Grafting Regulations into Business Protocols: Supporting the Analysis of Risks of Violation

Matteo Baldoni, Cristina Baroglio, Elisa Marengo, Viviana Patti

Abstract

The interaction of autonomous and heterogeneous business partners is often specified by business protocols. Particularly challenging is the case when such protocols must embed regulations, that change along time. In this work we focus on commitment-based protocols extended with temporal constraints among commitments and facts. We formalize a notion of "grafting" of new regulations inside protocols, where the new activities, foreseen by the regulation, are properly interleaved with the original ones thanks to the temporal contraints. We present a tool that exploits these principles to support the analysis of risks of violation, an apply it to two real-world case studies. This is a starting point for defining a notion of protocol compositionality that meets the requirements of flexible enactment, typical of cross-business settings, as well as that of modularity typical of Software Engineering.

1 Introduction

Business processes involve autonomous partners with heterogeneous software designs and implementations. In many practical settings, the reality in which business processes operate is characterized by a high degree of regulation. This is, for instance, the case of banking and of trading services, and of personal data flow management. The single organization needs to actively determine its processes on a permanent basis, to understand how regulations impact on the internal organization, to reason about possible *risks of violation*, and to ensure *compliance* to directives and laws. In such cases, the specification of the business interaction acquires a normative value and is commonly referred to as *business protocol*.

Business protocols must accomodate different needs. Interactions are usually *cross-business*: the minimization of the effort needed to define proper interfaces and of the altering of internal implementations calls for abstractions that capture the contractual relationships among the partners [20]. Business protocols must enable a *flexible enactment*, in order to allow the interacting parties, who are heterogeneous, autonomous, and basically self-interested entities, to find the way of interacting that better suits their characteristics and requirements. Flexibility is important to allow the business partners to profit of opportunities or to make the most efficient use of their time that is possible.

Moreover, business protocols must be *modular* in a way that simplifies keeping them compliant to regulations, which often change along time. Existing approaches to protocol specification (e.g. BPEL, WS-CDL) rely on the specification of control and business flows. This procedural view makes protocols not suitable to easily take in new regulations because the composition techniques, that can be applied, easily impose unnecessary orderings of the interactions w.r.t. what foreseen by the regulation, and, especially, because new regulations often impose the execution of new activities that are to be interleaved with the previously existing ones. In other words, these standards by and large require to rewrite the protocols from scratch.

Let us consider the case of directives issued by supranational authorities and institutions, like the European Union (EU) or the Organisation for Economic Co-operation and Development (OECD), that must be reconciled with the laws of the single nations. One example is the Markets in Financial Instruments Directive (MiFID for short), directive number 2004/39/EC [1], issued by the European Commission within the Financial Services Action Plan, which represents a fundamental step in the creation of an integrated and harmonized financial market within EU. Another example is the OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data [16], which regulates the management of personal data, by imposing new activities aimed at protecting the data owners. MiFID and OECD Guidelines do not simply stratify on top of previously existing sales protocols: they *graft* onto them, adding new activities which are interleaved with those of the previous protocols. For example, OECD requires that when some data is asked, before sending them, it is necessary to verify their accuracy and that the purpose for which they are requested matches with the purpose for which the data was left. In the case of MiFID, the bank must verify the order

before sending the contract. In both cases, asking data and sending data (or the contract) are activities foreseen by the original protocols, while the others are introduced by the new regulations.

Our proposal starts from commitment-based approaches [19] for representing business protocols, along the line of [20], in order to meet the requirement of flexibility. Indeed, commitments, abstracting from operational details, allow a flexible specification of business intents and relationships that involve autonomous and heterogeneous parties. The commitments that the interacting parties have towards the others are objectively inferrable from their observable behavior, thanks to the social semantics of the commitment approach. In order to meet, instead, the requirement of modularity and to allow the grafting of regulations onto business protocols, we adopt the formal framework in [4]. In this framework, commitment-based protocols include temporal regulations (e.g. that the execution of a purchase must occur only after the client was profiled) and the specification separates a constitutive and a regulative part. The new activities will be simply added to the constitutive specification of the protocol, while the new temporal regulations will declaratively specify when, how, where the added activities are to be used. Thus, the temporal regulations represent those "grafting points" that allow the accomodation of a new regulation into a business protocol.

Another contribution of this paper is a tool for allowing the business analyst to study all the possible enactments of a protocol in order to detecting the possible violations. In our framework, the interaction of a set of parties complies to a business protocol (i.e. it causes no violation) if, in the end, all the commitments they have taken were fulfilled and no temporal regulation was broken. The analysis of possible violations amounts to the identification of the risks the interaction could encounter. The evaluation of such risks will allow the definition of operational strategies, that will affect the business process, by, alternatively, preventing the occurrence of violations (*regimentation*) or implementing alerting mechanisms (*enforcement*) [13]. The tool is an extension of Winikoff et al.'s enhanced commitment machine [21] which allows exploring all the possible executions of a business protocol, showing all the violations. The implementation is done in *tuProlog* and the software interprets a 2CL business process specification by means of a parser written in Java. The tool was applied to the two mentioned real-world case studies, i.e MiFIDand OECD Guidelines. The MiFID case study is one of the benchmarks of the ICT4LAW project (`http://www.ict4law.org`).

The paper is organized as follows. Section 2 introduces the formal model, the notion of grafting, and the extended commitment machine. Section 3 describes the case studies and uses them to test the proposal. Conclusions and related works end the paper.

2 Business protocols

In this work we adopt a representation of the *business protocol*, i.e. of the specification of how the business interaction should be carried out, based on commitments. In particular, we adopt the approach discussed in [3, 4], that features an explicit distinction between a *constitutive* and a *regulative* specification. The former defines the protocol actions, while the latter encodes a set of temporal constraints. Both specifications are defined based on *commitments*. Commitments are directed from a debtor to a creditor. The notation $C(x, y, r, p)$ denotes that agent x commits to an agent y to bring about consequent condition p when the antecedent condition r holds. When r equals $true$, we use the short notation $C(x, y, p)$. The business partners share a social state that contains commitments and other literals that are relevant to their interaction. Every partner can affect the social state by executing actions, whose definition is given in terms of operations onto the social state, see [22]. The partners' behavior is affected by commitments, which have a *regulative* nature, in that debtors should act in accordance with the commitments they have taken.

Definition 1 (Business protocol) *A business protocol* P *is a tuple* $\langle Ro, F, A, C\rangle$*, where* Ro *is a set of roles, identifying the interacting parties,* F *is a set of literals (including commitments) that can occur in the social state,* A *is a set of actions, and* C *is a set of constraints.*

The set of social actions A, defined on F and on Ro, forms the *constitutive specification* of the protocol, while the set of constraints C, defined on F and on Ro too, forms the *regulative specification* of the protocol. We assume that facts persist in the social state, they denote observations about events that occurred.

Constitutive specification. The *constitutive specification* of actions is given by defining their meaning in terms of how they affect the social state. The specification follows the grammar below, where the *means* construct amounts to a *counts-as* relation [17]:

$A \rightarrow (Action$ **means** $Operation$ **if** $Cond)^+$
$Action \rightarrow protocolAction$
$Operation \rightarrow Op(commitment) \mid fact \mid$
$\quad Operation \wedge Operation$
$Op \rightarrow$ CREATE | DELETE | RELEASE | DELEGATE |
$\quad$ ASSIGN | ...
$Cond \rightarrow literal \mid Cond \ \wedge \ Cond \mid Cond \ \vee \ Cond \mid$
$\quad Cond$ XOR $Cond$

where *protocolAction* is the name of an action of the protocol; *Cond* specifies the context in which the counts-as relation holds; *Op* is one of the operations on commitments; *commitment* is a commitment of form $C(x, y, r, p)$, where

Relation	*Type*	*Operator*	*Meaning*
Response	pos.	$A \bullet\!\!\rightarrow\!\triangleright\ B$	If A occurs, B must hold at least once afterwards (or in the same state). It does not matter if B already held before A.
	neg.	$A \bullet\!\!\not\rightarrow\!\triangleright\ B$	If A holds, B cannot hold in the same state or after.
Before	pos.	$A \rightarrow\!\!\triangleright\!\bullet\ B$	B cannot hold until A becomes true. Afterwards, it is not necessary that B becomes true.
	neg.	$A \not\rightarrow\!\!\triangleright\!\bullet\ B$	In case B becomes true, A cannot hold beforehand.
Cause	pos.	$A \bullet\!\!\rightarrow\!\triangleright\!\bullet\ B$	It is the conjunction of the base *response* and base *before* relations: $A \bullet\!\!\rightarrow\!\triangleright\ B$ and $A \rightarrow\!\!\triangleright\!\bullet\ B$.
	neg.	$A \bullet\!\!\not\rightarrow\!\triangleright\!\bullet\ B$	It is the conjunction of the base *response* and base *before* negative relations: $A \bullet\!\!\not\rightarrow\!\triangleright\ B$ and $A \not\rightarrow\!\!\triangleright\!\bullet\ B$.

Table 1. 2CL **operators and their meaning.**

x and y are roles in Ro and r and p are formulas in disjunctive normal form of propositional literals in F; *fact* is a positive or negative proposition that does not concern commitments and which contributes to the social state (they are the conditions that are brought about); and *literal* can be either a commitment or a positive or negative proposition (where negation means that a certain literal does not hold in the social state)

Regulative specification. The *regulative specification* is expressed in 2CL [3, 4]. This language allows the designer to express many kinds of constraints describing the legal evolutions of the social state. As underlined in [2, 15], constraint-based declarative representations provide abstractions which allow to explicitly capture what is mandatory and what is forbidden, without the need to express the set of possible executions extensionally. For this reason, models remain compact improving flexibility: they specify what is desired and undesired, leaving all that remains unconstrained. This is an advantage with respect to procedural approaches, characterized by a prescriptive nature which requires the specification of *all* the allowed evolutions. It also accommodates naturally to the commitment-based approach, where a central issue is the respect of the agents' autonomy. 2CL follows the grammar:

$$C \rightarrow (Cond\ op\ Cond)^*$$

C, see Def. 1, is a set of constraints of the form $A\ op\ B$, where A and B are formulas of literals and op is one of the operators supplied by the language. The complete list of possible operators is fully described in [3, 4]. Table 1 reports only those that are used in the case study. 2CL allows the expression of temporal constraints on execution paths., allowed by the consitutive specification of the protocol A natural choice for formalizing such constraints is *Linear Temporal Logic* (LTL) [8]. This kind of logic allows the identification of those executions which satisfy the constraints of interest. The work in [4] introduces a LTL semantics for the 2CL operators, that formalizes the intuitive semantics that we reported in Table 1.

Grafting. One of the main advantages of the declarative approach, that we have proposed for the representation of business protocols, is that it supports a modular composition of such protocols, as hoped for in [14]. This achievement is obtained thanks to an explicit separation of the constitutive and of the regulative aspects, that enables to obtain the desired extension by performing a simple union of the components of the protocols. The grafting of two business protocol P_1 and P_2 is denoted by the operator $\uplus$ and defined as follows:

Definition 2 (Grafting) *Let* $P_1 = \langle Ro_1, F_1, A_1, C_1\rangle$ *and* $P_2 = \langle Ro_2, F_2, A_2, C_2\rangle$ *two business protocols. The grafting* $P_1 \uplus P_2$ *is the tuple:*

$$P_1 \uplus P_2 = \langle Ro_1 \cup Ro_2, F_1 \cup F_2, A_1 \cup A_2, C_1 \cup C_2\rangle$$

We assume that the action names of the two protocols are disjoint; in case this does not happen, a renaming can be applied. In the examples we will show how the constraints play the important role of "zippers" that tie temporally stratified specifications, by introducing coordination patterns.

2.1 A Commitment Machine for Business Protocols

The interaction of a set of parties will be compliant to a business protocol when all the commitments they have towards the others, and that are objectively inferrable from their observable behavior, are satisfied (as usual in the social approach), *and* the overall execution respects all the constraints. Intuitively, the addition of a regulative specification by means of 2CL constraints restricts the set of acceptable executions.

Some approaches to commitment protocols propose an operational semantics that relies on commitment machines to specify and execute protocols [22, 23, 21]. Some others, like [9], use interaction diagrams, operationally specifying commitments as an abstract data type, and analysing the commitment's life cycle as a trajectory in a suitable space. Other approaches rely on temporal logics to give a formal semantics to commitments and to the protocols defined upon them. Among these, [11] uses DLTL, while [7] adopt extensions of CTL*. Given the specification of a set of social actions, all these approaches allow the inference of

those executions, which are legal with respect to the protocol. In particular, commitment machines [22] (later refined in [21]) specify the possible states of an execution, the actions that are used for doing the transitions, and the possible final states of the protocol. The meaning associated with each state specifies which commitments are active in that state, and the meaning associated with each action defines how the commitments are affected by the action, leading to a state change. Intuitively, commitment machines allow the formalization of legal executions by taking into account only the *constitutive* specification of the social actions.

2CL allows the expression of temporal constraints on execution paths. Since commitment machines express legal execution paths, temporal constraints can be used to restrict this set. In this perspective, we implemented an extension of Winikoff et al.'s enhanced commitment machine [21] by introducing an automated verification of constraints. This extended commitment machine allows exploring all the possible executions of a business protocol, showing the regulative violations, i.e. both those states in which some constraint is violated and those states that contain unsatisfied commitments. The implementation is done in Prolog.

A *legal execution* of a commitment-based protocol, enriched by means of 2CL regulative specifications, is an execution that is accepted by the commitment machine built upon the constitutive specification of the protocol, and that, when interpreted as a linear temporal model, satisfies the LTL formulas corresponding to the regulative specification of the protocol. Based on this characterization, it is possible to provide mechanisms for verifying that an agent is behaving in respectance to the protocol. When this does not happen, we say that a *violation* has occurred. So, if in a commitment-based protocol, made only of the constitutive specification of actions, violations are detected only when a commitment remains unsatisfied in a final state, in our proposal, we also detect violations *during the execution*, violation amounting to the fact that a constraint is not respected.

The implementation builds upon the source code published by Winikoff et al. and extends the labelling of the states. The verification can be done efficiently for the 2CL operators reported in Table 1, because by working on facts/events, it is possible to perform it as verification of properties on single states instead of taking into account whole paths, as usually done by LTL model checking. For example, a *before* constraint of the kind $a \rightarrowtail b$ is violated when the state contains b but not a. Such simplification does not hold for all the operators of the complete 2CL [4].

The output of the commitment machine is an annotated and colored graph of all the possible interactions (it is a reachability graph). The graph includes all the interactions that are possible, considering only the constitutive specification of the actions. The annotation, highlighted by graphical conventions, accounts for all the regulative aspects, concerning both commitments and constraints. So the graph will include both legal states and states in which violations occur. Part of violations related to a state are, actually, just potential, depending on future behavior. Tendentiously they become violations if the interaction stops in that state. As usual in commitment protocols, interaction can, in fact, start/end in any state. In this case, highlighting the possible violations, therefore, amounts to alerting the user about a risk. More generally, the obtained graph is a tool that supports the analysis of a business protocol, by helping the identification of situations where it maybe necessary to perform some regimentation or enforcement.

The software exploits the *tuProlog*[1] interpreter and interprets a 2CL business protocol by means of a parser written in Java. The source code is available at the URL `http://www.di.unito.it/~alice/2CL`.

3 Modeling and Analysis of the Case Studies

Let us now introduce the two real-world case studies, that we use to test the framework. These two case studies are taken from highly regulated, though different, application domains. The former is the Guidelines on the protection of privacy and transborder flow of personal data [16], by the Organization for Economic Cooperation and Development, while the latter is the Markets in Financial Instruments Directive number 2004/39/CE [1], issued by the European Commission. The common characteristic is the need of integrating business protocols with new regulations. This integration requires a modification of the protocol, that is achieved by adding new actions, which correspond to new activities foreseen by the regulation, and also some temporal constraints that regulate the protocol enactment.

3.1 The OECD Guidelines case study

The aim of these guidelines is twofold. On the one hand, they aim at protecting data owners by preventing the violation of their fundamental rights. These violations can be caused, for instance, by the unauthorised use of personal data or the storage of inaccurate data. On the other hand, guidelines sustain also the international flow of personal data: the different nations, in fact, have different laws on personal data management. OECD Guidelines, by imposing additional and shared regulations, give guarantees on how personal data will be treated, increasing trust between countries and, thus, encouraging the flow of data.

For the sake of readability, we consider only three of the OECD Guidelines for data flow management:

- **Data Quality Principle**: states that data must be relevant, accurate, complete and up-to-date;

[1] `http://www.alice.unibo.it/xwiki/bin/view/Tuprolog/`

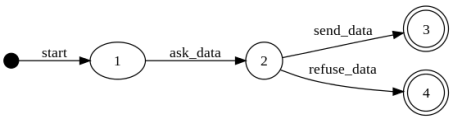

Figure 1. Reachability graph of the basic data flow protocol.

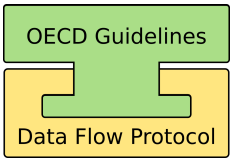

Figure 2. Grafting the OECD Guidelines in the data flow protocol.

- **Purpose Specification Principle**: states that the purposes for which the data are collected must be clearly specified and that the subsequent uses of data are limited to the fulfilment of such purposes (any change of purpose must have the consent of the data subject);
- **Use Limitation Principle**: states that data cannot be disclosed or forwarded without the consent of the data subject or unless imposed by law.

Pre-OECD data flow protocol. Before the definition of OECD Guidelines, different countries had different laws on privacy and the protection of data. The basic interaction for data flow between two countries is captured by the prodocol $DF = \langle Ro_{DF}, F_{DF}, A_{DF}, C_{DF} \rangle$ and shown in Figure 1: when some data are asked (*ask_data*) than the data controller (dc) can send them (*send_data*) or refuse to send it (*refuse_data*). In this model, we suppose the initial state is not empty but rather contains a conditional commitment $\mathsf{C}(dc,asker,asked\_data,sent\_data)$, by which the data controller dc, i.e. the person or the institute storing the data, binds to have the data sent in case someone ($asker$) asks for it. Therefor, Ro_{DF} is made of $asker$ and dc. The set of actions A_{DF} is $\{ask\_data, send\_data, refuse\_data\}$. Such actions are specified as:

(a) *ask_data* **means** *asked_data* **if** $\neg$*asked_data*.
(b) *send_data* **means** *sent_data*
if *asked_data* $\wedge$ $\neg$*sent_data* $\wedge$ $\neg$*refuse_data*.
(c) *refuse_data* **means** *refuse_data*, CANCEL(C(*dc*,*asker*,*sent_data*))
if *asked_data* $\wedge$ $\neg$ *sent_data* $\wedge$ $\neg$ *refuse_data*.

(a) encodes the power to ask for data and causes the activation of the initial conditional commitment; (b) causes the fulfillment of the commitment; (c) the refusal causes instead the canceling of the commitment. The policy by which the decision whether applying (b) or (c) is taken is up to the local laws of each country. The set of constraints C_{DF} is, instead, empty.

Grafting of OECD Guidelines. The adoption of OECD Guidelines requires the integration of the protocol with new actions. Specifically, the actions for checking the purpose, checking the accuracy (periodically or on demand) of data, and for notifying the data subject:

(d) *periodically_verify_accuracy* **means** *accuracy_verified*
if $\neg$*asked_data* $\wedge$ $\neg$*accuracy_verified*.
(e) *check_accuracy* **means** *accuracy_verified*
if *asked_data* $\wedge$ $\neg$ *accuracy_verified*.
(f) *verify_purpose* **means** *purpose_verified*
if *asked_data* $\wedge$ $\neg$*purpose_verified*.
(g) *notify_owner* **means** *owner_notified*
if *sent_data* $\wedge$ $\neg$*owner_notified*.

These are not sufficient, because we need to specify how the new regulation grafts into the original protocol (Figure 2). This is done with the help of temporal constraints:

(c1) $purpose\_verified \rightarrow\!\!\bullet\ sent\_data$
(c2) $accuracy\_verified \rightarrow\!\!\bullet\ sent\_data$
(c3) $sent\_data \bullet\!\!\rightarrow owner\_notified$
(c4) $purpose\_verified \rightarrow\!\!\bullet\ refuse\_data$
(c5) $accuracy\_verified \rightarrow\!\!\bullet\ refuse\_data$

The grafting is simple: accuracy and purpose must be verified ($accuracy\_verified$ and $purpose\_verified$ must appear in the social state) before sending the data ($sent\_data$) as well as before refusing to send the data ($refuse\_data$)). Finally, constraint $c3$ capture the condition that once data are sent, the data controller must notify (*notify_owner*) the data subject ($owner$).

More formally, these guidelines are represented by the tuple $G = \langle Ro_G, F_G, A_G, C_G \rangle$, where $Ro = \{dc, owner\}$, facts, actions, and constraints are those listed above. The grafting $DF \uplus G$, shown intuitively in Figure 2, is formally defined as follows: $\langle Ro_{DF} \cup Ro_G, F_{DF} \cup F_G, A_{DF} \cup A_G, C_{DF} \cup C_G \rangle$.

Figure 4 shows a dependency graph between DF and G. Dashed arrows denote temporal dependencies that are encoded by constraints. Notice that G actions may depend on actions of DF (solid arrows) but the opposite never happens. This characteristic is important to correctly model, similarly to what happens in the object-oriented paradigm when extending a class, the introduction of new regulations into a pre-existing reality which preserves the features of the extended protocol.

Analysis of the business protocol. The resulting set of legal and illegal possible executions of $DF \uplus G$ is reported in Figure 3 (the graph is the produced by the

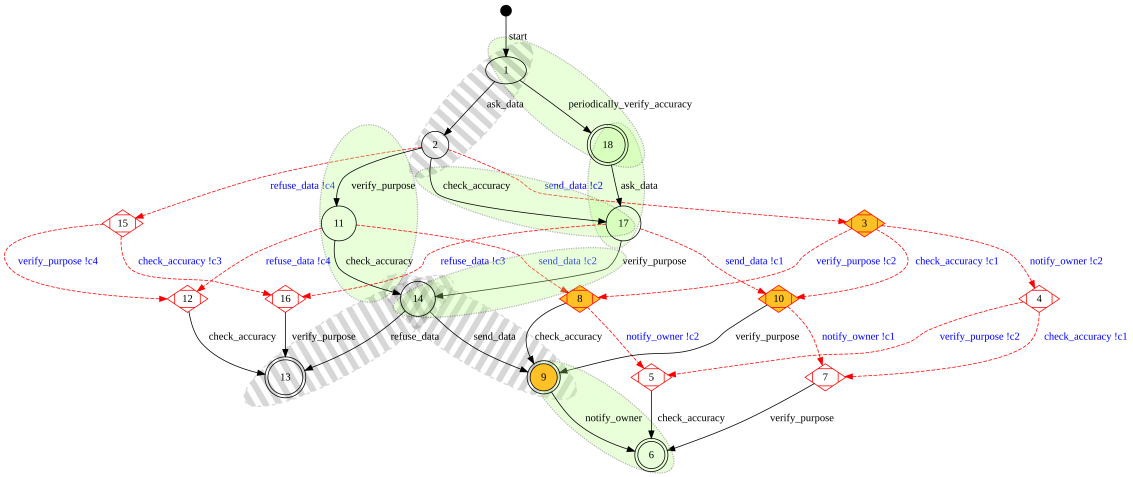

Figure 3. Reachability graph for the Data Flow protocol extended with OECD Guidelines.

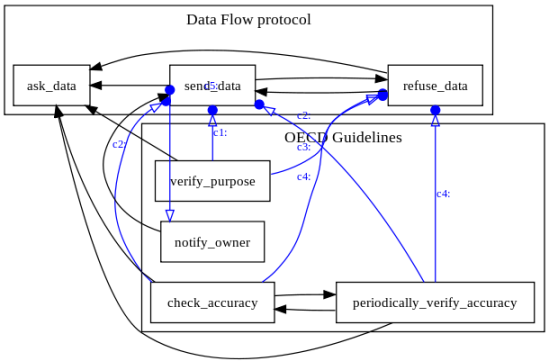

Figure 4. Data Flow protocol and OECD Guidelines dependency graph.

extended commitment machine that we implemented). The legal executions are highlighted in green and gray (the highlightning is added by hand). Notice how the actions of the OECD Guidelines (highlighted in green) are immersed in the original data flow protocol (highlighted in gray). The colors also help to highlight the composition of the regulations. All other paths amount to possible violations, as expected since we aggregated actions that have no mutual preconditions. More in details, each state in the graph represents a possible configuration of the social state. Arrows correspond to actions and are directed. The source is a state where the "if" condition of the action labelling the arc holds. The target is, instead, the state obtained by applying the meaning of the executed action to the source state. States that are drawn as diamonds with an incoming red arrow (e.g. states 3 and 4) represent the fact that a *before* constraint, or a *cause* constraint or their negation has been violated. Some states are yellow (e.g. 8 and 9), the meaning (independently from the shape) is that some *response* constraints or a *cause* constraint[2] are not fulfilled yet. White states with a single outline (e.g. 1 and 2), independently from the shape, mean that there are some active commitments (not discharged, released or cancelled). States with a double outline (independently from the shape) do not contain any active commitment (e.g. 5, 9, 18). Final acceptable states are, therefore, white and are denoted by double circles to express that there is no active commitment and all constraints are satisfied (e.g. 6, 13, 18). A legal path connects the initial state with one of the final states and is made by all black arrows (black arrows denote legal moves). For instance, the execution $(ask\_data, verify\_purpose, check\_accuracy, send\_data, notify\_owner)$ is legal. Instead, $(ask\_data, check\_accuracy, send\_data, verify\_purpose, notify\_owner)$ contains a violations. Constraint (c1) is not respected by $send\_data$. As Figure 3 shows, *check_accuracy* and *verify_purpose* must be executed after *ask_data* but since there is no relation between them, they can be executed in any order. The protocol, however, does not need to specify explicitly each of the interleavings. Moreover, state 9 in the figure, which is reached after sending the data to the asker by executing legal paths, is not final as it would have, instead, been before the introduction of OECD Guidelines: in this state constraint (c3) is not yet satisfied because the data subject has not been notified yet. Notice that we could not achieve the same result by using action preconditions nor by adding

[2] *Cause* is the conjunction of *before* and *response*, see Table 1.

new commitments as an additional effect of *send_data*. The former solution fails because preconditions do not compel agents to execute applicable actions. The latter solution, instead, modifies an action belonging to the basic data flow protocol making it fit the new protocol. However, in this case, it is difficult to foresee the costs of the modification, for instance: that action might be used in different business protocols, normed by different regulations; the implementation of that action might be expensive to update (especially because it would be embedded in a complex software system). One could think to bypass the problem by adding the new commitment as an effect of one of the new actions added by the grafted regulation. This, however, generally makes no sense because the commitment at issue is not naturally part of the constitutive specification of those actions.

3.2 The MiFID case study

The MiFID case study is more complex and underlines more strongly the need of a model which accounts explicitly also for a regulative specifications. One of the main concerns of the directive is the protection of the clients of financial service agencies, thereby it introduces new regulations that financial services must follow. Specifically, in this work we model the regulation that applies to the offer of investment services off-site. This is the case when a bank promotes and sells financial products with the help of external collaborators (called "tied agents" or intermediaries). According to MiFID the proposal of products and the definition of a contract must comply to a set of constraints:

- **Identification**: the client must be identified by an identity card or equivalent document;
- **Qualification**: the intermediary supplies all the foreseen documentation about his/her professional qualification and the rules that he/she must stick to;
- **Profiling**: the intermediary must profile the client, gathering information about the balance sheet, knowledge about financial subjects, investment aims. This phase requires the filling the MiFID form, which explicitly specifies the resulting category of client and which is to be signed also by the client;
- **Selection**: the proposed financial products must agree with the client's profile. This requires that financial products are classified w.r.t. the different client profiles;
- **Evaluation**: the proposal is evaluated through a simulation: if it is adequate an order is filled and signed both by the client and by the intermediary, otherwise the product is discarded;
- **Verification**: the documentation is sent to the investment trust, which must check that there are no errors or missing data. In this case, the documentation is corrected and sent back to the intermediary, otherwise the contract is stamped and sent to the client;
- **Withdrawal**: The client can decide to cancel an order.

MiFID grafts onto the previously existing financial product sales protocols. What happens if an intermediary buys a financial product for a client, violating some of the constraints imposed by MiFID? The *sale is valid*, the client results to be the owner of the product. This happens because MiFID does not define *sales* (sales are defined by a different regulation) but dictates how the interaction with the client should be carried on by adding a new layer of regulations on top of existing ones. So, the violation of some constraint does not affect the sale directly but creates both a *risk of sanction* and an *risk of exposure* for the intermediary. This is witnessed by a sentence by the Italian Supreme Court (*Cassazione civile a sezioni unite*, num. 26724 and 26725 [10]) which decided that in case of violations, like the above, if the client was economically damaged he/she can ask for a compensation and, in the most serious cases, for the cancellation of the contract between the client and the intermediary. This will be transparent to the seller, who will not be involved in the quarrel and will have no consequences (specifically he/she will not have to give money back). We will show that our business protocols precisely capture this situation, thanks to the separation of the constitutive and regulative parts.

Pre-MiFID sale business protocol. As for the OECD case study, let us begin by presenting a sales business protocol, that could be used before the introduction of MiFID. The actions involve three parties: an investor (inv), an intermediary (the financial promoter fp), and a bank ($bank$). This protocol foresees an initial state containing a commitment, $\mathsf{C}(fp, inv, invested)$, from the intermediary to the investor to find a good investment. By the action *propose_solution*, the intermediary presents a selected financial product to the investor. The proposal is characterized by a risk level, and can be rejected (*reject_proposal*) or accepted (*sign_order*). In the first case, the commitment of the intermediary is released. When the order is signed, the investor commits to the bank to respect the purchase contract ($\mathsf{C}(inv, bank, contract\_ended)$). The bank is expected to countersign the contract (it does it by the action $countersign\_contract$, which creates a commitment $\mathsf{C}(bank, inv, executed\_order)$ from the bank to the investor to actually execute the order), and send a copy of it to the investor ($send\_contract$). When the bankcountersigns the contract, the initial commitment of the intermediary is discharged. Moreover, the bank is also expected to $notify$ the intermediary the contract was countersigned. The notification guarantees to the intermediary that everything was

fine and he/she will get his/her commission. This should be done after the contract was sent but before the natural end of the contract. The natural end of the contract is captured by the action *end* which causes the discharge of the pending commitments of the investor and of the bank.

(a) *propose_solution* **means** *proposed_RiskL*
if $\neg$*proposed_RiskL* $\wedge$ $\neg$*rejected_proposal*.
(b) *reject_proposal* **means** *rejected_proposal*,
RELEASE(C(*fp*, *inv*, *invested*))
if $\neg$*accepted_proposal* $\wedge$ *proposed_RiskL* $\wedge$ $\neg$*rejected_proposal*.
(c) *sign_order* **means** CREATE(C(*inv*, *bank*, *contract_ended*)),
accepted_proposal, *order_signed*
if $\neg$*order_signed* $\wedge$ *proposed_RiskL* $\wedge$ $\neg$*rejected_proposal*.
(d) *countersign_contract* **means** *contract_countersigned*,
CREATE(C(*bank*, *inv*, *executed_order*)), *invested*
if *order_signed* $\wedge$ *proposed_RiskL* $\wedge$ $\neg$*contract_countersigned*.
(e) *send_contract* **means** *contract_sent*
if $\neg$*contract_sent* $\wedge$ *contract_countersigned*.
(f) *notify* **means** *notified*
if *contract_countersigned* $\wedge$ $\neg$*notified* $\wedge$ $\neg$*contract_ended*
$\wedge$ $\neg$*contract_abort*.
(g) *end* **means** *executed_order*, *contract_ended*
if *contract_sent* $\wedge$ $\neg$*contract_ended* $\wedge$ $\neg$*contract_abort*.

The protocol also includes the a few temporal regulations concerning the notification:

(c1) $notified \rightarrow\!\bullet\ contract\_ended$
(c2) $contract\_sent \bullet\!\rightarrow notified$

These constraints give the bank the freedom to choose whether notifying the intermediary before sending the investor copy of the contract, or the other way around. In the latter case, (c2) imposes that after the contract was sent, the bank must perform the pending notification. Notice that, substituting the above constraints with action preconditions (in particular that the contract was sent before notifying the intermediary) does not allows this flexibility and has the further disadvantage that it does not compel the bank to notify the intermediary. In fact, preconditions allow the identification of the executable actions but cannot actively cause their enactment.

Grafting of MiFID. This directive introduces new regulations that financial services must follow in their interaction with the client, so as to protect the investor. Its application requires the enrichment of the business protocol with new, specific actions, aimed at: identifying the investor and supplying the foreseen documentation (*interview*), profiling the investor (*profile*) and assigning him/her a risk category (*investor_classified*), classifying the financial products according to the possible risk levels ($classify$). In the profiling process, the intermediary commits to evaluate, with the help of a simulation, financial products in order to identify one that suits the client (C($fp, inv, evaluation$)). Such evaluation (*fi_evaluation*) commits the intermediary to propose a product with a risk level, that is adequate to the investor's profile (C($fp, inv, proposed\_RiskL$)). This commitment is crucial to satisfy the MiFID's requirements. A solution that is not adequate can be discarded (*fi_discard*). In this case the intermediary's commitments will be canceled. The withdrawal phase of MiFID is implemented by the action *withdraw*, which concludes a contract by aborting it and by releasing the commitment from the bank to execute the order. The selection and evaluation of a new proposal are modeled as a new interaction.

(h) *interview* **means** *investor_identified*, *document_supplied*
if $\neg$*investor_identified* $\wedge$ $\neg$*contract_abort* $\wedge$ $\neg$*contract_ended* $\wedge$
$\neg$*rejected_proposal* $\wedge$ $\neg$*fi_discarded*.
(i) *profile* **means** CREATE(C(*fp*, *inv*, *evaluation*)), *investor_classified*
if $\neg$*investor_classified* $\wedge$ *investor_identified* $\wedge$ $\neg$*contract_ended* $\wedge$
$\neg$*contract_abort* $\wedge$ $\neg$*rejected_proposal* $\wedge$ $\neg$*fi_discarded*.
(j) *classify* **means** *classified*
if $\neg$*classified* $\wedge$ $\neg$*contract_abort* $\wedge$ $\neg$*contract_ended* $\wedge$
$\neg$*rejected_proposal* $\wedge$ $\neg$*fi_discarded* $\wedge$ $\neg$*proposed_RiskL*.
(k) *fi_evaluation* **means** CREATE(C(*fp*, *inv*, *proposed_RiskL*)), *evaluation*
if *classified* $\wedge$ *investor_identified* $\wedge$ $\neg$*evaluation* $\wedge$ $\neg$*contract_abort* $\wedge$
$\neg$*contract_ended* $\wedge$ $\neg$*rejected_proposal* $\wedge$ $\neg$*fi_discarded*.
(l) *fi_discard* **means** *fi_discarded*, CANCEL(C(*fp*, *inv*, *invested*)),
CANCEL(C(*fp*, *inv*, *proposed_RiskL*))
if *evaluation* $\wedge$ $\neg$*proposed_RiskL* $\wedge$ $\neg$*contract_abort* $\wedge$
$\neg$*contract_ended* $\wedge$ $\neg$*fi_discarded*.
(m) *order_verification* **means** *order_verified*,
CREATE(C(*bank*, *inv*, *executed_order*))
if $\neg$*order_verified* $\wedge$ *order_signed*.
(n) *withdraw* **means** *contract_abort*,
RELEASE(C(*bank*, *inv*, *executed_order*)),
CANCEL(C(*inv*, *bank*, *contract_ended*))
if *contract_sent* $\wedge$ $\neg$*contract_ended* $\wedge$ $\neg$*contract_abort*.

These actions alone are not enough to implement the directive. Actions *(h–l)* should be executed before the actual sale occurs, while *(m)* and *(n)* complete the sales process (see Section 3.2). This could be done by modifying the action that implements a sale but this is not in the powers of the MiFID regulation, as we have explained. The integration of the new directive with the previous regulation is, therefore, done by means of a set of 2CL constraints, which relate facts and commitments which can appear in the social state and, in particular, those pertaining MiFID and those pertaining sales. Hereafter are reported the 2CL constraints that capture the most relevant dictates of the MiFID regulation:

(c3) C($fp, inv, invested$) $\bullet\!\rightarrow$ $investor\_identified \wedge$
$document\_supplied$
(c4) $investor\_classified \rightarrow\!\bullet$ C($fp, inv, propose\_riskL$)
(c5) $evaluation \wedge \neg fi\_discarded \rightarrow\!\bullet\ proposed\_RiskL$
(c6) $order\_verified \rightarrow\!\bullet\ contract\_countersigned$

(c3) states that once the intermediary took the commitment to serve the investor, he/she must have the investor identified and must supply the necessary documentation to him/her. (c4) expresses the fact that before committing to propose a solution with a certain degree of risk, the investor must have been classified. (c5) states that before proposing a financial product it is necessary to have it positively

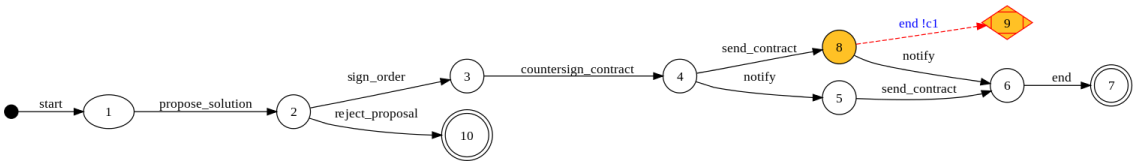

Figure 5. Execution graph of the pre-MiFID sale regulation.

evaluated by the simulation. Finally, before the contract is countersigned by the bank, the data of the order must have been verified. The grafting of MiFID inside the sales protocol is given by the union of the respective components, in particular: actions (a) – (g) with (h) – (n); constraints (c1) – (c2) with (c3) – (c6).

Analysis of the business protocol. The graph of the possible executions is too big to be included in a readable format. At `http://www.di.unito.it/~alice/2CL` it is possible to find the complete examples and all figures related to the examples in the paper. The designer, by analysing the graph, can identify the points where it could be helpful to intervene to reduce the possible violations, for instance, by applying enforcement policies or by regimenting some steps. For example, one action on which it would make sense to intervene is *propose_solution*. The reason is that most of illegal paths start from a bad use of this action. By adding the condition $\neg$*fi_discarded* $\wedge$ *evaluation*, we obtain the graph in Figure 6. Of course, this choice depends on many factors (e.g. the cost of the implementation of the prospected solution, or the time needed to update the financial services office's software) that are out of the scope of the directive.

4 Conclusion and Related Works

We have proposed a declarative approach to business protocol specification that extends [20] by explicitly including 2CL temporal regulations. We implemented a tuProlog extended commitment machine which was applied to the MiFID and the OECD case studies, whose output allows the analysis of the business protocol and of possible violations. Indeed, in these contexts it is important to define mechanisms for detecting possible violations and decide about possible regimentations/enforcements. The analysis of the case studies proved the advantages of the declarative approach, that we have proposed, underlining, in particular, the usefulness of the modular composition of protocols, hoped for in [14], which is realized in our framework thanks to the introduction of the grafting operation. Another advancement w.r.t. the literature is that we developed an analysis tool, which supports the business analyst in performing task like: understanding the impact of new regulations on the business protocol or deciding about enforcement policies or regimentation.

We are currently working at the formalization of an "extends" operator which, similarly to "extension" in object orientation, allows a more sophisticate composition of protocols.

Telang and Singh [20] proposed a commitment-based approach to representing business protocols and identified a set of common patterns of interaction, that can be used by the business analyst. Along this line, also [5] proposes commitment patterns that capture common business patterns, showing which robustness requirements are met by each of them. These requirements are supposed to guide the protocol designer in the selection and composition process. Concerning composition, [24] proposes temporal operators to compose the data flow in a commitment-based approach. Our proposal extends the ones above by enriching the protocol with expressive temporal constraints. This is an added value in the modeling of business interactions because it enables the embedding of regulations that stratify along time.

Recently, many works, like [6, 12], focused on the problem of verifying the compliance of a business process to a body of norms. This issue is different in that the business process is rigidly modeled as a (YAWL or BPM) workflow, and the verification aims at checking if this process strictly respects the norms, providing, in some cases, a yes or no answer and, in some others, a degree of compliance.

Another interesting work, which however does not tackle business protocols and regulations, is [18] by Sergot. This work discusses the relationship between the norms that govern a single agent with those that express a designer's view on what overall system behaviors are deemed to be legal. This proposal foresees two models that are to be compared, and to this aim a colored labeled transition system is used. The main difference w.r.t. our proposal is that in Sergot's approach the focus is on verifying the behavior of a single agent against a global model. This is a complementary task w.r.t. the one we face. It would be interesting to study how to combine the two approaches in order to supply a complete toolkit to the business analyst.

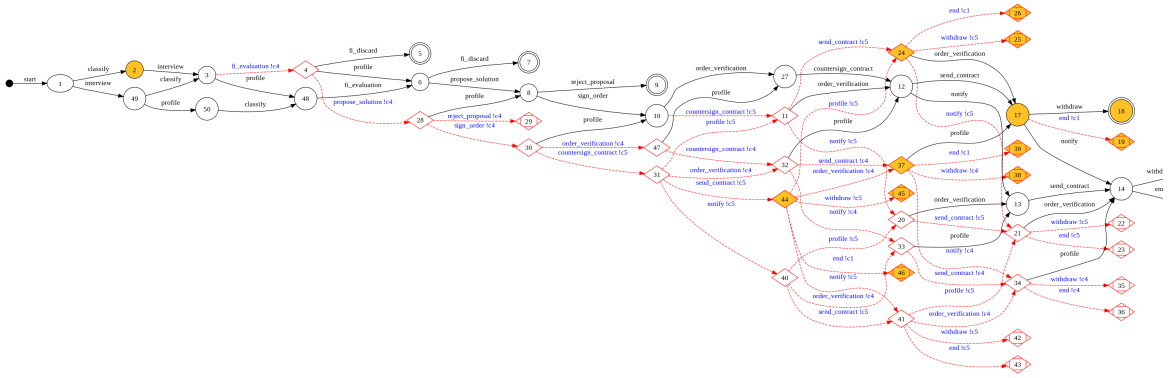

Figure 6. MiFID with regimented *propose_solution*.

Acknowledgements. This research was partially funded by "Regione Piemonte" through the project ICT4LAW.

References

[1] Directive 2004/39/EC of the European Parliament and of the Council of 21 April 2004 on markets in financial instruments. *Official Journal of the European Union*, L145:1–44.

[2] M. Baldoni, C. Baroglio, I. Brunkhorst, N. Henze, E. Marengo, and V. . Patti. Constraint Modeling for Curriculum Planning and Validation. *International Journal of Interactive Learning Environments*, 19(1):83–123, 2011.

[3] M. Baldoni, C. Baroglio, and E. Marengo. Behavior-Oriented Commitment-based Protocols. In *Proc. of ECAI*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 137–142. IOS Press, 2010.

[4] M. Baldoni, C. Baroglio, E. Marengo, and V. Patti. Constitutive and Regulative Specifications of Commitment Protocols: a Decoupled Approach. *ACM Trans. on Int. Sys. and Tech., Spec. Iss. on Agent Communication*, 2011. To appear.

[5] A. K. Chopra and M. P. Singh. Specifying and applying commitment-based business patterns. In *Proc. of AAMAS*. IFAAMAS, 2011.

[6] D. D'Aprile, L. Giordano, V. Gliozzi, G. Martelli, A.and Pozzato, and D. Theseider Duprè. Verifying business process compliance by reasoning about actions. In *Proc. of CLIMA*, pages 99–116. Springer, 2010.

[7] M. El-Menshawy, J. Bentahar, and R. Dssouli. Verifiable Semantic Model for Agent Interactions Using Social Commitments. In *LADS*, volume 6039 of *LNCS*, pages 128–152. Springer, 2010.

[8] E. A. Emerson. Temporal and Modal Logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 995–1072, 1990.

[9] N. Fornara and M. Colombetti. A Commitment-Based Approach To Agent Communication. *Applied Artificial Intelligence*, 18(9-10):853–866, 2004.

[10] G. Gibilaro. Cassazione Civile Sentenza, Sez. SS.UU., 19/12/2007, n. 26724 e 26725. Intermediazione finanziaria, nullità del contratto e risarcimento del danno, 2007.

[11] L. Giordano, A. Martelli, and C. Schwind. Specifying and Verifying Interaction Protocols in a Temporal Action Logic. *Journal of Applied Logic*, 5(2):214–234, 2007.

[12] G. Governatori. Law, Logic and Business Processes. In *Proc. of Requirements Engineering and Law, RELAW 2010*, pages 1–10. IEEE, 2010.

[13] A. J. I. Jones and M. Sergot. *On the characterization of law and computer systems: the normative systems perspective*, pages 275–307. John Wiley & Sons, Inc., 1994.

[14] T. Miller and J. McGinnis. Amongst first-class protocols. In *Proc. of Eng. Societies in the Agents World VIII*, volume 4995 of *LNCS*, pages 208–223. Springer, 2008.

[15] M. Montali. *Specification and Verification of Declarative Open Interaction Models: a Logic-Based Approach*, volume 56 of *LNBIP*. Springer, 2010.

[16] Organisation for Economic Co-operation and Development. OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data. Available on-line, 1980. http://www.oecd.org/.

[17] J. Searle. *The construction of social reality*. Free Press, New York, 1995.

[18] M. Sergot. Action and agency in norm-governed multi-agent systems. In *Engineering Societies in the Agents World VIII*, pages 1–54. Springer, 2008.

[19] M. P. Singh. An Ontology for Commitments in Multiagent Systems. *Art. Int. and Law*, 7(1):97–113, 1999.

[20] P. R. Telang and M. P. Singh. Abstracting Business Modeling Patterns from RosettaNet. In *Service-Oriented Computing: Agents, Semantics, and Engineering*, 2010.

[21] M. Winikoff, W. Liu, and J. Harland. Enhancing Commitment Machines. In *In Proc. of DALT 2004*, volume 3476 of *LNCS*, pages 198–220. Springer, 2004.

[22] P. Yolum and M. P. Singh. Commitment Machines. In *Intelligent Agents VIII, Proc. of ATAL*, volume 2333 of *LNCS*, pages 235–247. Springer, 2001.

[23] P. Yolum and M. P. Singh. Designing and Executing Protocols using the Event Calculus. In *Proc. of the 5th Int. Conf. on Autonomous Agents*, pages 27–28, 2001.

[24] N. V. Desai, A. K. Chopra, M. Arrott, B. Specht, and M. P. Singh. Engineering Foreign Exchange Processes via Commitment Protocols. In *IEEE Int. Conf. SCC 2007*, pages 514–521.