation of atomic objectives by processes. Steps in these processes will contribute to the satisfaction of different atomic control objectives. So we ask, "Does execution of a particular control process lead to the satisfaction of the top-level objectives?" This allows us to claim compliance by *adding independent controls*.
- on the *process level*, we recognise that control processes are usually woven into processes instead of being separate processes by themselves. For example, checking a user's authentication and authorization are usually parts of processes instead of being realized as separate processes. If we are given, for each process step, a list of atomic control objectives to whose satisfaction it contributes, we ask, "does every execution of this process lead to the satisfaction of the top-level control objective?" This allows us to claim compliance by *adding process-specific controls or by analysing controls that are alerady in place*.

The rest of this paper is organized as follows: after introducing our case study (§ 2), we formalize the problem (§ 3). Then, we look at the problems of objective satisfaction through the implementation of atomic control objectives (§ 4) and compliance of control process. (§ 5). After that, we review related work (§ 6) and finish with conclusions and further work (§ 7).

2 Example: Outpatient Drug Reimbursement

The case study considered in this paper is based on a concrete process from Hospital San Raffaele (HSR) in Milano, Italy, and cerncers drug reimbursement.

Private Hospitals with a officially recognized public functions (such as HSR) are charged with administering drugs or with providing diagnostic services to patients that use their structure (e.g,, because the corresponding public services are overbooked) and then are authorized to claim the cost of drug dispensation or diagnostic provisioning from the regional state health administration.

The Italian Direct Drug Reimbursement process is a mechanism that allows refunding hospitals for drugs administered or supplied in the outpatient departments to patients that are not hospitalized; this mechanism is called "File F" and guarantees continuity of care regardless of the different forms in which that care is provided.

As a consequence of their public function and because they treat sensitive data, the processes of HSR are highly regulated:

- First, the e-health services have to respect the Health Governmental Authority (e.g. Ministry, Regional Health authority, etc.) indications; these regulations or guidelines have to be followed by all the healthcare institutions and concern a wide spectrum of norms, e.g., from the Personal Electronic Health Record to the Accreditation procedures, from the clinical practice to the price of the hospital treatments. Moreover, the e-health services usually follow the healthcare standards related to a specific domain, such as HL7, DICOM, HIPAA, etc., depending on the service.
- Other regulations to consider are the Governmental indications about the privacy matters (personal data protection); the European framework is regulated by the "Directive 95/46/EC - privacy framework" that have to be implemented by each European state. There is also to consider the Directive 2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector and other governmental regulations regarding digital signatures, health data storage, etc.
- The final regulatory framework consists of Information & Communication Technologies Security standards, such as ISO/IEC 27002 [14] "code of practice for Information Security management", the ISO 15408 "Common Criteria for IT Security Evaluation" [15], the COBIT framework [13], ITIL [22], etc. Sometimes also business agreements between suppliers and customers impose security requirements.

In order to give a rough idea of the sheer volume of regulation, the simple process of authorization and accounting for the dispensation and recompensation of drugs (called "File F") is subject to the following (not exhaustive) set of regulations: Legislative Decree no. 196 of 30 June 2003 [23] "personal data protection code", "Additional Measures Applying to Processing of Sensitive or Judicial Data" clause 20; "Computerized Authentication System" clauses 1, 2, 3, 5; annex B, "processing by electronic means", "authorisation system" clauses 12, and 13, as well as regional circular 17/SAN 3.4.1997, which is successively amended by various notes and circulars auch as Circular No. 5/SAN 30_1_2004 [3], Circular No. 45/SAN 23_12_2004 [2], Note 30.11.2007 H1.2007.0050480 [6], Note 27.3.2008 H1.2008.0012810 [5], and Note 04.12.2008 H1.-2008.0044229 [4].

All the regulations and best practices above mentioned contribute to the definition of the control objectives of the HSR business process for performing regulatory compliance analysis. The set of control objectives for the File F process activities is augmented by various business objectives (also called business goals) that have to be satisfied to reach the correct process results.

3 Conceptual Model

Recall from Section 1 that we view processes as being governed by desirable or prescribed features of their execution, features which we called *control objectives*. In this section, we will formalize the concepts of objectives and objective decomposition, as well as the concept of implementing an atomic objective.

Control Objectives (COs) are requirements on the internal operations of a business that describe what needs to be done (e.g.,. follow certain industry best practices) or what needs to be achieved (e.g., certain states or outcomes). However, control objectives are not actionable because are phrased as requirements, not as procedures.

Example 1 (Regulatory Requirement). For the File F process, one regulatory requirement is “Legislative Decree no. 196 of 30 June 2003 ‘personal data protection code’, ‘Additional Measures Applying to Processing of Sensitive or Judicial Data’, clause 20”.

Example 2 (Control Objective). For the regulatory requirement described above, the following objectives (from ISO 27002) are particularly relevant: “access control” and “user access management”.

Control objectives like “access control” are not actionable. Rather, they have to be refined to a level that it is clear to which part of the business these refined objectives pertain and such that further refinement is no longer needed. We call such atomic objectives *control activities* (CAs). They are the policies, procedures, mechanisms, and organizational structures that are put in place to assure that control objectives are achieved. Control activities are embedded in business processes; that is they affect and change the inner workings of a business. Common synonyms include controls, countermeasures, and safeguards as well. Control activities, by definition, *are* actionable, because they are phrased as procedures.

Example 3 (Control Activity). One control activity that is pertinent to the control objective would be (in ISO 27002 parlance) “User registration”, or (in procedural parlance) “register users before granting them access”.

The problem is now to translate somehow from control objectives to control activities so that if we implement and execute the control activities, we automatically satisfy the control objectives.

To this end, we introduce the concept of *control objective refinement*, i.e., the replacement of a control objective (the *super-objective* by a number of more specific control objectives (the *sub-objectives* that together contribute to the satisfaction of the super-objective.

Example 4 (Objective Decomposition). The control objective “access control” in the File F example is achieved by having (from ISO 27002) “user access management” and “user responsibilities”.

A decomposition can therefore be seen as a graph whose nodes are the *control objectives*, which have the property of being satisfiable. Control objectives have a number of sub-objectives that contribute to the satisfaction of the super-objective, which is expressed by drawing a directed edge from the super-objective to the sub-objective. We distinguish between two cases:

- When the satisfaction of a single sub-objective is sufficient to satisfy the super-objective, we call the super-objective *OR-decomposed*.
- When all sub-objectives need to be satisfied in order to satisfy the super-objective, we call the super-objective *AND-decomposed*.

Leaves (control objectives that have no sub-objectives) are so specific that they are actionable and are therefore *control activities*. For the purpose of checking compliance, they have the property of being *implemented* or not. It is also reasonable to assume that refinement is acyclic, i.e., that no objective ultimately depends on itself for fulfillment. More formally, we have therefore:

Definition 1 (Objective Model). *An* objective model *is a non-empty, directed, acyclic graph $G = (V,E)$, where V is a set of* nodes *that can be either control objectives or control activities, and where $(m,n) \in E$ if n is a subgoal of m so that n contributes to the satisfaction of m. For $n \in V$, we write $n.\text{parents} := \{m \mid (m,n) \in E\}$ and $n.\text{children} := \{m \mid (n,m) \in E\}$.*

Since G is nonempty and acyclic, there exists a nonempty set of nodes n with $n.\text{parents} = \emptyset$. These are those objectives that do not function as sub-objectives to other objectives and are therefore called *global objectives*.

In our model, control activities they are implemented by *control processes* (CPs), including any configuration and maintenance work that is needed to keep the control operational. Control processes can be *structurally* composed of subprocesses, for which we use the notation shown in Fig. 1.

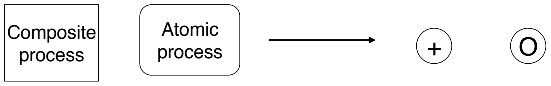

Fig. 1. Constructs of process decomposition. Composite processes are represented as boxes, tasks (atomic process that cannot be decomposed) are shown as rectangles with rounded corners. The flow of decomposition is denoted by an arrow. AND-decomposition (execution of all subprocesses required) is shown by a circle with plus inside, while (exclusive) OR (execution of at most one subprocess is required) is denoted as a circle with a O inside.

Example 5 (Process Decomposition). In Fig. 2 (left), we used a standard business process notation to show a process of File F dispensation. Its structural decomposition is shown in Fig. 3 (right). Namely, P_1 is the entire process depicted in Fig. 2 (left); P_2 is a sequence, consisting of all tasks before the first conditional diamond, P_3 is everything that is executed after the first conditional diamond. Note that P_2 and P_3 constitute an AND-decomposition of P_1. Next, P_2 is decomposed into and AND-structure consisting tasks $A2.1$, $A2.2$, $A2.3$. The decomposition of P_3 is more complex since it is the OR-decomposition consisting of the branches of the first conditional diamond. Therefore, it is either P_4, which is the YES-branch, or P_5, which is NO-branch.

Example 6 (Process-to-Objective Assignment). The overall conceptual model is shown on Fig. 4. The upper part of the figure is an objective model, where ovals and hexagons

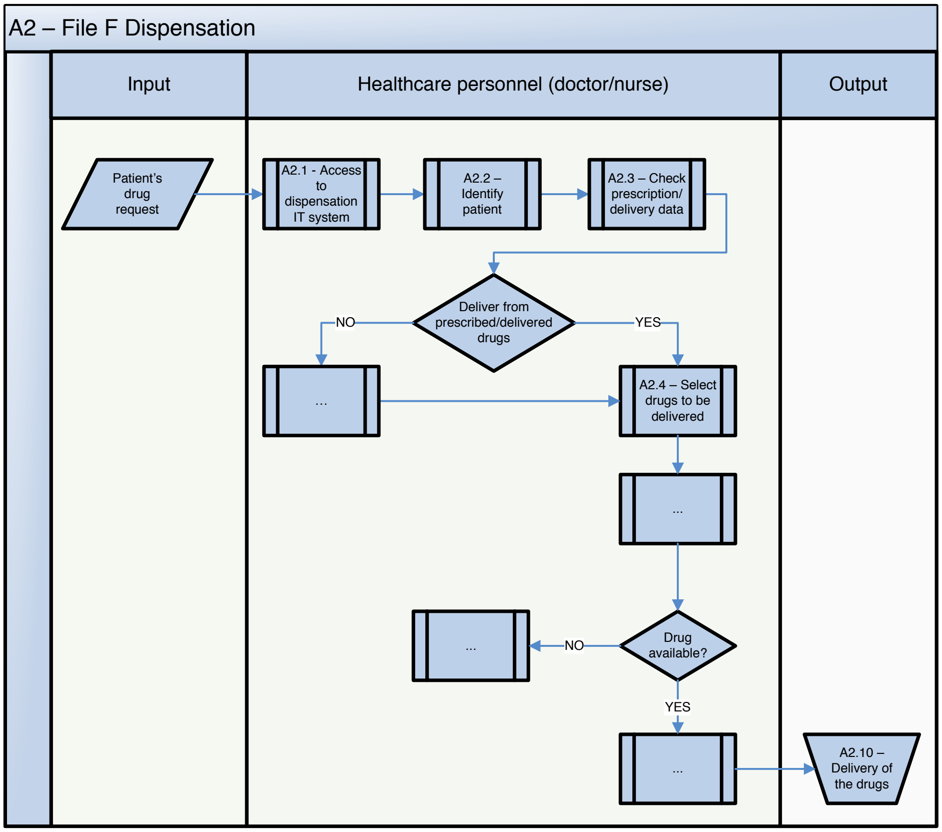
A2 – File F Dispensation
Input
Healthcare personnel (doctor/nurse)
Output
Patient's drug request
A2.1 - Access to dispensation IT system
A2.2 – Identify patient
A2.3 – Check prescription/ delivery data
Deliver from prescribed/delivered drugs
NO
YES
...
A2.4 – Select drugs to be delivered
...
Drug available?
NO
...
YES
...
A2.10 – Delivery of the drugs

Fig. 2. File F Dispensation process

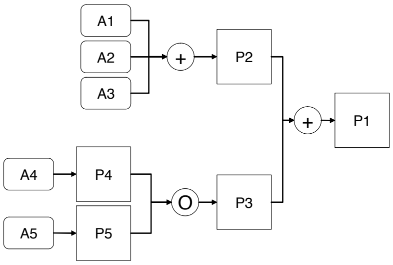

Fig. 3. File dispensation process structural decomposition

represent COs and CAs respectively. Namely, the global objective CO_3 can be satisfied if all CO_2 and $CO_{3.3}$ and AC (standing for $Access\ Control$) are satisfied. In their turn, satisfaction of CO_2 depends on satisfaction of both $CO_{2.1}$ and $CO_{2.2}$, where the first depends on implementation of activities CA_3 and CA_4 and the second depends on activity CA_3 only. On the other hand, the satisfaction of CO_3 requires an implementation of both CA_3 and CA_6. AC is satisfied if either $CO_{3.1}$ or $CO_{3.2}$ are satisfied. The satisfaction of

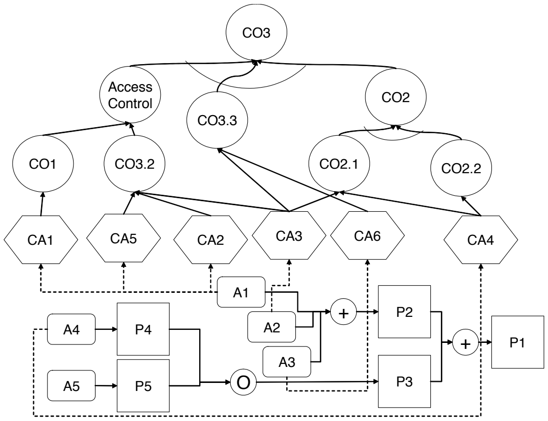

Fig. 4. Control objective satisfaction by executing control processes.

the latter COs rely on implementation of $\{CA_1, CA_2, CA_5\}$ and CA_3 respectively. The explanation of all COs and CAs will be presented in Sec. 4.

The lower part of Fig. 4 is dedicated to an executable process P_1 which requires a necessary execution of both P_2 and P_3. The former is implemented by tasks A_1, A_2, A_3. The latter is decomposed into choice execution of P_4 and P_5 that are implemented by tasks A_4 and A_5 respectively.

While executing a control process, we may contribute to the satisfaction of one of the sub-objectives of a global objective and hence ultimately to the satisfaction of that global objective itself. This is what we intuitively mean by *compliance* and what we show in Fig. 4 by a dotted line.

In order to check for compliance, we need to resolve three problems:

1. Given a set of implemented CAs, we need to check whether the global COs are satisfied ("Problem of Satisfaction").
2. Given an implementation of CAs by tasks and (sub)processes, we need to check whether the entire process is compliant to the objective model ("Problem of Compliance").
3. Given the execution of a control process, we need to identify a set of (sub)processes, execution of which leads to a satisfaction of a concrete control objective ("Problem of Contribution").

Resolving the first problem will show us whether the mechanisms that we have (or want to have) in place are sufficient to satisfy our global objectives. The resolution of the last two problems will then make it possible for us to do check for compliance more effectively because, if we know exactly which sub-process contributes to which control objective, we can more easily establish what we need to monitor.

4 Specification: Control Objective Satisfaction

Considering the case study, starting from the Regulatory Requirements, we can obtain the control activities that have to be performed. In a practical way, it is possible to use

Table 1. Control objectives, sub-objectives and control activities vs. regulatory requirements

Control objective (from ISO 27002)	**Sub-objective (from ISO 27002)**	**Control activities (from ISO 27002)**	**Source of Regulatory Requirement**
CO1: Access control	CO1.1: User access management	CA1 - User registration	Legislative Decree no. 196 of 30 June 2003 "personal data protection code", "Additional Measures Applying to Processing of Sensitive or Judicial Data" clause 20
		CA2 - User password management	Legislative Decree no. 196 of 30 June 2003 "personal data protection code", "Computerised Authentication System" clause 5. Legislative Decree no. 196 of 30 June 2003 "personal data protection code", "Additional Measures Applying to Processing of Sensitive or Judicial Data" clause 20
	CO1.2: User responsibilities	...	...
CO2: Information systems acquisition, development and maintenance	CO2.1: Correct processing in applications	CA3 - Control of internal processing	circular No.5/SAN 30_1_2004 and Note 30.11.2007 H1.2007.0050480 and Note 27.3.2008 H1.2008.0012810
		CA4 - Output data validation	circular No.5/SAN 30_1_2004 and Note 30.11.2007 H1.2007.0050480 and the Note 27.3.2008 H1.2008.0012810
	CO2.2: Technical vulnerability management	...	...

standard control objectives, sub-objectives and control activities related to a specific domain; for instance in our case we can consider the ISO 27002 "Information security management systems" standard. The specific control mechanisms to be implemented are then customized on the particular business process.

Table 1 reports an example of this methodology, where it has been performed a mapping between the Regulatory requirements and the ISO control activities; the control mechanisms that will be implemented on our business process will have to satisfy the regulatory requirements [3].

On the other hand we can identify the control objectives, the sub-objectives and the control activities coming from the business objectives of our process, as shown in Table 2. Also in this case it is possible to refer to the ISO standard for having a common reference for the control activities.

Then control activities are implemented by processes compliant to regulations. However, for example, due to tight budget, only a restricted set of activities can be implemented. However, the implemented activities should lead to the satisfaction of "global" control objectives.

In Fig. 5, we show our algorithm for satisfaction of global control objectives given the satisfaction of some control activities. The algorithm begins at the leaves (the CAs) of the objective model and proceeds upwards. To propagate satisfaction, we use an array UNTIL-SATISFIED that contains for all nodes the minimum number of sub-nodes that need to be satisfied in order for the node to be satisfied as well. Lines 1–6 compute the initial value of UNTIL-SATISFIED: an implemented control activity is automatically satisfied; a non-implemented CA can never be satisfied. If a node is an AND-decomposed CO, all of its sub-objectives need to be satisfied; for an OR-decomposed CO, the satisfaction of a single sub-objective suffices.

[3] We do not include CO_1 (and its subobjectives) into our objective model shown in Fig 4 because it is subsumed by $CO_{3.1}$.

Table 2. Business objectives of the File F process

Process phase	**Business objectives (or control objectives)**	**Sub-objectives**	**Control activities (from ISO 27002)**
A2. File F Dispensation	CO3: – Deliver the right drugs to the right patient; – Give input to logistic stock management	CO3.1: Doctors and nurses must have authorization and credentials for accessing the dispensation IT system	– CA1 - User Registration – CA2 - User Password management – CA5 - Review of User Access Right
		CO3.2: The original copy of prescription sheet with the signature of the doctor must be given to the nurse as dispensation request	CA3 - Control of internal processing
		CO3.3: Prescription data must be univocally assigned to a patient	– CA3 - Control of internal processing – CA6 - Documented operating procedures

```
Algorithm PROBLEM OF SATISFACTION RESOLUTION
Input: A control objective model $G = (V, E)$
Output: Identifies if global COs are satisfied.
 1: for all $n \in V$ do
 2:     if $n$ is a CA then
 3:         UNTIL-SATISFIED$[n] := 0$ if $n$ is implemented, 1 otherwise;
 4:     else
 5:         UNTIL-SATISFIED$[n] := |n.\text{children}|$ if $n$ is AND-decomposed, 1 otherwise;
 6: Insert into queue $Q$ all nodes $n \in V$ with UNTIL-SATISFIED$[n] = 0$;
 7: while $Q$ is not empty do
 8:     $n \leftarrow Q$;
 9:     for all $n' \in n.parents()$ do
10:         if UNTIL-SATISFIED$[n] \neq 0$ then
11:             UNTIL-SATISFIED$[n']$ := UNTIL-SATISFIED$[n'] - 1$;
12:             if UNTIL-SATISFIED$[n'] = 0$ then
13:                 $Q \leftarrow n'$;
```

Fig. 5. Algorithm PROBLEM OF SATISFACTION RESOLUTION

We also use a queue that contains satisfied control objectives. Initially, the queue contains all satisfied leaves (implemented CAs). At each iteration of the **while** loop starting at line 7, one node is removed from the front of the queue and its parents are examined. If the satisfaction of the current node is enough to also satisfy the parent, the parent is also marked as satisfied and appended to the end of the queue.

Theorem 1. *The algorithm is correct, terminates and has time complexity* $O(|V|^2)$ [4].

In a nutshell, the correct outcome of algorithm run should result in UNTIL-SATISFIED$[CO] := 0$ if CO is satisfied, 1 otherwise; for any CO. This issue can be easily demonstrated by proof by contradiction. Termination of the algorithm is evident due to acyclicity of objective model and the fact that each node can be added to queue only once. Since in cycle **while** in line 7 lasts $O(|V|)$ iterations, $O(|V|)$ operations are performed in line 9 at each iteration, the complexity is $O(|V|^2)$.

[4] Due to the lack of the space, we omit a detailed proof of theorem

5 Compliance of Control Processes

In this section, we want to tackle problems 2 and 3, namely which parts of a control process contribute to the satisfaction of control objectives.

Definition 2. *A* Process-to-Activity Assignment *is a mapping $\mathscr{A}$ from the set of processes to the set of control activities such that a process P is mapped to a control activity A if A is satisfied after P has completed. We write this assignment $P \rightarrow_{\mathscr{A}} A$ and say "P implements A".*

At the moment, we have no way of automating a process-to-activity assignment, so we assume that this is done manually.

Having identified process-to-activity assignment, we try to "dig" into each structural subprocess and to identify if this assignment may be alleviated. In other words, there might be many control processes assigned to one particular CA. This may happen not only because of the complexity of CA, but also because there is a need of "reserve" implementation that could be launched in the case of failure of the "main" implementation. Some of these assignments might be more costly to implement or difficult to audit, others might not. So, we want to identify the core subset of process-to-activity assignment which is necessary to implement in order to fulfill the root objectives. On the other hand, if we are able to distinguish some additional assignments leading to a "reserve" satisfaction of some COs, we will have a possibility to configure process-to-objective assignment in different ways w.r.t. our requirements to implementation cost or auditing difficulty.

Definition 3. *Let $G = (V,E)$ be a control objective model with a set G of global objectives, let $\{A_1,\ldots,A_n\} \subseteq V$ be a set of control activities, and let P be a process, composed of sub-processes $\{P_1,\ldots,P_m\}$ that implement control activities $A_1,\ldots,A_n$. Let $\mathscr{A}$ be a process-to-activity assignment. We call $\mathscr{A}$* correct *if the gloabl objectives in G are satisfied when P completes. In this case, we write $P \models G$.*

Given a process-to-activity assignment, we would like to test its correctness and also to identify which part of the process (1) is compliant with a particular control objective, and (2) contributes to a satisfaction of a particular control objective. We can answer these questions with the help of algorithm presented in Fig. 6. For this purpose, for each control objective $n \in V$, we maintain a set $\text{IMPLEMENTEDBY}[n]$ of those subrocesses of P that contribute to control objective satisfaction, and for each subprocess p of P, we maintain a set $\text{ACHIEVES}[p]$ that are implemented by that p.

The input of the algorithm is (1) a process-to-activity assignment, (2) the objective model, and (3) the control process and its sub-processes. From the process-to-activity assignment, we can easily instantiate IMPLEMENTEDBY and ACHIEVES for corresponding subprocesses and control objectives according to lines 7–10.

After this initialization, subprocesses start the propagation of their implementation to superprocesses. More precisely, each process p' implements those control objectives that are available for propagation of satisfaction from control objectives implemented by subprocesses of p'. To calculate such a reachability, we use the function Reach which is an algorithm PROBLEM OF SATISFACTION RESOLUTION having as input a set of

Algorithm PROBLEM OF COMPLIANCE AND CONTRIBUTION RESOLUTION

```
Input: A process-to-activity assignment $\mathscr{A}$, an objective model $G = (V, edges)$, an executable process $P$, composed of subprocesses $P_1, \ldots, P_n$
Output: Structures that represent (1) compliance of each subprocess to a certain control objective, and (2) satisfaction of each control objective with a certain set of subprocesses of executable process.
Put into $Q_P$ all process that implement some CAs;
for all nodes $e$ of objective model do
    $\textsc{ImplementedBy}(e) = \emptyset$;
for all subprocesses and tasks $P'$ of the executable process $P$ do
    $\textsc{Achieves}(P') = \emptyset$'
    $\textsc{Visited}[P'] := false$;
for all assignments $P \rightarrow_{\mathscr{A}} \{A_1, A_2, \ldots, A_n\}$ do
    $\textsc{ImplementedBy}(A_i) := \textsc{ImplementedBy}(A_i) \cup \{P\}$;
    $\textsc{Achieves}(P) := \textsc{Achieves}(P) \cup \{A_1, A_2, \ldots, A_n\}$;
    $\textsc{Visited}[P] := true$;
while $Q_P$ is not empty do
    $P' \leftarrow Q_P$;
    for all $p \in P'.superprocesses$ do
        if $p$ is choice then
            $ReachableObjectives := \text{Reach}(G, \cap_j\{\textsc{Achieves}(p.subprocesses())\})$;
        else if $p$ is flow or sequence then
            $ReachableObjectives := \text{Reach}(G, \cup_j\{\textsc{Achieves}(p.subprocesses)\})$;
        $\textsc{Achieves}(p) := ReachableObjectives$;
        for all objectives $CO \in ReachableObjectives$ do
            $\textsc{ImplementedBy}(CO) := \textsc{ImplementedBy}(CO) \cup \{P'\}$;
        if not $\textsc{Visited}[P']$ then
            $\textsc{Visited}[P'] := true$;
            $Q_P \leftarrow P$;
```

Fig. 6. Algorithm PROBLEM OF COMPLIANCE AND CONTRIBUTION RESOLUTION

satisfied control objectives or implemented CAs that are pushed into the queue. Respectively, values associated to the corresponding nodes in objective model are equal to "0", while the other nodes are associated the values according to the algorithm. The algorithm proceeds propagating satisfaction bottom-up. As soon as the algorithm terminates, satisfied control objectives represent the result of Reach function.

If superprocess p' is AND-decomposed (flow or sequence of subprocesses), it satisfies all control objectives satisfied/implemented by each of its subrocesses. It means that we can propagate also satisfaction in the objective model. The satisfaction is propagated from the *union* of control objectives implemented by subprocesses of p'. On the other hand, if p' is OR-decomposed (choice), it can implement only those control objectives that are reachable from the *intersection* of control objectives implemented by subprocesses of p'. That is why OR-decomposition of process cannot implement AND-decomposition of objectives.

Theorem 2. *Algorithm* PROBLEM OF COMPLIANCE AND CONTRIBUTION RESOLUTION *is correct, terminates, and has complexity $O((|V|^2 + |nodes|) \times |P|)$, where $|V|$ is the number of nodes in objective model, $|P|$ is the number of subprocesses of P.*

Proof. The correctness means that for each CO, IMPLEMENTEDBYCO contains only those elements that contribute to satisfaction of CO; for each p, ACHIEVESp contains only those elements that are satisfied by p.

We prove the correctness by the method of induction. The base of induction is a process-to-activity assignment $\mathscr{A}$, which is correct by default: for all assignments

Table 3. Result of algorithm run

CO	ImplementedBy(CO)	P	Achieves(P)
CA_1,CA_2,CA_5	A_1	A_1	$\{CA_1,CA_2,CA_5\}$
CA_3	A_2	A_2	CA_3
CA_4	$\{A_4,P_4\}$	A_3	CA_6
CA_6	A_3	A_4	CA_4
$CO_{3.1},CO_{3.2},CO_{3.3},AC$	$\{P_2,P_1\}$	P_3,P_5,A_5	$\{\emptyset\}$
$CO_{2.1},CO_{2.2},CO_2,CO_3$	$\{\emptyset\}$	P_2,P_1	$\{CO_{3.1},CO_{3.2},CO_{3.3},AC\}$
		P_4	CA_4

$P \rightarrow_{\mathscr{A}} \{A_1,A_2,\ldots,A_n\}$, the corresponding ImplementedBy and Achieves are calculated correctly.

Now let us consider any process p from process decomposition such that $p \not{in}\mathscr{A}$. Let's assume that for subprocesses of p, all Achieves and corresponding ImplementedBy are calculated correctly. Step of induction: Achieves(p) is calculated correctly. Indeed, if Achieves(p) is incorrect, then *ReachableObjectives* is calculated incorrectly. Since function Reach is correct by Theorem 1, then Achieves of subprocesses of p are calculated incorrectly, which contradicts to the assumption of the induction step. Therefore, for all processes of process decomposition, Achieves are calculated correctly.

Let us assume that there exists CO such that ImplementedByCO is calculated incorrectly. It means, that there exists a process p such that its *ReachableObjectives* is calculated incorrectly, which contradicts to the statement proved previously.

We will prove termination by showing that in cycle **while** in line 11, processes can be added to the queue at most once. Since process decomposition is finite and since one node is removed on every iteration, termination then follows. Initially (line 1), the elements of queue are all distinct. Line 13 guarantees that only superprocesses are added to the queue. For an element to be added to the queue twice, it would therefore have to be its own superprocesses which is impossible.

Finally, we prove the complexity result. The most complex calculation is hold in cycle **while** in line 11. Above, we have proved that the queue length is $O(|P|)$. At each iteration we pop exactly one process. For each popped process, in line 13 we check its parents. Due to the *structural* nature of our process decomposition model, each process has only *one* super process. Thus, for a single parent, we run algorithm Reach which has complexity $O(|V|^2)$ because of Theorem 1. In line 19, we have to update ImplementedBy for some COs the total number which is less than $O(|V|)$. Thus the complexity is not more than $O((|V|^2+|V|)\times|P|)$.

Example 7. In this example we will show the run of the algorithm Problem of Compliance and Contribution Resolution. Since A_1, A_2, A_3 are composed into AND-execution (i.e., sequence in the original workflow), we have to calculate function Reach over the *union* of activities that they implement; and the union is CA_1, CA_2, CA_3, CA_5, CA_6. Thus, $\text{Achieves}(P_2) = \text{Reach}(G,\{CA_1,CA_2,CA_3,CA_5,CA_6\})=$ $\{CO_{3.1},CO_{3.2},AC,CO_{3.3}\}$, and P_2 is added to corresponding sets ImplementedBy of control objectives respectively.

On the other hand, A_4 implements CA_4. And $\text{Achieves}(P_4) = \text{Reach}(G,AC_4) = \{CA_4\}$. For P_5 we cannot calculate ImplementedBy since A_5 does not implement anything. Therefore at P_3 we have the problem: intersection of CA_4 with empty set is an empty set.

P_2 and P_3 are composed into sequence (AND-composition), therefore Reach will be calculated over $\textsc{Achieves}(P_2) \cap \textsc{Achieves}(P_3)$=$\{CO_{3.1}, CO_{3.2}, AC, CO_{3.3}\}$. This will be a set of COs that are satisfied by P_1. Unfortunately, P_1 does not satisfy the root objective CO_3.

The complete output of the algorithm is shown in Table 3.

If we investigate column $\textsc{ImplementedBy}$ of Table 3, we will notice that there is a set of processes that do not satisfy any CO. These processes can be organized as a path of a tree-like structure of process decomposition model, e.g., $P_3 \to P_5 \to A_5$. Thus, we can detect the source of incompliance. Namely, the situation that is described as in the example above could be fixed if A_5 were designed to implement some CA in objective model. It seems to be obsolete and even more non compliant to the hospital regulations and therefore it should be deleted. If we eliminate the process P_5 then P_3 obtains $\textsc{ImplementedBy}(P_4) = \{CA_4\}$ which being united with $\textsc{ImplementedBy}(P_2)$ for calculation of $\textsc{ImplementedBy}(P_1)$ will result in the complete satisfaction of objective model.

6 Related Work

The problem of compliance to regulatory requirements was investigated from different angles and by means of different methodologies.

A recent survey on compliance checking [17] classifies various proposals either *design time* or *execution time* or *audit time* compliance checking. It's easy to see that our proposal can be attributed to the first class, i.e., design time compliance checking. Indeed, we reason on compliance by analysing business process structure, CO model, and process-to-objective assignment. There have been proposed other design time compliance checking methodologies.

A logical language PENELOPE proposed in [10] makes use of temporal deontic assignments from compliance requirements (obligations and permissions) to create state space. The latter is refined than in control flow. In contrast to [10], we concentrate do not consider workflow but rather its structural complexity, i.e., AND/OR decomposition.

A range of various model-driven proposals for compliance analysis were proposed, for example, in [16] (TROPOS [1]), [19] (pi-calculus and temporal logic), [20] (Deontic logic), [9] (REALM framework [8]). The idea of all these proposals is that requirements are modelled in the first turn either by means of logical prepositions or goal model, and then a workflow model is derived from requirements model. Thus, the derived business process should be compliant to requirement model by design.

Schmidt et al. [25] designed a compliance ontology w.r.t. regulations. The compliance checking is based on verification of instantiated classes of compliance ontology against process ontology. This automatic procedure could be used in ours at the stage of process-to-objective assignment which we assume to be a human-related task. Similar approach was proposed in [18], where semantic-based architecture for compliance checking was sketched. The difference is that compliance ontology is called *semantic* *policy* which is assumed to be enforced against business process semantic. The core policy ontology was designed as well.

The notion of *compliance pattern* (i.e., commonly occurring business process model which is proven to be compliant to some requirements) was introduced in [7]. Compliance patterns are used for compliance violation detection and also provide heuristic guidance to resolve non-compliance by modification of the business process.

An attempt to design a formal framework for business process compliance is presented in [21]. Basically, the framework relies on propositional logic to model risks, controls, business process activities.

A methodology that refines regulatory requirements to control activities according to risks (as in the current paper) was proposed in [24], [11]. After that, controls that is supposed to verify the compliance is encoded into prepositions of Formal Contract Language [12].

7 Conclusion and Further Work

In this paper, we presented a methodology of design-time compliance checking between regulations and a control process. Our methodology is based on the notion of objective model which is derived from regulations and refined into simple instructions that can be easily implemented into small functions and procedures and later organized into a control process. The correctness of implementation can be checked by the algorithms presented in this paper. Namely, the proposed algorithms (1) verify the satisfaction of the root objective, (2) identify which subprocess contributes to satisfaction of which COs, (3) means to detect the source of incompliance.

Currently, we are working on the model of compliance of a *controlled* process that is process interwoven with a control process considered in the current paper. As the future work, we would like to develop an automated procedures for process-to-objective assignment which for now we assume to be performed manually by a human. Next, we are going investigate runtime compliance, i.e., compliance of business process execution traces to objective model. This will allow to include temporal COs into considerations and thus to extend the range of applicability of our methodology. Finally, we plan to introduce a notion of *compliance to some extend* in our methodology. Namely, we are working currently on the notion of key indicators which are metrics that are specific to business processes and so avoid one persistent metrics-related problem.

References

[1] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini. TROPOS: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, May 2004.

[2] Il Dirigente del Sanita Regione Lombardia. File f circular no. 45/san 23_12_2004. `http://www.sanita.regione.lombardia.it/circolari/04_45san.pdf`, 2009.

[3] Il Dirigente del Sanita Regione Lombardia. File f circular no. 5/san 30_1_2004. `http://www.sanita.regione.lombardia.it/circolari/04_05san.pdf`, 2009.

[4] Il Dirigente del Sanita Regione Lombardia. File f note 04.12.2008 h1.2008.0044229. `http://www.sanita.regione.lombardia.it/circolari/nota2008_44229.pdf`, 2009.

[5] Il Dirigente del Sanita Regione Lombardia. File f note 27.3.2008 h1.2008.0012810. `http://www.sanita.regione.lombardia.it/circolari/nota2008_12810.pdf`, 2009.

[6] Il Dirigente del Sanita Regione Lombardia. File f note 30.11.2007 h1.2007.0050480. `http://www.sanita.regione.lombardia.it/circolari/nota2007_50480.pdf`, 2009.

[7] Aditya Ghose and George Koliadis. Auditing business process compliance. In *Proceedings of the 5th international conference on Service-Oriented Computing (ICSOC"07)*, pages 169–180, Berlin, Heidelberg, 2007. Springer-Verlag.

[8] Christopher Giblin, Alice Y Liu, Samuel Müller, Birgit Pfitzmann, and Xin Zhou. Regulations expressed as logical models (realm). In *Proceedings of the 18th Annual Conference on Legal Knowledge and Information Systems (JURIX 2005)*, pages 37–48, Amsterdam, 2005. IOS Press.

[9] Christopher Giblin, Samuel Müller, and Birgit Pfitzmann. From regulatory policies to event monitoring rules: Towards model-driven compliance automation. Technical Report RZ 3662, IBM Research, 2006.

[10] Stijn Goedertier and Jan Vanthienen. Designing compliant business processes with obligations and permissions. In *Business Process Management Workshops (BPM'06)*, volume 4103 of *Lecture Notes in Computer Science*, pages 5–14. Springer, 2006.

[11] Guido Governatori, Jorg Hoffmann, Shazia Sadiq, and Ingo Weber. Detecting regulatory compliance for business process models through semantic annotations. In *4th International Workshop on Business Process Design*, 2008.

[12] Guido Governatori and Zoran Milosevic. A formal analysis of a business contract language. *International Journal of Cooperative Information Systems*, 15(4):659–685, 2006.

[13] ISACA. Cobit. `www.isaca.org/cobit/`, 2008.

[14] ISO/IEC. ISO/IEC 27001:2005: Information security management systems, 2005.

[15] ISO/IEC. ISO/IEC 15408: Common criteria for information technology security evaluation. `http://www.commoncriteriaportal.org/thecc.html`, 2009.

[16] Raman Kazhamiakin, Marco Pistore, and Marco Roveri. A framework for integrating business processes and business requirements. In *EDOC '04: Proceedings of the Enterprise Distributed Object Computing Conference, Eighth IEEE International*, pages 9–20, Washington, DC, USA, 2004. IEEE Computer Society.

[17] Marwane El Kharbili, Ana Karla A. de Medeiros, Sebastian Stein, and Wil M. P. van der Aalst. Business process compliance checking: Current state and future challenges. In *Modellierung betrieblicher Informationssysteme - Modellierung zwischen SOA und Compliance Management (MobIS'08)*, volume 141 of *LNI*, pages 107–113, 2008.

[18] Marwane El Kharbili and Sebastian Stein. Policy-based semantic compliance checking for business process management. In Peter Loos, Markus Nuttgens, Klaus Turowski, and Dirk Werth, editors, *MobIS Workshops*, volume 420 of *CEUR Workshop Proceedings*, pages 178–192. CEUR-WS.org, 2008.

[19] Y. Liu, S. Müller, and K. Xu. A static compliance-checking framework for business process models. *IBM Syst. J.*, 46(2):335–361, 2007.

[20] Kioumars Namiri and Nenad Stojanovic. A model-driven approach for internal controls compliance in business processes. In Martin Hepp, Knut Hinkelmann, Dimitris Karagiannis, Rudiger Klein, and Nenad Stojanovic, editors, *SBPM*, volume 251 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[21] Kioumars Namiri and Nenad Stojanovic. Towards a formal framework for business process compliance. In *Proceedings of Multikonferenz Wirtschaftsinformatik (MKWI'08)*. GITO-Verlag, Berlin, 2008.

[22] Office of Governance Commerce. IT infrastructure library. `http://www.itil.org/en/`, 2009.

[23] The President of the Italian Republic. Personal data protection code: Italian legislative decree no. 196 dated 30 june 2003. `http://www.garanteprivacy.it/garante/document?ID=1219452`, 2009.

[24] Shazia Wasim Sadiq, Guido Governatori, and Kioumars Namiri. Modeling control objectives for business process compliance. In *Proceedings of the 5th International Conference on Business Process Management (BPM"07)*, volume 4714 of *Lecture Notes in Computer Science*, pages 149–164, September 24-28 2007.

[25] Rainer Schmidt, Christian Bartsch, and Roy Oberhauser. Ontology-based representation of compliance requirements for service processes. In *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management held in conjunction with the 3rd European Semantic Web Conference (ESWC'07)*, volume 251 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.