Autonomous Agents and Multi-Agent Systems manuscript No.
(will be inserted by the editor)

Automated Multi-level Governance Compliance Checking

Thomas C. King · Marina De Vos · Virginia Dignum · Catholijn M. Jonker · Tingting Li · Julian Padget · M. Birna van Riemsdijk

Received: date / Accepted: date

Abstract An institution typically comprises constitutive rules, which give shape and meaning to social interactions and regulative rules, which prescribe agent behaviour in the society. Regulative rules guide social interaction, in particular when they are coupled with reward and punishment regulations that are enforced for (non-) compliance. Institution examples include legislation and contracts. Formal institutional reasoning frameworks automate ascribing social meaning to agent interaction and determining whether those actions have social meanings that comprise (non-) compliant behaviour. Yet, institutions do not just govern societies. Rather, in what is called *multi-level governance*, institutional designs at lower governance levels (e.g., national legislation at the national level) are governed by higher level institutions (e.g., directives, human rights charters and supranational agreements). When an institution design is found to be non-compliant, punishments can be issued by annulling the legislation or imposing fines on the responsible designers (i.e., government). In order to enforce multi-level governance, higher governance levels (e.g., courts applying human rights) must check lower level institution designs (e.g., national legislation) for compliance; in order to avoid punishment, lower governance levels (e.g., national

The final publication is available at Springer via http://dx.doi.org/10.1007/s10458-017-9363-y

Thomas C. King
Lancaster University
t.c.king@lancaster.ac.uk

Marina De Vos
University of Bath
mdv@cs.bath.ac.uk

Virginia Dignum
Delft University of Technology
M.V.Dignum@tudelft.nl

Catholijn M. Jonker
Delft University of Technology
C.M.Jonker@tudelft.nl

Tingting Li
Imperial College London
tingting.li@imperial.ac.uk

Julian Padget
University of Bath
j.a.padget@bath.ac.uk

M. Birna van Riemsdijk
Delft University of Technology
m.b.vanriemsdijk@tudelft.nl

governments) must check their institution designs are compliant with higher-level institutions before enactment. However, checking non-compliance of institution designs in multi-level governance is non-trivial. In particular, because institutions in multi-level governance operate at different levels of abstraction. Lower level institutions govern with concrete regulations whilst higher level institutions typically comprise increasingly vague and abstract regulations. To address this issue, in this paper we propose a formal framework with a novel semantics that defines compliance between concrete lower level institutions and abstract higher level institutions. The formal framework is complemented by a sound and complete computational framework that automates compliance checking, which we apply to a real-world case study.

Keywords Institutions, Normative reasoning, Multi-level Governance

1 Introduction

Institutions (e.g., legislation) guide societies towards subjectively-ideal and coordinated behaviour. An institution, such as the written law, comprises regulations imposed on agents taking part in the governed society, coupled with the means to detect compliance and impose regulations that reward and punish agents for (non-)compliance. An institution comprises constitutional and regulative rules. Constitutional rules define concepts, for example "making an electronic bank transfer counts-as payment". Regulative rules impose obligations and prohibitions to instantiate the defined concepts, for example "you are obliged to make a payment". Institutions, comprising interacting constitutive and regulative rules, need to be understood in order to be applied to the governed society. Hence, increasingly institutional reasoning is formalised and computerised with automated normative and institutional reasoning frameworks (see [3] for a review). Such formal institutional reasoning frameworks support governing bodies in automatically penalising agents as well as individual agents in understanding their legal duties.

However, institutions are not typically written in a vacuum. Rather, institution designs are constrained and regulated by higher level governing bodies. This is what is called *multi-level governance* [45]. In multi-level governance, legislators design institutions comprising rules and regulations, but whose design is also subject to regulation. For example, in 2006 the European Union issued the Data Retention Directive [22] for harmonising member states' data retention regulations. In 2009 the UK implemented the directive with the Data Retention Regulations [74] in order to avoid being fined. Yet, in 2014 the European Court of Justice ruled [21] that the EU directive was non-compliant with the EU's Charter of Fundamental Rights [23], and consequently annulled the EU's Data Retention Directive. We will use this case throughout, referring to the Charter of Fundamental rights as the EU-CFR, the EU's Data Retention Directive as the EU-DRD, and the UK's implementing Data Retention Regulations as the UK-DRR. The main point is that multi-level governance exposes legislators to the risk of punishment for non-compliant institution designs and burdens a judiciary with determining compliance of institution designs.

So far, institutional reasoning frameworks have focussed on single-level societal governance. Typically, automated institutional reasoning deals with regulations operating at the level of institutions governing agents and/or corporations. For example, the UK-DRR [74] obliges communications providers to store communications metadata. However, there lacks formalisation for cases where regulations themselves are regulated by higher level institutions in multi-level governance. For example, how EU directives govern national legislation but where EU directives are in turn governed by human rights charters. In this paper we look at how lower level institutions themselves are regulated by higher level institutions.

In particular, we look at increasingly abstract regulations at higher levels of governance, which govern more concrete regulations at lower levels of governance. Such abstraction sets multi-level governance apart from single-levelled governance of societies. In multi-level governance at the highest-level, such as human rights charters, regulations are intentionally abstract and open to interpretation. Such abstract regulations provide many ways in which to (non-) comply. At a lower level, such as EU directives, regulations are more concrete and less open to interpretation. At the lowest level, such as national or sub-national legislation, regulations are concrete and should have the least ambiguity. Despite the

differences in abstraction between levels, each level's institution design must somehow be demonstrated to be compliant with relatively more abstract regulations at higher levels.

To give an example, the EU-CFR [23] contains vague regulations requiring that people's private and family life is respected. The EU-DRD [22] contains a more concrete regulation requiring communications service providers (e.g., internet service providers) to store people's communications metadata (e.g., a phonecall's time and place) within a fixed time frame. The EU-CFR governs EU directives. Hence, the EU-DRD's communications metadata regulation must be shown to be compliant with the EU-CFR's more abstract right to a private and family life. At the same time, the EU-DRD itself governs the design of institutions, namely member states' legislation. Member states must implement the directive in a compliant way in order to avoid fines. The directive gives some scope for member states to implement the legislation differently, allowing the data retention period to be between 6 and 24 months. The UK-DRR [74] is more concrete and must be shown to ensure communications metadata to be stored within the required time frame, no shorter and no longer. In fact, the UK-DRR does just that, concretely requiring that communications metadata is stored for 13 months which complies with the abstract requirement of the directive to store data between 6 and 24 months.

In this paper, we give a rigorous formal account and automate checking of compliance in multi-level governance between concrete lower level and abstract higher level institutions with a novel framework. Our framework provides a representation for defining institutions and their multi-level governance relationship. A semantics defines the regulatory outcomes of each institution in different (potentially hypothetical) contexts.

Specifically, a semantics re-interprets concrete regulations at lower levels in terms of their more abstract meaning with respect to higher level institutions. Taking concrete regulations and determining their abstract interpretation is based on Searle's constitutive institutional rules, which define the links between concrete and abstract concepts. By interpreting concrete regulations in terms of their abstract meaning, it is determined if the concrete regulations are (non-)compliant with the abstract regulations in higher level institutions. To give an example, the EU-DRD [22] requires member states to store communications metadata. According to the semantics we infer that storing communications metadata without someone's consent is, abstractly, unfair data processing. Since the EU-CFR prohibits unfair data processing [23, Art. 8.2] the EU-DRD's more concrete regulations are determined to be non-compliant. This paper contributes a framework for semantically determining if concrete regulations at lower levels of governance are compliant with more abstract regulations at higher levels of governance.

This paper continues by providing the conceptual background of the framework in Section 2. The approach we take in formalising multi-level governance compliance is described in Section 3. The new formal framework is presented in Section 4. A practical approach to multi-level governance reasoning is provided with a computational framework presented in Section 5. The computational framework provides a sound and complete translation from the formal framework to an executable logic program. An implementation automates the translation between high-level institution specifications and a logic programming language program, which in turn automates compliance checking as we demonstrate for a real-world case study. At the end of this paper we compare our framework to related work in Section 6. We conclude with reflections and avenues for future work in Section 7.

2 Governance Concepts

2.1 Institutions

An institution, alternatively called a normative system, is in our view a specification of rules and regulations that guide agents in a Multi-Agent System (MAS) towards ideal and coordinated behaviour. An institution is operationalised by interpreting and applying its rules and regulations on the agents acting in the MAS that the institution governs. The interpretation process involves assessing how agents in the MAS are behaving and the MAS' state in order to see which rules and regulations apply and when.

We view an institution's rules as being classified into two types in line with existing formal work. To quote Searle [72]:

> "Some rules regulate antecedently existing activities. For example, the rule 'drive on the right-hand side of the road' regulates driving; but driving can exist prior to the existence of that rule. However, some rules do not merely regulate, they also create the very possibility of certain activities."

In other words two rule types exist in an institution, those that ascribe facts such as social activities and those that prescribe facts, respectively known as constitutive rules and regulative rules (norms) according to Searle's philosophy of institutions [72], formal theories of institutions [11, 13, 12, 35] and legal scholarship [9].

Searle's [70, 73] constitutive counts-as rules establish institutional facts (e.g., that an agent possesses money) from physical/brute facts (e.g., that an agent possesses a piece of paper commonly viewed as money). Regulatory rules, which we also call norms, specify how agents or a system should behave (e.g., obliging an agent to pay for goods) and/or what the state of affairs should be.

In our view (following preceding work on e.g., InstAL [13]) operationalising an institution involves interpreting institutional rules of both types. Through institutional rule interpretation, a social reality is established comprising institutional facts and various deontic positions such as obligations. Ultimately, determining whether agents and society are behaving in a compliant way is based on whether the created social reality conforms to the prescriptions imposed by norms. We will now describe in detail constitutive rules and norms.

2.1.1 Constitutive Rules

Constitutive rules [70, 73] construct a social reality, where things such as 'money' and 'personal data' exist, from a brute reality where physical brute facts exist independently of an institution or society (e.g., that there is a piece of paper that looks like money, or that an analog signal has been sent down a wire in what we might call personal data communication). These constitutive rules have the now ubiquitous counts-as form of "some brute or institutional fact A counts-as an institutional fact B in a social context C". For example, "storing communications metadata counts-as storing personal data in the context that the metadata is about the communications of a person". Searle argues that such constitutive rules ascribe an institutional meaning in the form of an institutional fact, the 'B' in such a rule (e.g., storing personal data), to an 'A' in such a rule which is either a brute fact or another more concrete or basic institutional fact (e.g., storing communications meta-data). Such rules are conditional on a social context, which is a part of the social reality built by such counts-as rules (e.g., the context that someone is a 'person' exists whenever an agent that exists in the brute reality is ascribed the status of 'personhood' by a constitutive rule).

A similar example is "storing communications content data counts-as storing personal data in the context that it is a person's communications being stored". In both of these examples, content data and metadata are also institutional facts that are defined by other constitutive rules as either referring to a more concrete institutional fact or a brute fact. Ultimately, through a chain of derivations, all institutional facts exist because of constitutive rules that ascribe an institutional fact as being constituted by brute facts. It is a bit tricky to exemplify a counts-as rule that ascribes an institutional fact from a brute fact. The reason being, any time we try to refer to a brute fact we will be using words from a language, and since language is a 'base institution' these words we use will always be institutional facts (to give Searle's example [73] "It seems intuitively right to say that you can have language without money, but not money without language."). Hence, we will use the terms "the thing we call X" or "the observable event X" to represent a brute fact distinct from the institutional fact/symbol X that refers to the brute fact. So, for example, meta-data is an abstract institutional fact that refers to a brute fact according to a constitutive rule such as "the thing we call storing communications metadata counts-as storing communications metadata". In

other words, institutional facts are ascribed as being constituted by brute facts, giving the physical reality a social meaning.

These examples are about ascribing abstract institutional events. But, constitutive rules also establish the institutional properties that *hold* at a particular point in time. For example, from an institutional event that occurs, the establishment of an institutional property that *holds* is ascribed "someone signing a form stating a communications provider is allowed to store their personal data counts-as establishing that the person has consented to personal data storage". This means that the establishment of an institutional 'consent' fact in a *state* is a special meaning ascribed to the event where the agent signs a consent form.

One final example is "storing personal data counts-as non-consensual data processing in the context that the person who the data concerns has not consented". In this final example we can see that by transitivity it follows that from storing metadata (which is personal data) in the context that the person who it concerns has not consented we derive non-consensual data processing from the aforementioned abstracting constitutive rules. In conclusion, constitutive rules establish abstract institutional events and properties from more concrete brute events or institutional events/properties. Constitutive rules build an abstract institutional reality of institutional facts from brute facts, in turn the institutional reality can be further abstracted according to constitutive rules.

It is important to note that counts-as rules make institutional facts possible. As Searle argues [72]:

> "[...] institutional facts exist only within systems of constitutive rules. The systems of rules create the possibility of facts of this type; and the specific instances of institutional facts such as the fact that I won at chess or the fact that Clinton is president are created by the application of specific rules [...]"

In other words, a status or institutional fact assigned to a particular brute or institutional fact exists *only* because a constitutive rule makes it so. For example, 'personal data' cannot exist in a social reality without a constitutive rule ascribing it as being a status of a more concrete brute or institutional fact (e.g., meta-data). An important distinction must be made with the physical reality, taking a classical example often used for explaining abduction. We may know that "if it rains then the grass becomes wet", however the grass being wet is not a fact introduced by the rule, rather the rule is representative of a predicted causal relationship in a pre-existing physical reality. Consequently, if it has not rained, that does not mean that the grass is not wet, perhaps the grass can become wet by some other means (e.g., a sprinkler is turned on). In comparison, if we only have the two constitutive rules "communications meta-data counts-as personal data" and "communications content counts-as personal data" then the social meaning of data being personal can only be attributed to meta-data or content data, since the constitutive rules introduce the fact of personal data. Accordingly, counts-as rules are commonly known as having the property of being *ascriptive* (i.e., introducing new concepts) [28, p. 420].

In this paper we characterise two counts-as rule types: those that ascribe abstract meaning to events and those that ascribe abstract meaning to fluents (properties that hold in states). For these counts-as rules types we give a simple semantics where if we have a rule "A counts-as B in context C" and an A holds/occurs in a context C then a B holds/occurs in the same context C. Counts-as rules semantics is intentionally simple, since we focus on the relation between counts-as rules and norms. Specifically, we will later argue that ascriptive counts-as rules, which introduce abstract institutional facts to refer to concrete institutional or brute facts, are sufficient to interpret norms at different levels of abstraction such that concrete deontic positions (e.g., obligations) *count-as* more abstract deontic positions.

2.1.2 Norms

Institutions, in our framework, use norms to govern a society or to govern other institutions' normative effects. A choice needs to be made on the representation and semantics for norms to take. We will discuss this choice by first describing two common forms for norms in the literature. Namely, an evaluative form [2, 37] and a modal form [70, p. 63]. Then, we will compare evaluative and modal norms in terms of the ease with which we can represent and reason about norms that govern other institutions' normative

effects. Or, in other words, norm governing norms. We will conclude that modal norms offer a simpler way for representing norm governing norms, which in a modal norm representation are a generalisation of norms governing agents.

An evaluative norm provides a qualitative evaluation of an institutional fact in a specific context. For example, "storing communications meta-data is good". More precisely, evaluative norms ascribe institutional facts as being good/bad/a violation/compliant. They take a specialised constitutive form of "A counts-as being good/bad/a violation/compliant in a context C". If regulations take an evaluative form, then they place evaluative statements in the social reality stating how ideal the social reality itself is (e.g., whether there is a violation). Evaluative norms do not place statements in the social reality stating what should be done, only evaluations of what has been done (e.g., stating a norm has been complied with, or the social reality is 'good'). Rather, it is the evaluative rules themselves that state what should and should not be done (e.g., "storing meta-data counts-as compliance" states that meta-data should be stored).

Modal regulatory rules ascribe deontic positions of obligation/permission/prohibition/etc. over particular institutional facts. Modal norms have the form of "An institutional fact A causes the imposition of an obligation/prohibition/permission/etc. to do B in a context C". If norms are modal, then they ascribe 'into' the social reality explicit deontic positions stating what should (not) be done or which state of affairs should (not) be brought about. For example, the social reality can contain an obligation to store communications' metadata. In turn, whether there is compliance or violation is derived from the deontic statements that hold in the social reality. For example, from an obligation to store metadata and the occurrence of storing metadata, compliance is derived. Modal norms place deontic statements in the social reality explicitly stating what should be done, based on which the social reality is evaluated (i.e., whether the deontic positions are complied with).

In this paper we adopt a modal representation for norms. This is because they offer a simpler way to represent and reason about norms at higher levels of governance, which govern norms at lower levels of governance. For example, expressing that it is *required to not require* storing communications metadata if the user has not consented. To see why modal norms are simpler for norm governing norms, we compare evaluative and modal norm representations.

In the evaluative form one possible representation is through rule nesting - "(storing metadata counts-as being good in a context C) counts-as being bad if context C is somehow compatible with the user not consenting". In this form, the instantiation of the nested rule violates the outer rule if the two have compatible contexts. There may be other evaluative representations, but this appears to be the simplest which fully captures the requirement. Determining compliance seems to differ between an evaluative norm about an evaluative norm compared to an evaluative norm governing an agent's actions. On the one hand, determining compliance with an evaluative norm governing an agent's actions involves inspecting the social reality in order to determine whether an agent's actions are compliant. On the other hand, determining compliance with an evaluative norm governing another evaluative norm seems to involve comparing evaluative rules themselves to evaluate the rules' compliance. Hence, evaluative norms governing norms are not a simple generalisation of those governing agents.

In comparison, a possible modal representation is to nest deontic modalities as opposed to rules. An example is the following unconditional modal norm - "it is prohibited to oblige a user's metdata to be stored in the context that they have not consented". Determining compliance for a modal norm about another modal norm seems to be a simple generalisation of determining compliance of a modal norm about an agent's actions. Determining compliance of an agent with a deontic modal statement requires seeing if, in the social reality, the agent is performing actions or bringing about social facts that are obliged/prohibited. Likewise, determining compliance of a deontic modal statement with another deontic modal statement requires seeing if, in the social reality, there is an obligation/prohibition that is itself obliged/prohibited.

We adopt modal norms in this paper as a simple way to reason about norms governing norms. By adopting modal norms the social reality comprises both institutional facts from descriptive constitutive rules and deontic positions from norms stating what is obliged and prohibited.

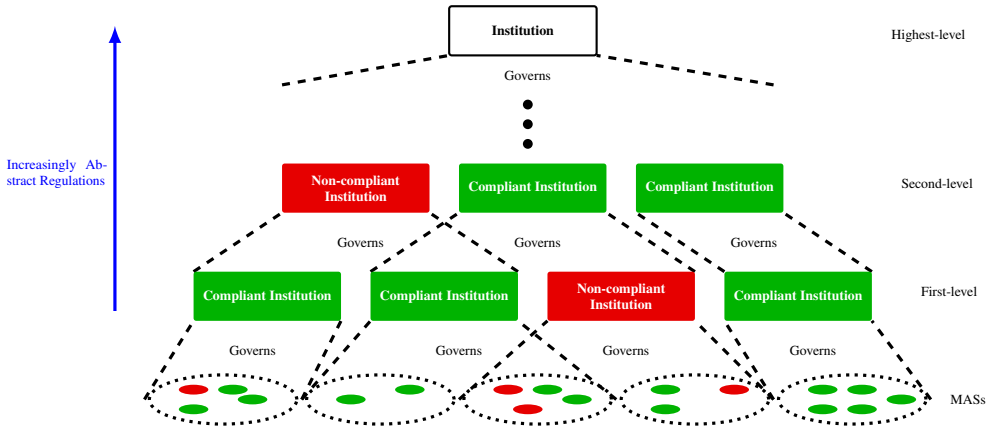

Fig. 1: A high-level depiction of institutions operating in multi-level governance.

2.2 Multi-level Governance

In our view, multi-level governance acts as a mechanism to guide rather than regiment institutional design. The purpose is to coordinate regulations across institutions (e.g., collaborative cross-EU policies for data retention) and ensure institutions do not place unacceptable limits on agents' rights. At the same time, multi-level governance aims to appeal to the principle of subsidiarity (what can be done at the local level, should be left up to the local level). This means that, higher-level institutions do not force lower-level institutions to be designed in a specific way. Rather, higher-level institutions guide the design of lower-level institutions by abstractly defining what obligations and prohibitions lower-level institutions should impose. Where through abstraction, lower-level institution designers are able to comply in multiple ways as deemed appropriate for their jurisdiction. For example, the EU-DRD [22] was designed to coordinate all member states in enacting legislation to store communications' metadata for future criminal investigations. Appealing to subsidiarity, it gave scope for member states to define the length of time metadata is stored for. Another example is the EU-CFR [23], which aims to prevent legislation in the EU from violating agents' rights such as the right to a private life. If legislation is enacted that is non-compliant, fines can be issued, and legislation annulled or abrogated.

We view multi-level governance as comprising three distinctive characteristics relevant to compliance checking, schematically depicted in Figure 1. We draw these characteristics from political science literature [45], work on multi-level governance for artificial societies in AI [67, 68] (in what is called *polycentric* governance), and the real-world case study we focus on. These three characteristics are:

- Regulation of regulation: higher level institutions govern lower level institutions' designs with norms that govern norms. This differs from regimenting legislation changes, which due to institution designers' autonomy might not be possible. Since we adopt regulations as being modal, "A establishes an obligation/prohibition in a context C", regulations governing regulations oblige/prohibit the imposition of obligations/prohibitions. We call these regulations higher-order norms (first-order norms impose obligations/prohibitions on agent actions and/or societal outcomes of agent actions) and they have the form "A establishes an obligation/prohibition for an obligation/prohibition to hold in a context C".
- Multiple connected levels: in multi-level governance, higher-level institutions govern lower-level institutions. We view these institutions as being connected in the sense that the regulations of a lower-level institution can be (non-)compliant with the regulations in a higher-level institution. For example, the EU-DRD is a level 2 institution that requires EU member states' legislation, level 1 institutions, to ensure people's personal communications data is stored. The EU-DRD is governed by

the EU-CFR, a level 3 institution. The directive violates the charter of fundamental right's regulation that demands rights to privacy are respected.

– Abstraction: increasingly abstract regulations, which can be interpreted in many different ways are prescribed at increasingly higher levels of governance. To give an example, at the (typically) highest level of governance, human rights charters use abstract terminology such as 'fairness' or 'privacy' which can have many different interpretations. At a slightly lower level the terminology is more precise, such as in EU directives or supranational agreements between governments, but there are many possible compliant institution designs. For example, the EU-DRD [22] states that member states should legislate for communications' metadata to be stored between 6 and 24 months. This regulation is far clearer than human rights regulations, but does not provide the precise data retention time. At a slightly lower level regulations are more concrete, such as at the level of nation-states. For example, the UK-DRR which implements the EU directive specifies a precise time in that data should be stored. In multi-level governance increasingly abstract regulations, which can be interpreted in many different ways, are prescribed at increasingly higher levels of governance.

A key question is on what basis are concrete regulations determined to be non-compliant with abstract regulations? Legal monitors such as courts interpret the concrete and abstract regulations in order to determine if concrete regulations violate more abstract regulations. To go back to our example, the European Court of Justice [21] determined that the EU-DRD's relatively concrete requirement for metadata to be stored violated the EU Human Rights Charter's for personal data to be processed fairly [23]. The basis of the judgement [21] was an interpretation that storing metadata was the same as storing personal data, and storing personal data without someone's consent was the same as processing data unfairly. In a different context, where someone has given consent, storing metadata would not be unfair data processing.

Hence, a relationship between concrete concepts having a context-sensitive abstract meaning is used to determine compliance between concrete and abstract regulations. According to the concept of institution we use, the context-sensitive rules linking concrete and abstract institutional facts are constitutive rules. Hence, the relation between concrete and abstract *norms* is derived from constitutive rules and based on this relationship concrete norms are determined to be, themselves, (non-) compliant. Specifically, in the most basic case given that if X counts-as Y in a context C then we derive an abstracting relation obliged X counts-as obliged Y in the context C.

There is, however, a well-known argument against this type of derivation. Statements of belief, desire, obligation etcetera. are known as Intentional statements, which are mental states directed at states of affairs (borrowing from Searle [71, p. 3], a capital-I distinguishes the technical term Intention from the specific mental state of intention). Many Intentional statements are also intensional-with-an-s meaning that they fail at a substitution of identicals, to quote Searle [71, p. 23]:

> A sentence such as "John believes that King Arthur slew Sir Lancelot" is usually said to be intensional-with-an-s because it has at least one interpretation where it can be used to make a statement which does not permit existential generalization over the referring expressions following "believes", and does not permit substitutability of expressions with the same reference, *salva veritate*.

In other words, if it is a fact that Sir Lancelot is-a person that never existed, we cannot substitute Sir Lancelot with "a person that never existed" to obtain "John believes that King Arthur slew a person that never existed" *salva veritate*. Hence, the belief Intention is intensional-with-an-s. On the other hand if John believes that King Arthur is a tall person, then it is possible to make a substitution resulting in "John believes that a tall person slew Sir Lancelot". A substitution of X with Y is possible in an intensional-with-an-s statement *if* the substituting property (Y) is held within the same Intention (John believes).

In our case of deriving abstract norms from concrete, a problem stems from the fact that it is a substitution of identicals in Intentional statements (*viz.* obligations) that can also be intensional-with-an-s.

We are substituting obliged X with obliged Y because X counts-as Y (i.e., Y is-an X). To give an example, storing meta-data *counts-as* storing personal data and hence we might argue that there is a derivation to obliging storing meta-data *counts-as* obliging storing personal data. However, if it is not also obliged that storing meta-data *count-as* storing personal data then the substitution fails *salva veritate*. Likewise, King Arthur can only be substituted with "a very tall person" in John's belief, if John believes King Arthur is a very tall person. In order to manage our expectations in this paper, and since this is a difficult topic in its own right that has been covered elsewhere ([71]), we will leave it here and make a simplifying assumption: we assume that if a constitutive rule "X counts-as Y in context C" is included in an institution, through design or interpretation, then the designers/interpreters are implying that it is obliged that "X counts-as Y in context C" and based on that assumption we will also assume a substitution of identicals for abstracting norms is correct *salva veritate.*. To summarise, at the core of our proposal we are abstracting norms based on constitutive rules, which is a substitution of identicals in otherwise intensional-with-an-s statements (norms in our case), and through such abstraction we will determine compliance of institution designs.

3 Approach

In this section we describe the approach we take to automatically determining compliance in multi-level governance. Since we are reasoning about institutions in multi-level governance, we build on an existing institutional reasoning framework. Our proposal requires representation and reasoning for: constitutive rules, modal norms, higher-order norms, connections between institutions and reasoning about regulation abstraction. The InstAL (Institution Action Language) framework [13, 12] provides constitutive rules and modal norms. Hence, we base our proposal on the InstAL framework and extend it to multi-level governance with higher-order and abstract norm representation and reasoning.

We also modify InstAL from capturing institutions that are prohibitive by default (where anything not permitted is forbidden) to permissive institutions (everything is permitted unless explicitly prohibited). The main motivation is simply that the institutions in our running case study, which comprises three institutions in a multi-level governance relationship from real-world law, are inherently permissive. Hence, by representing those institutions in a framework that captures permissive institutions we are able to show a clearer link between our formalised rules and their natural-language counterparts.

Based on InstAL [13, 12], an institution in our framework specifies six elements: events, fluents, constitutive rules that generate institutional events, rules that initiate and terminate fluents, constitutive rules that derive abstract institutional fluents from more concrete institutional fluents and an institution's initial set of inertial fluents that hold in its initial state. Each element is described subsequently in more detail.

Events can represent observable changes to reality, corresponding to the notion of brute fact. Events can represent changes to the social reality, corresponding to the notion of institutional fact. For example, the brute fact we call storing metadata is an observable event, whilst storing metadata and storing personal data are institutional events.

Fluents describe institutional facts holding in a social reality and are subject to changing over time. For example, a user consenting to processing their data causes a fluent to hold stating that they have consented, which is removed if they revoke their consent. Some fluents represent the deontic positions that hold, in our case: obligations, prohibitions and empowerments.

Fluents representing obligations and prohibitions are normative fluents. For example, "an obligation to pay a fine". Higher-order normative fluents can also be specified, for example an obligation to oblige paying a fine. We deal with institutions in a temporal setting, so the various deontic normative fluents express that something should be done before a deadline. For example, an obligation to pay a fine within one month.

Empowerments, in contrast, represent the *institutional power* to perform institutionally-recognised actions as given various formalisations by Jones and Sergot [47], Artikis et al. [7] and Cliffe et al. [13], amongst others. In our use of the concept, a typical example is that of bidding in an auction, multiple

agents may raise their hand which *typically* constitutes bidding, but only those agents empowered to bid can actually do so (e.g., by being registered for the auction in the auction institution). In the context of our case-study, whilst multiple telephony providers may perform an action that constitutes storing communications content, only those providers located in the United Kingdom are empowered to perform that action such that it affects the UK's legal institutions (e.g., by being legal or illegal). To be clear, in line with Jones and Sergot [47], we apply empowerments to agents (in our case study), rather than roles. But in general we make no distinction in our formalism at the meta-level between events occurring in the environment, or institutional actions such as performatives taken by agents or by roles. Hence empowerment is used in a very general sense of making institutional actions *possible* by which we mean legally recognisable.

In contrast, Jones and Sergot [47] formalise institutional power as a non-primitive derived from counts-as rules. Specifically, an agent taking a particular action, such as consenting, constituted by another, such as signing a form, is empowered to take that action (i.e., counts-as rules empower institutional actions to be taken). Whilst we also adopt counts-as rules in their canonical form to ascribe institutional facts, our use of empowerment is as an additional restriction on what actions are empowered to occur - for example, an agent may be able to de facto raise their hand which counts-as bidding, but only if the auctioneer has decided to empower the agents in being able to bid can the agent actually do so. In other words, empowerments represent hard constraints on the actions recognised by an institution, in line with Cliffe et al.'s earlier conceptualisation [13].

Event generating constitutive rules cause institutional events to occur when observable (brute) events or institutional events occur in a given context. For example, "the observable event of storing metadata counts-as the institutional event of storing metadata". An example of a rule where an institutional events causes further institutional events to occur is "storing personal data counts-as unfair data processing in the context that a user has not consented".

Fluent initiation and termination rules cause inertial fluents to hold in a state when initiated and persist from one state to another over time until terminated. For example, "a user consenting to storing their data initiates the fluent stating that the user has consented". Rules that establish what we call normative fluents are norms. For example, "a user using a communications device initiates an obligation for their communications' metadata to be stored". Higher-order norms impose higher-order normative fluents. Once a fluent is initiated by such a rule it holds until it is terminated by another rule. That is, these rules initiate and terminate inertial fluents.

Constitutive rules that derive fluents based on other fluents holding extend a state comprising relatively concrete institutional facts to a state comprising more abstract institutional facts. For example, "an obligation to store personal data non-consensually counts-as unfair data processing, unconditional on any specific social context". Generally, these rules have the form "fluent A counts-as fluent B in context C". Viewed as counts-as rules, these rules ascribe a special meaning B to a fluent A in a context C. For example, an obligation to store personal data non-consensually has the special meaning of being unfair data processing. So long as the fluent 'A' holds in a context 'C' then its special meaning 'B' also holds. But, unlike fluent initition and termination rules, the special meaning 'B' does not hold until terminated, rather, it holds when 'A' holds in the context 'C'. That is the 'Bs' in rules of this type are *non-inertial* fluents, since the Bs do not persist over time by default until terminated (i.e., they do not possess inertia). Unlike the previous rules, constitutive rules that derive non-inertial fluents from other fluents are not present in the InstAL framework. Similar non-inertial fluent rules with the form "in context C non-inertial fluent A also holds" are present in subsequent InstAL developments [54, 65, 66].

Each fluent in an institution's set of initial inertial fluents, which can be the empty set, holds in the institution's first state and continues to hold until terminated. To summarise, an institution specifies events, fluents and constitutive rules which ascribe institutional events or institutional fluents.

Multi-level governance is operationalised with a semantics. This semantics defines how each institution evolves from one state to the next in response to a trace of observable events. These events can be real events occurring in the MAS, or hypothetical events if a pre-runtime check for compliance is performed. An institution's evolution is schematically depicted in Figure 2 and described as follows.

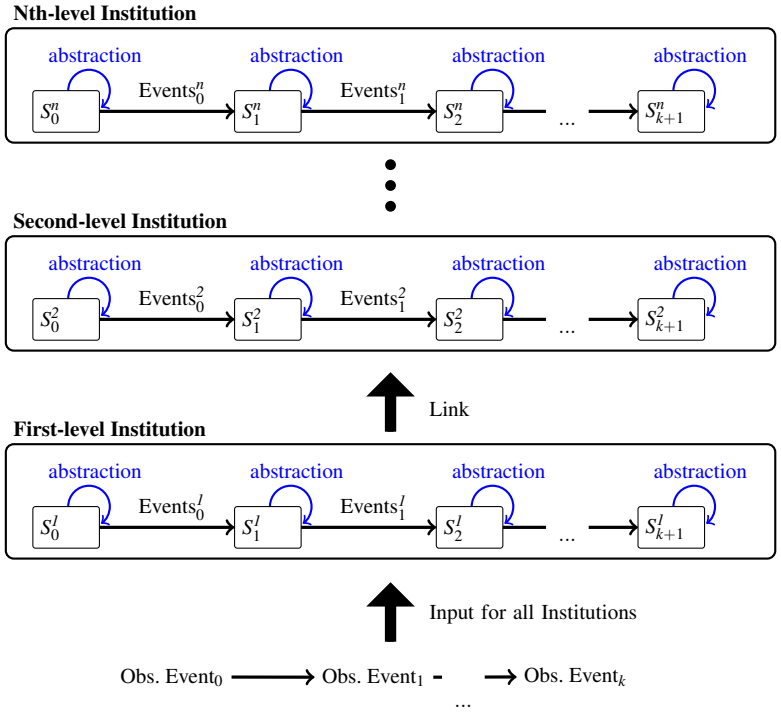

Fig. 2: Overview of Multi-level Governance Reasoning

The institution starts in an initial state in which its initial set of inertial fluents holds. State transitions are driven by observable events occurring in the MAS (potentially hypothetically). During a state transition, further events occur in an institution according to its constitutive rules, building up an institutional interpretation of reality based on the observable events that have occurred. Further events signifying there is (non-)compliance also occur, for example if there is an obligation to store communications' metadata within one month and the data is not stored within one month, then a norm violation occurs. If it is prohibited to oblige storing communications' metadata, then a higher-order norm violation occurs. That is, norm violations are *institutional events* denoting non-compliance. A newly transitioned to state can contain different fluents from the previous state, based on each institution's constitutive rules variously initiating and terminating fluents from one state to the next. Thus, each institution evolves over time from one state to the next transitioned by events.

Recall that concrete lower level institution norms are abstracted, in order to determine whether they are compliant, in higher level institutions according to constitutive rules. The approach we take is to firstly, link each institutional level such that concrete normative fluents holding in lower level institutions are 'passed up' to the corresponding state in higher level institutions. For example, an obligation to oblige storing communications metadata in the EU-DRD is 'passed up' to the EU-CFR for monitoring. Likewise, so too are norm compliance events.

Then, in each institutional state of a higher level institution the concrete normative fluents coming from lower level institutions are re-interpreted and *abstracted* based on constitutive rules. To give an example, storing communications metadata counts-as non-consensual data processing in the context that the person whom the data concerns has not consented. Since storing metadata in such a context is ascribed the special status of non-consensual data processing, an obligation to oblige storing communications metadata is re-interpreted as an obligation to oblige non-consensual data processing.

In turn, from these abstractions any further abstractions are also derived. For example, the obligation to oblige non-consensual data processing is abstracted simply to being unfair data processing, if such an

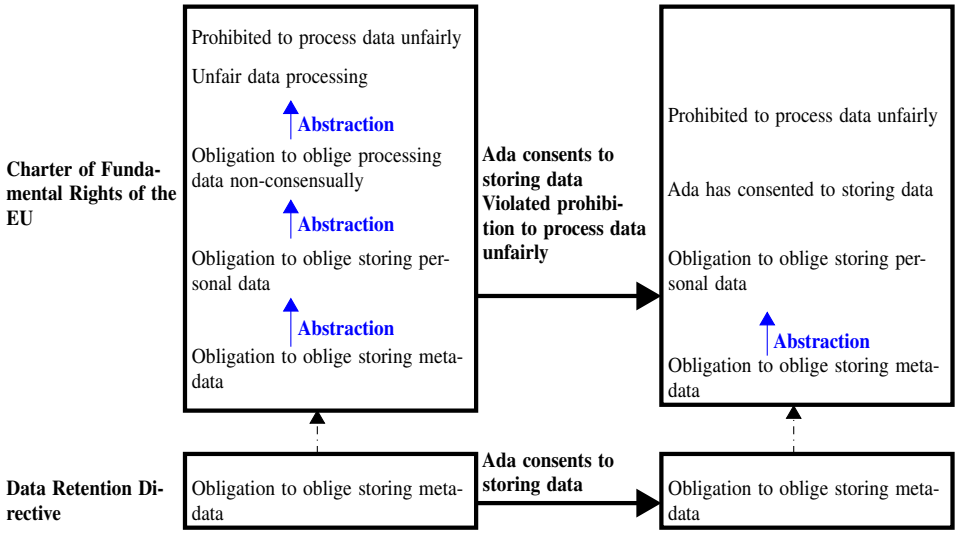

Fig. 3: An example of abstracting normative fluents at different levels of governance based on the context. Normative fluents oblige/prohibit an aim a occurs before or at the same time as a deadline d. We use $<$ to denote one thing occurring strictly before another and $\leq$ to denote one thing occurring before or at the same time as another.

ascription exists according to constitutive rules. Thus, each institutional state contains concrete normative fluents from lower levels and the state contains the closure of all abstractions on these concrete normative fluents based on constitutive rules. So, it is the concrete normative fluents *imposed by norms* in lower level institutions that are re-interpreted as more abstract normative fluents at higher levels. Hence, concrete normative fluents are determined in their abstract incarnation whether they cause non-compliance and thus whether their originating concrete norms are compliant with abstract norms.

An example is depicted in Figure 3 based on the running case study and described as follows:

1. In the EU-DRD's first state there is an obligation to oblige storing communications' metadata, which is passed up to the EU-CFR.
2. In the EU-CFR's initial state the EU-DRD's obligation to oblige storing communications' metadata is abstracted. This is because concrete normative fluents are abstracted based on whether the prescribed event *counts-as* a more abstract event in a context entailed by the state. Specifically:
   i The obligation to oblige storing metadata is abstracted to an obligation to oblige storing personal data, because storing metadata counts-as storing personal data.
   ii The obligation to oblige storing personal data is abstracted to an obligation oblige processing data without consent, because storing personal data counts-as non-consensual data processing in the context where an agent has not consented.
3. An obligation to oblige processing data non-consensually counts-as 'unfair data processing' and is hence abstracted to 'unfair data processing'.
4. Unfair data processing is prohibited and thus a norm violation event occurs in the transition to the EU-CFR' next state.

In the EU-CFR institution the next state lacks an obligation to oblige processing data without consent because a user has consented. So, unfair data processing also does not hold. That is, the abstract meaning of concrete normative fluents evolves as the context evolves. Consequently, compliance of normative fluents is context sensitive because normative fluents' abstraction is context sensitive.

To summarise, our semantics for multi-level governance defines the evolution of each institution over time in response to a sequence of events. Specifically, the semantics takes concrete normative fluents from lower-level institutions and abstracts them in higher-level institutions. Abstracted normative fluents can cause higher-order norm compliance events through discharging or violating higher-order norms. Thus, non-compliance can be determined by inspecting the sequence of events in higher level institutions for higher-order norm compliance events. In the next section we will define the representation and a semantics as described.

4 Formal Framework

In this section we present the syntax for representing multi-level governance (subsection 4.1) and alongside give the intuition/informal semantics for each syntactic construct. Then, we provide a semantics for reasoning about multi-level governance (subsection 4.2).

4.1 Syntax

We begin with representing normative fluents. These are fluents that represent temporal obligations and prohibitions, meaning they have an aim which should be achieved before a deadline. Obligation and prohibition fluents are respectively represented as *obl(aim, deadline)* and *pro(aim, deadline)*. The aims and deadlines can be events, fluents or other normative fluents to represent higher-order normative fluents. Two special events are used in aims and deadlines, *now* and *never*[1]. The event 'now' occurs immediately to represent something should (not) be done immediately. For example, *obl(aim, now)* means the aim should occur 'now'. Our representation is inspired by the formalisation of instantaneous norms in a variant of dynamic logic [17], which similarly use 'now' to refer to the present state. An aim or deadline event *never* represents that the aim/deadline never occurs, potentially meaning that the normative fluent lasts indefinitely. For example *pro(aim, never)* means it is always prohibited for the aim to occur or in other words that the aim should 'never' occur.

The informal semantics for normative fluents' is described in terms of when obligations/prohibitions are discharged and violated, overviewed in Figure 4. An obligation fluent, of the form *obl(aim, deadline)*, represents that the aim should occur/hold before or at the same time as the deadline to *discharge* the obligation (fulfil all duties). If the deadline occurs/holds strictly before the aim then the obligation is *violated*. Prohibition fluents, of the form *pro(aim, deadline)*, are the inverse of obligations. Prohibitions represent that the aim should not occur/hold strictly before the deadline. When a normative fluent n is discharged it causes the event $disch(n)$ to occur. If n is violated then the event $viol(n)$ occurs.

Higher-order norms impose higher-order normative fluents. A higher-order normative fluent obliges/prohibits another normative fluent is imposed before a deadline. The deadline is an event or another normative fluent. Compliance-focussed higher-order norms can also be expressed, which oblige/prohibit compliance with an obligation/prohibition occurs before/after an event occurs or another normative fluent is imposed (e.g., it is obliged a norm is violated before a fine is imposed). Where for an obligation $n = obl(a, d)$ or prohibition $n = pro(a,d)$ norm discharge is the event $disch(n)$ and violation is the event $viol(n)$. A grammar to specify normative fluents is formalised as:

[1] We allow a normative fluent's aim to be now or never, even though, for example, it sounds odd to say "it is obliged to be now". This is for symmetry between obligations and prohibitions - for example, obliged never before an event E is the same as saying prohibited E until never (i.e., forever).

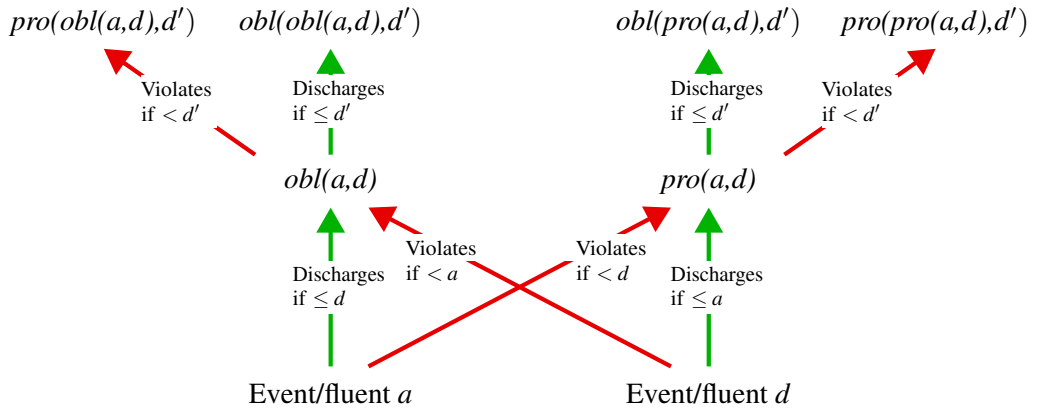

Fig. 4: Discharge and violation (higher-order) normative fluent conditions. $< X$ denotes the event/fluent holding or occurring strictly before X causes a violation. $\leq X$ denotes the same, but the condition is not strictly before.

Definition 1 Normative Fluents Let P be a set of propositions denoting events with typical element p. The set of normative fluents $\mathcal{N}|_P$ is the set of all normative fluents n expressed as:

$$\begin{array}{lll} aim & ::= & p \mid n \mid now \mid never \mid disch(n) \mid viol(n) \\ deadline & ::= & p \mid n \mid now \mid never \mid disch(n) \mid viol(n) \\ n & ::= & obl(aim, deadline) \mid pro(aim, deadline) \end{array}$$

We give some examples concerning two agents, a law enforcement officer called Charles and an internet communications user called Ada, and an internet communications provider colloquially called an ISP. The UK-DRR implements the EU-DRD. The UK-DRR states that if a law enforcement official (e.g., *charles*) requests the data stored by a communications provider (e.g., *isp*) of a user (e.g., *ada*) then the communications provider is obliged to provide the data within one month (*m1*):

$$obl(provideData(isp, charles, ada), time(m1))$$

Instantaneous normative fluents express that something should (not) be done or a normative fluent should (not) be imposed *now*. One way an higher-level institution designer might use instantaneous norms is to express that as soon as something happens a normative fluent should be imposed. For example, as soon as a norm is violated it is obliged that there is an obligation to punish the violator. The EU-DRD as we formalise it, requires that any implementing legislation should impose punishment as soon as regulations are violated. Thus, when there is a violation it imposes a normative fluent obliging an obligation to punish the violator is imposed immediately:

$$obl(obl(punish(isp), time(m6)), now)$$

Compliance-focussed normative fluents can be used to express that an agent should discharge/violate a normative fluent before another normative is imposed that rewards/punishes the agent. For example, in our previous work [52], an obligation expressed "it is obliged that a norm is violated before a fine is imposed". Such compliance focussed normative fluents do not state that a normative fluent being discharged should *cause* a reward/punishment. Rather, they state that discharge/violation should occur before the reward/punishment is imposed. Following this paper's case-study - it is obliged that the communications provider *isp* violates the obligation to provide *charles* with data which concerns *ada* before any obligation to punish the communications provider *isp* is imposed.

obl(viol(obl(provideData(isp, charles, ada), time(m1))), obl(punish(isp), time(m6)))

Normative fluents can also be explicitly first-order, but implicitly higher-order by obliging/prohibiting fluents that abstractly represent other normative fluents. Recall that various obligations in the EU-DRD can abstractly be interpreted as unfair data processing. Hence, the following is an example of an abstract first-order norm that indirectly governs other norms. The EU-CFR states that it is prohibited to process Ada's data unfairly (indefinitely):

pro(unfairDataProcessing(ada), never)

We now proceed to representing individual institutions. In short, institutions are specified as a tuple, extending the formal specification of an institution in the InstAL framework [13]. Generally, speaking, an individual institution describes the things that can occur (events) and hold (fluents) in the institution as well as the institution's rules causing events to occur and fluents to hold. An institution's constitutive rules - cause institutional events to occur in response to other events ("an event A counts-as an event B in context C"), fluents to hold in response to events ("an event A counts-as establishing/removing a fluent B in context C"), and further, more abstract, fluents to be derived from other fluents ("a fluent A counts-as a fluent B in context C"). Rules stating fluents are derived are not present in InstAL but we introduce them since they provide an abstracting relation *between* fluents and thus contribute to our goal of reasoning about abstraction in multi-level governance. Regulative rules are just modal norms represented as constitutive rules that establish normative fluents, "an event A counts-as establishing an obligation/prohibition in context C".

Specifically, institutions comprise the following elements:

Events - a set of propositions ($\mathcal{E}$) denoting events that can occur in the institution, s.t. $now, never \notin \mathcal{E}$, meaning that the institution cannot define when the events *now* and *never* occur. The set of events comprises:

- Observable events ($\mathcal{E}_{obs}$) that are exogenous to the institution corresponding to the notion of a *brute fact* denoting an event.
- Internal institutional events ($\mathcal{E}_{inst}$) representing an institutional description of reality.
- Compliance events ($\mathcal{E}_{norm} = \{disch(n), viol(n) \mid n \in \mathcal{F}_{cnorm} \cup \mathcal{F}_{anorm}\}$) indicating a normative fluent (in the set of concrete and abstract normative fluents $\mathcal{F}_{cnorm} \cup \mathcal{F}_{anorm}$) has been discharged or violated.

Fluents - a set of propositions ($\mathcal{F}$) denoting the fluents that can hold in the institution, comprising:

- *Domain fluents* ($\mathcal{F}_{dom}$) providing an institutional description of the state of reality (e.g., an agent has consented to their data being processed).
- *Empowerment fluents* ($\mathcal{F}_{pow} \subseteq \{pow(e) \mid e \in \mathcal{E}_{inst}\}$) denoting an event is *recognised* by the institution in a state and has the power to affect the institution (i.e., is empowered).
- *Normative fluents* ($\mathcal{F}_{norm} = \mathcal{F}_{cnorm} \cup \mathcal{F}_{anorm}$) comprising mutually disjoint sets of *concrete normative fluents* ($\mathcal{F}_{cnorm} \subseteq \mathcal{N}|_{\mathcal{E} \cup \mathcal{F}_{dom}}$) and *abstract normative fluents* ($\mathcal{F}_{anorm} \subseteq \mathcal{N}|_{\mathcal{E} \cup \mathcal{F}_{dom}}$):
  - *Concrete normative fluents* denote obligations and prohibitions imposed by the institution about events or domain fluents. These normative fluents are concrete in the sense of being explicitly imposed by an institutional norm, rather than being abstract interpretations of other normative fluents that have been imposed.
  - *Abstract normative fluents* denote obligations and prohibitions imposed by the institution about events or domain fluents. These are abstract in the sense of not being imposed by the institution, but rather represent an abstract interpretation of other *more* concrete normative fluents. For example, an obligation to store personal data is a more abstract interpretation of an obligation to store communications metadata.
- *Inertial and non-inertial fluents*, We assume that fluents are either inertial or non-inertial represented as mutually disjoint sets of *inertial* fluents ($\mathcal{F}_{inert}$) and *non-inertial* fluents ($\mathcal{F}_{ninert}$) such that $\mathcal{F} = \mathcal{F}_{inert} \cup \mathcal{F}_{ninert}$ and $\mathcal{F}_{inert} \cap \mathcal{F}_{ninert} = \emptyset$. Institutions define fluents that can be initiated by

the institution's state consequence function and then persist from one state to the next by default until they are terminated. That is, some fluents are *inertial*. Other fluents hold due to constitutive rules stating more abstract fluents are derived from more concrete fluents. These abstract fluents hold whenever the concrete fluents hold and do not persist from state to state by default. That is, they are *non-inertial* fluents. Concrete normative fluents are inertial, since an institution explicitly imposes them by initiation and termination according to the state consequence function ($\mathcal{F}_{cnorm} \subseteq \mathcal{F}_{inert}$). Abstract normative fluents are non-inertial since they are derived from other normative fluents and do not persist from state to state by default ($\mathcal{F}_{anorm} \subseteq \mathcal{F}_{ninert}$).

Contexts - these characterise a condition on a state and denote the social context each rule is conditional on. A context is a set of positive and weakly negative fluents, which acts as a condition on a state that is true if all of the positive fluents hold and none of the negative fluents hold. Formally, the set of all contexts is $\mathcal{X} = 2^{\mathcal{F} \cup \neg \mathcal{F}}$ s.t. $\neg \mathcal{F} = \{\neg f \mid f \in \mathcal{F}\}$ is the set containing the negation of all elements in the set $\mathcal{F}$.

State change rules ($\mathcal{C} : \mathcal{X} \times \mathcal{E} \to 2^{\mathcal{F}_{inert}} \times 2^{\mathcal{F}_{inert}}$), described as a state consequence function. They specify that due to the occurrence of events conditional on a context holding in a state, inertial fluents are initiated and terminated from one state to the next. State change rules can be descriptive (e.g., a user consenting to their data being stored initiates a fluent stating that they have consented) and regulative rules by initiating and terminating normative fluents (e.g., someone using electronic communications initiates an obligation for the communications provider to store their communications' metadata).

Event generation rules - ($\mathcal{G} : \mathcal{X} \times \mathcal{E} \to 2^{\mathcal{E}_{inst}}$), described as an event generation function. These rules are only descriptive. They specify that when an exogenous or institutional event occurs, conditional on a social context holding in a state, another institutional event occurs.

Fluent derivation rules - ($\mathcal{D} : \mathcal{X} \times \mathcal{F} \to 2^{\mathcal{F}_{ninert}}$), described as a fluent derivation function. These rules state that a fluent holding in a state derives a *non-inertial* fluent holding in the same state, conditional on a social context.

According to these notions, an individual institution is formally defined as:

Definition 2 Individual Institution An institution is a tuple $\mathcal{I} = \langle \mathcal{E}, \mathcal{F}, \mathcal{C}, \mathcal{G}, \mathcal{D}, \Delta \rangle$ such that:

- $\mathcal{E} = \mathcal{E}_{obs} \cup \mathcal{E}_{inst} \cup \mathcal{E}_{norm}$ is the set of events.
- $\mathcal{F} = \mathcal{F}_{dom} \cup \mathcal{F}_{norm} \cup \mathcal{F}_{pow}$ is the set of fluents.
- $\mathcal{C} : \mathcal{X} \times \mathcal{E} \to 2^{\mathcal{F}_{inert}} \times 2^{\mathcal{F}_{inert}}$ is the state consequence function.
- $\mathcal{G} : \mathcal{X} \times \mathcal{E} \to 2^{\mathcal{E}_{inst}}$ is the event generation function.
- $\mathcal{D} : \mathcal{X} \times \mathcal{F} \to 2^{\mathcal{F}_{ninert}}$ is the fluent derivation function.
- $\Delta \subseteq \mathcal{F}_{inert}$ is the set of inertial fluents that *initially* hold in the institution's zeroeth state (and until terminated will hold in subsequent states).

Some further useful constructs are:

- $\Sigma = 2^{\mathcal{F}}$ to denote the set of all states for $\mathcal{I}$.
- Given a context $X \in \mathcal{X}$ and an event $e \in \mathcal{E}$ we denote the result of the consequence function as $\mathcal{C}(X,e) = \langle \mathcal{C}^{\uparrow}(X,e), \mathcal{C}^{\downarrow}(X,e) \rangle$ s.t. the set of initiated fluents is $\mathcal{C}^{\uparrow}(X,e)$ and the set of terminated fluents is $\mathcal{C}^{\downarrow}(X,e)$.
- For readability if an institution is denoted with a superscript, such as *id* then all its elements have the same superscript, such as $\mathcal{I}^{id} = \langle \mathcal{E}^{id}, \mathcal{F}^{id}, \mathcal{C}^{id}, \mathcal{G}^{id}, \mathcal{D}^{id}, \Delta^{id} \rangle$, the set of states being Σ^{id} and the set of contexts being $\mathcal{X}^{id}$.

We exemplify using institutional specification fragments where for clarity we use a superscript denoting the name of each institution. The EU-CFR [23, Art. 8.2] states that a person's data must be processed fairly. For an agent called 'ada', the set of inertial fluents initially holding in the CFR institution includes:

$$pro(unfairDataProcessing(ada), never) \in \Delta^{cfr}$$

A communications provider, called 'isp', storing metadata is by default an event *empowered* to affect the Data Retention Regulations:

$$pow(storeData(isp,\ ada,\ metadata)) \in \Delta^{drd}$$

Now we give some example counts-as rules, fluent initiation and termination rules and norms (where, for clarity, we use $\ni$ to denote right-hand side's membership of the left-hand-side). According to the European Court of Justice's (ECJ) judgement [21] on the EU-DRD, with respect to the EU-CFR, storing communications metadata counts-as storing personal data. If an agent's metadata is stored, such as Ada's, unconditional on a specific context (the empty set) then the event of storing the Ada's personal data is generated. Additionally, if Ada's personal data is stored in the context that Ada has not consented then the event of non-consensually processing Ada's data occurs. The following rules are a part of the EU-CFR, incorporating the ECJ's judgement.

$$\mathcal{G}^{cfr}(\emptyset, storeData(isp,\ ada,\ metadata)) \ni storeData(isp,\ ada,\ personal)$$

$$\mathcal{G}^{cfr}(\{\neg consentedDataProcessing(ada,\ isp)\}, storeData(isp,\ ada,\ personal)) \ni nonConsensualDataProcessing(ada)$$

The DRD [22, Art. 8] requires data concerning people is transmitted to authorities on request before any undue delay. A fluent initiation rule is conditional on an agent, Charles, requesting the data from a communications provider, ISP, of another agent, Ada. In the context that Charles is a law enforcement official the rule initiates an obligation to immediately oblige that ISP provides Charles with data concerning Ada before any undue delay.

$$\mathcal{C}^{drd\uparrow}(\{is(charles,\ lawEnforcement)\}, requestData(ada,\ isp,\ charles)) \ni obl(obl(provideData(isp,\ charles,\ ada),\ undue\_delay),\ now)$$

According to the ECJ's interpretation of the EU-DRD [21] with respect to the EU-CFR. Obliging that personal data is processed non-consensually counts-as unfair data processing. We represent the ECJ's interpretation as a fluent derivation rule in the CFR institution. It states that obliging an agent, Ada's, personal data is processed without consent in any social context (the empty set) derives the fluent of (counts-as) unfair data processing.

$$\mathcal{D}^{cfr}(obl(nonConsensualDataProcessing(ada),\ now), \emptyset) \ni unfairDataProcessing(ada)$$

In multi-level governance, institutions are related in the sense that institutions designed at lower levels of governance are governed by institutions designed at higher levels of governance. In our approach, this means that if a lower level institution imposes an obligation or a prohibition, then the same obligation/prohibition holds in any higher level institution that governs it. Institutions are linked in this way in what we call a *multi-level governance institution*, where institutions are ordered by the level they operate at and linked with a relation between lower level and higher level institutions. The relations linking institutions are expressed as a set of directed edges R between lower level institutions and higher level institutions they are governed by. Each relation is restricted such that higher levels cannot be governed by lower levels, meaning that collectively the relations are always acyclic. Formally, a multi-level governance institution is:

Definition 3 Multi-level Governance Institution A Multi-level Governance Institution is a directed graph $\langle \mathcal{T}, R \rangle$. The vertices are represented as a tuple $\mathcal{T} = \langle \mathcal{I}^1, ..., \mathcal{I}^n \rangle$ of institutions. The arrows are a set of pairs $R = 2^{[1,n]\times[1,n]}$ of institution indexes in $\mathcal{T}$ such that $\forall \langle i, j \rangle \in R : i < j$.

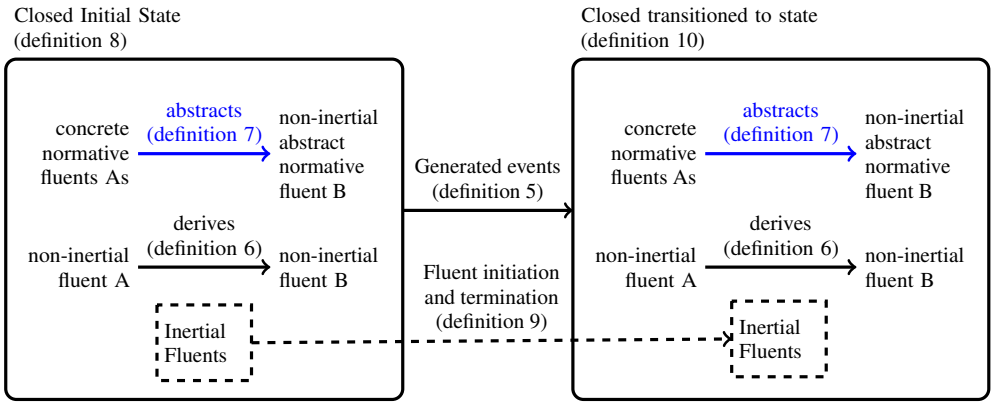

Fig. 5: An overview of the semantics, depicting the transition from the initial state to the next state and state closure.

For example, in our running case study a multi-level governance institution is $\langle \mathcal{T}, R \rangle$ with a hierarchy of institutions comprising the UK-DRR, the EU-DRD and the EU-CFR such that $\mathcal{T} = \langle \mathcal{I}^{drr}, \mathcal{I}^{drd}, \mathcal{I}^{cfr} \rangle$ and $R = \{\langle drr, drd \rangle, \langle drd, cfr \rangle\}$. To demonstrate the representation in its full entirety, the case study is formalised in Appendix A.

According to these definitions, we can represent the three main aspects of multi-level governance we focus on in this paper. Firstly, regulations that regulate other regulations in higher level institutions with higher-order normative fluents in prescriptive rules. Secondly, the links between lower level institutions governed by higher level institutions by composing multi-level governance institutions. Thirdly, the interpretation of concrete concepts and normative fluents as more abstract concepts and normative fluents.

In our framework abstraction can occur in institutions related via multi-level governance in two ways. Firstly with constitutive rules, which state concrete concept counts-as a more abstract concept in a particular context. Such abstracting constitutive rules are represented as the event generation function and the fluent derivation function. The event generation function represents abstracting constitutive rules of the form "a concrete event A counts-as a more abstract event B in the context C". The fluent derivation function represents abstracting constitutive rules of the form "a concrete fluent A counts-as an abstract fluent B in the context C". The second abstraction method is the implicit abstraction of concrete normative fluents regulating concrete concepts to more abstract normative fluents regulating abstract concepts. Normative fluent abstraction requires no explicit representation, since it is defined semantically based on constitutive rules between concrete and abstract concepts according to the event generation and fluent derivation functions.

4.2 Semantics

In this section we present the formal semantics for multi-level governance. Given a multi-level governance institution specification the semantics define a *model*, comprising for each institution states transitioned between by events, in response to a supplied trace of observable events. The key idea behind the semantics, depicted in Figure 5 is to transition from one state to another, driven by generated events, by initiating and terminating *inertial* fluents. Then each state is closed by deriving *non-inertial* fluents according to an institution's fluent derivation function and abstracting concrete fluents to non-inertial abstract normative fluents according to normative fluent abstraction. Given a multi-level governance institution model it can be determined whether individual institutions are compliant with the institutions that govern them in

different contexts. The formal semantics provide a mechanism for automated compliance-checking in multi-level governance.

In order to reduce repetition the following definitions are with respect to several common objects. First, a multi-level governance institution $\mathcal{ML} = \langle \mathcal{T}, R \rangle$ where $\mathcal{T} = \langle \mathcal{I}^1, ..., \mathcal{I}^n \rangle$ is a tuple of institutions with typical elements being $\forall i \in [1,n] : \mathcal{I}^i = \langle \mathcal{E}^i, \mathcal{F}^i, \mathcal{C}^i, \mathcal{G}^i, \mathcal{D}^i, \Delta^i \rangle$. Second, a tuple of states, representing the state of each institution for a single point in time j - $\langle S^1_j, ..., S^n_j \rangle$. Third, a tuple of event sets, representing the events occurring in each institution for a single point in time j - $\langle E^1_j, ..., E^n_j \rangle$.

4.2.1 State Conditions

Institutional rules are conditional on states and the occurrence of events. Therefore, determining if a rule is 'fired' requires determining in part if its state condition, a social context, holds in a state. We begin by defining when contexts are *modelled* by (hold in) a state.

Informally, a state formula is modelled by a state if for each positive fluent in the formula there is an equivalent fluent that is a member of the state and for each negative fluent in the formula there is not an equivalent fluent that is a member of the state. Rather than defining modelling a state formula in terms of whether the positive/negative fluent is *in* the state, we use equivalence. This is because two normative fluents can have an equivalent semantics whilst being syntactically different - this is not unusual, in 'Standard Deontic Logic' [76] forbidden X is defined as obliged not X.

In our case, we define equivalences between two fluents based on whether they are are syntactically identical and two normative fluents based on whether their (non-) compliance coincide. Referring again to Figure 4, there is an obligation for an event/fluent a to occur/hold before or at the same time as a deadline d when the obligation fluent *obl(a, d)* holds, likewise a is prohibited strictly before d when the prohibition fluent $pro(a,d)$ holds. Given two normative fluents *obl(a, d)* and $pro(a',d')$ where a is equivalent to d' and d equivalent to a' the obligation's and prohibition's discharge and violation coincide and therefore they are equivalent. The equivalences ($\equiv$) of obligations and prohibitions according to their discharge and violation is summarised as $obl(a,d) \equiv pro(a',d')$ if $a \equiv d'$ and $d \equiv a'$, a definition that generalises to higher-order normative fluents[2].

Accordingly, we define modelling a state formula as:

Definition 4 State Formulae Let $f \in \mathcal{F}^i$ be a fluent. We define $\equiv$ and $\models$ for all contexts $X \in \mathcal{X}^i$ as:

$$\begin{aligned} f &\equiv f \\ obl(a,d) \equiv pro(a',d') &\Leftrightarrow a \equiv d' \textbf{ and } d \equiv a' \\ S^i \models f &\Leftrightarrow \exists f' \in S^i : f \equiv f' \\ S^i \models \neg f &\Leftrightarrow \nexists f' \in S^i : f \equiv f' \\ S^i \models X &\Leftrightarrow \forall x \in X : S^i \models x \end{aligned}$$

4.2.2 Events

In this section we semantically define the events occurring in an institution, in response to other events in specific contexts. Precisely, an event generation operation GR^i defines for an institution $\mathcal{I}^i$ in a multi-level governance institution which events occur in a state S^i in response to a set of events E^i. An event occurs in an institution if it is generated by the institution's event generation function $\mathcal{G}^i$, or if it represents the discharge/violation of a discharged/violated normative fluent holding in the institution's state or that of a lower-level institution the institution governs. The event generation operation is formalised below and explained subsequently:

Definition 5 Event Generation Operation The event generation operation $GR^i : \Sigma^i \times 2^{\mathcal{E}^i} \rightarrow 2^{\mathcal{E}^i}$ is defined for each institution $\mathcal{I}^i$ w.r.t. the tuple of multi-level governance states $\langle S^1_j, ..., S^n_j \rangle$ and event sets

[2] An example of higher-order equivalence generalisation is $obl(obl(a,d),d') \equiv obl(pro(d,a),d') \equiv pro(d',obl(a,d))$, etc.

$\langle E_j^1, ..., E_j^n \rangle$. The operation is defined as $GR^i(S^i, E^i) = E'$ iff E' *minimally* (w.r.t. set inclusion) satisfies all of the following conditions:

$$now \in E' \tag{D5.1}$$
$$E^i \subseteq E' \tag{D5.2}$$
$$\exists X \in \mathcal{X}^i, e \in E', e' \in \mathcal{G}^i(X, e) : S^i \models X \wedge S^i \models pow(e') \Rightarrow e' \in E' \tag{D5.3}$$
$$S^i \models obl(a,d) \wedge (a \in E' \vee S^i \models a) \Rightarrow disch(obl(a,d)) \in E' \tag{D5.4}$$
$$S^i \models obl(a,d) \wedge (d \in E' \vee S^i \models d) \wedge disch(obl(a,d)) \notin E' \Rightarrow viol(obl(a,\ d)) \in E' \tag{D5.5}$$
$$S^i \models pro(a,d) \wedge (d \in E' \vee S^i \models d) \Rightarrow disch(pro(a,d)) \in E' \tag{D5.6}$$
$$S^i \models pro(a,d) \wedge (a \in E' \vee S^i \models a) \wedge disch(pro(a,d)) \notin E' \Rightarrow viol(pro(a,d)) \in E' \tag{D5.7}$$
$$\exists \langle h, i \rangle \in R, e \in \mathcal{E}^h_{norm} \cap \mathcal{E}^i_{norm} \Rightarrow e \in E' \tag{D5.8}$$

In more detail:

- D5.1 - the event of *now* always occurs.
- D5.2 - events that have already occurred still occur (monotonicity).
- D5.3 - an event generated by the institution's event generation function in response to another event, conditional on a social context modelled by the state and the event being empowered to occur.
- D5.4 to D5.7 - a compliance event occurring signifying a normative fluent is discharged or violated in a state, by an obliged/prohibited event, fluent or another normative fluent. Compliance events do not need to be empowered in order to occur.
- D5.8 - norm compliance events occurring in lower level institutions linked to this institution, also occur in this institution.

Collectively, these conditions and the minimality constraint *close* a set of events by producing all events in response to those events (etc.). Note that GR^i is increasingly monotonic, well-defined and can be a partial function. The function GR^i is partial if there is a fault in the institutional specification or the set of events passed are inconsistent. Specifically, if an institution is defined such that violating a normative fluent causes an event that discharges the same normative fluent via the event generation function $\mathcal{G}$ (either directly or transitively). We will see later in subsection 4.2.5 how events generated cause fluents to be initiated and terminated when all the definitions are put together to define a multi-level governance institution model in definition 13.

4.2.3 Derived Fluents

In this section we semantically define deriving fluents from other fluents in a given state. We define a fluent derivation operation FD^i which, operating on an institutional state, extends the state to include derived fluents based on fluent derivation rules of the form "fluent A derives fluent B in context C" described by the fluent derivation function $\mathcal{D}^i$. These derived fluents are the 'Bs' from fluent derivation rules where the context 'C' holds and the fluent 'A' is modelled by the state. By deriving fluents from other fluents in a state, it is possible further fluents should be derived. Thus, the fluent derivation operation FD^i is defined to close a state by producing an extended state that includes all derived fluents *with respect to* the extended state itself. The fluent derivation operation is formally defined as:

Definition 6 Fluent Derivation Operation The fluent derivation operation $FD^i : \Sigma^i \to \Sigma^i$ is defined for each institution $\mathcal{I}^i$ and a state $S^i \in \Sigma^i$ such that $FD^i(S^i) = S'$ iff S' *minimally* (w.r.t. set inclusion) satisfies all of the following conditions:

$$S^i \subseteq S' \tag{D6.1}$$
$$\exists X \in \mathcal{X}, f \in S', f' \in \mathcal{D}^i(X, f) : S' \models X \Rightarrow f' \in S' \tag{D6.2}$$

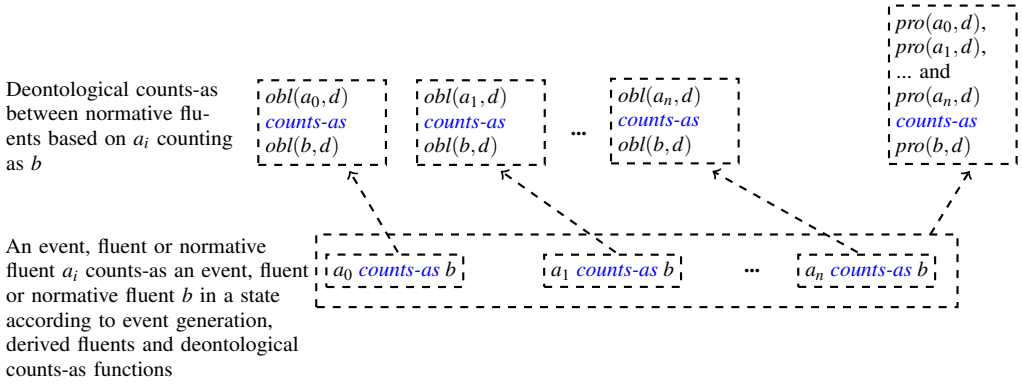

Fig. 6: Overview for deontological counts-as semantics between concrete and abstract normative fluents, based on counts-as relations between the elements they prescribe holding in a context entailed by a single state.

In more detail:

- D6.1 - Closure of the state does not remove any fluents from the input state.
- D6.2 - A fluent derived from another fluent conditional on a social context modelled by the state according to the institution's fluent derivation function is a member of the closed state.

Collectively, these conditions and the minimality constraint *close* a state under fluent derivations. Note that the fluent derivation operation is undefined if an institution's fluent derivation function has an output that is inconsistent with its input. For example $\mathcal{D}(\{\neg B\},A) \ni B$ or in words "A counts-as B in the context that B does not hold". In other cases, the fluent derivation operation is multi-valued if at least two rules defined by the institution's fluent derivation function are mutually inconsistent. For example $\mathcal{D}(\{\neg B2\},A) \ni B1$ and $\mathcal{D}(\{\neg B1\},A) \ni B2$, or in words "A counts-as B1 in the context that B2 does not hold" and vice versa "A counts-as B2 in the context that B1 does not hold". Such properties indicate an institution design problem, which should be resolved by the institution designer.

4.2.4 Abstracting Normative Fluents

This section presents a semantics for abstracting concrete normative fluents. The basic idea, depicted in Figure 6, is to establish new counts-as relations between concrete normative fluents and abstract normative fluents, based on the concrete concepts they prescribe counting-as more abstract concepts. Before we go into the actual semantics for abstracting concrete normative fluents, we describe the intuition and general semantics, then give numerous examples and finally the formalisation.

We call the relation between concrete and abstract normative fluents *deontological counts-as* and derive it based on three counts-as rule types (referring again to Figure 6). Firstly, based on counts-as between events according to an institution's event generation function. Here, we derive relations stating concrete normative fluents about events count-as an abstract normative fluent about an event. Secondly, based on counts-as between fluents according to an institution's fluent derivation function. Here, we derive relations stating concrete normative fluents about fluents count-as an abstract normative fluent about a fluent. Thirdly, based on counts-as between normative fluents themselves according to the deontological counts-as relation we define. Here, we derive relations stating *higher-order* concrete normative fluents prescribing normative fluents count-as a more abstract *higher-order* normative fluent prescribing a normative fluent. So, a *deontological* counts-as relation between concrete and abstract normative fluents is derived from more primitive *ontological* counts-as relations according to an institution's event generation

and fluent derivation functions, and deontological counts-as itself in order to derive deontological counts-as between higher-order normative fluents.

How the deontological counts-as relations between concrete and abstract normative fluents are derived is described as follows. The intuition is that concrete normative fluents count-as a more abstract normative fluent if and only if: the events or fluents that cause compliance with the concrete normative fluents (i.e., discharging or not violating) in turn count-as a certain institutional event to occur or fluent to hold that *guarantees* the abstract normative fluent is also complied with (i.e., discharged or not violated).

Following this intuition, we start by describing deontological counts-as for obligations. In reference to Figure 6, whenever any of a_0 to a_n occur or hold we are guaranteed b occurs/holds. If there is a concrete obligation imposed on one of $a_0, ..., a_n$ to occur/hold before a deadline d, then it is guaranteed that complying with the concrete obligation (discharging or not violating) means a more abstract obligation for b to hold before the same deadline d is also guaranteed to be discharged or not violated. Therefore, we derive a deontological counts-as relation stating that a concrete obligation on any of $a_0, ..., a_n$ before d *counts-as* a more abstract obligation for b to occur before d.

Prohibitions are different. If $a_0, ..., a_n$ count-as b, then unlike obligations we cannot apply modus ponens and say that prohibiting a_0 before d counts-as prohibiting b before d. The reason being, a_0 not occurring/holding *does not mean* b does not occur/hold. Thus, prohibiting a_0 on its own does not mean b should not occur. In other words, discharging or not violating a prohibition on a_0 before d does not guarantee that a prohibition on b before d is discharged or not violated. The reason is b can occur due to any of $a_1, ..., a_n$ occurring/holding (all counting-as b) and thus violate a prohibition on b before d. We might be tempted to apply modus tollens and say that b not occurring/holding means $a_0, ..., a_n$ do not occur/hold, therefore prohibiting b before d counts-as prohibiting $a_0, ..., a_n$ before d. However, this would be *concretisation* since $a_0, ..., a_n$ are more concrete than b (recall that concrete concepts count-as abstract concepts, and $a_0, ..., a_n$ count-as b). On the other hand, we are interested in *abstraction*. To summarise, unlike obligations modus ponens is incorrect to base counts-as between prohibitions on (since a prohibition is an obligation to the contrary), whilst modus tollens is inappropriate since it concretises rather than abstracts.

Instead, we derive a deontological counts-as relation between prohibitions stating that prohibiting all of $a_0, ..., a_n$ from occurring/holding before d counts-as a prohibition on a more abstract event/fluent b occurring/holding before d. This is based on the fact that counts-as is *ascriptive*, with reference to the discussion in Section2, defining all ways an abstract institutional concept can occur/hold when more concrete concepts occur/hold. Since abstract institutional events/fluents are *ascribed* by an institution's counts-as rules, if none of $a_0, ..., a_n$ occur/hold then b is not ascribed and therefore does not occur/hold. Note that this is entirely different from material implication. For example, given if it rain then the grass will be wet, the fact that it does not rain does not mean the grass is not wet. Counts-as rules in contrast make institutional facts possible, if some institutional fact B has no counts-as rules ascribing it, then that institutional fact cannot exist in an institution's social reality. In reference to the discussion in the background section, counts-as rules introduce institutional facts. Thus, complying with (discharging or not violating) all prohibitions on $a_0, ..., a_n$ occurring/holding before d guarantees that a prohibition on b before d is also complied with (discharged or not violated) - if $a_0, ..., a_n$ should not occur/hold before d then neither should b before d.

These informal semantics abstract concrete normative fluents with different concrete aims to an abstract normative fluent with a more abstract aim. Normative fluents' deadlines are also abstracted. However, as we observed when defining equivalences between normative fluents, the aim of an obligation is by definition obliged, whilst the deadline is prohibited and vice versa for prohibitions. Thus, the abstraction of obligation fluents' deadlines should be under the same semantics as prohibitions' aims and vice versa for prohibitions. So, given that a_0 counts-as b, a prohibition for z to occur before a_0 counts-as a prohibition for z to occur before b. Alternatively, we can just apply the equivalences between normative fluents such that we have an obligation for a_0 to occur before z that counts-as an obligation for b to occur before z, which is in turn equivalent to a prohibition for z to occur before b. Since a state with a prohibition

fluent also models an equivalent obligation fluent and vice versa, we define *deontological counts-as* based on the normative fluents a state *models* and obtain the abstraction of normative deadlines 'for free'.

This summarises the intuition behind deontological counts-as. More formally, deontological counts-as is defined as the function $DC^i : \Sigma^i \rightarrow 2^{\mathcal{F}^i_{norm}} \times \mathcal{F}^i_{norm}$. The function specifies for a state (S) a relationship ($\langle N, n\rangle \in DC(S^i)$) between sets of relatively concrete normative fluents (N) that count-as an abstract normative fluent (n) in the state S^i.

We exemplify the deontological counts-as function using our running case study. We focus on the EU-DRD's prescriptions formalised as an institution $\mathcal{I}^{drd}$. Counts-as between events according to the DRD's event generation $\mathcal{G}$ function state that a communications provider (*isp*) storing the content of a user's (*ada*) communications data (*storeData(isp, ada, content)*) counts-as (causes the institutional event of) storing personal data (*storeData(isp, ada, personal)*). Likewise, storing communications' metadata (*storeData(isp, ada, metadata)*) counts-as storing personal data.

Storing metadata or content data counts-as storing personal data. Thus, obliging metadata *or* obliging content data is stored immediately counts-as obliging personal data is stored immediately, since if a communications provider stores metadata or content data then it also stores personal data:

$$\langle\{obl(storeData(isp,\ ada,\ content),\ now)\}, obl(storeData(isp,\ ada,\ personal),\ now)\rangle \in DC^i(S^i)$$
$$\langle\{obl(storeData(isp,\ ada,\ metadata),\ now)\}, obl(storeData(isp,\ ada,\ personal),\ now)\rangle \in DC^i(S^i)$$

Prohibiting storing both content and metadata indefinitely counts-as prohibiting storing personal data indefinitely:

$$\langle\{pro(storeData(isp,\ ada,\ content),\ never),\ pro(storeData(isp,\ ada,\ metadata),\ never)\}, pro(storeData(isp,\ ada,\ personal),\ never)\rangle \in DC^i(S^i)$$

Higher-order normative fluents are abstracted using the same intuitions as first-order normative fluents, but with abstraction based on *deontological counts-as*. According to our case study, obliging an obligation to store content data counts-as obliging an obligation to store personal data. Likewise, obliging an obligation to store metadata counts-as obliging an obligation to store personal data.

$$\langle\{obl(obl(storeData(isp,\ ada,\ content),\ now),\ now)\},\ obl(obl(storeData(isp,\ ada,\ personal),\ now),\ now)\rangle \in DC^i(S^i)$$
$$\langle\{obl(obl(storeData(isp,\ ada,\ metadata),\ now),\ now)\},\ obl(obl(storeData(isp,\ ada,\ personal),\ now),\ now)\rangle \in DC^i(S^i)$$

Likewise, but for prohibitions, prohibiting storing metadata and prohibiting storing content data counts-as prohibiting storing personal data. Thus, obliging to immediately prohibit storing metadata *and* obliging to immediately prohibit content data counts-as obliging to immediately prohibit storing personal data. Only obliging to prohibit storing content data, does not mean it is obliged to prohibit storing personal data:

$$\langle\{obl(pro(storeData(isp,\ ada,\ content),\ never),\ now),\ obl(pro(storeData(isp,\ ada,\ metadata),\ never),\ now)\},\ obl(pro(storeData(isp,\ ada,\ personal),\ never),\ now)\rangle \in DC^i(S^i)$$

Higher-order prohibition abstraction semantics generalises the intuition of deontological counts-as for first-order prohibitions, but based on deontological counts-as itself. Prohibiting *all* concrete normative fluents that count-as a more abstract normative fluent, counts-as prohibiting the more abstract normative fluent.

According to our case study, indefinitely prohibiting obliging storing content data and prohibiting to oblige storing metadata, counts-as indefinitely prohibiting obliging storing personal data. Likewise, for prohibiting prohibitions.

$$\begin{aligned}
&\langle\{pro(obl(storeData(isp,\ ada,\ content),\ now),\ never),\\
&\quad pro(obl(storeData(isp,\ ada,\ metadata),\ now),\ never)\},\\
&\quad pro(obl(storeData(isp,\ ada,\ personal),\ now),\ never)\rangle \in DC^i(S^i)\\
&\langle\{pro(pro(storeData(isp,\ ada,\ content),\ never),\ never),\\
&\quad pro(obl(storeData(isp,\ ada,\ metadata),\ never),\ never)\},\\
&\quad pro(pro(storeData(isp,\ ada,\ personal),\ never),\ never)\rangle \in DC^i(S^i)
\end{aligned}$$

Abstracted normative fluents can also be further abstracted. To give an example, in the EU-DRD the *event* of storing personal data without someone's consent counts-as a non-consensual data processing event. Hence in the context that the agent Ada has not consented ($S \models \neg$*consentedDataProcessing(ada)*) we have the following deontological counts-as relation. It states the EU-DRD is effectively obliging an obligation for data to be processed non-consensually:

$$\begin{aligned}
&\langle\{obl(obl(storeData(isp,\ ada,\ personal),\ now),\ now)\},\\
&\quad obl(obl(nonConsensualDataProcessing(ada),\ now),\ now)\rangle \in \quad DC^i(S^i)
\end{aligned}$$

Deontological counts-as relations are also derived from the fluent derivation function $\mathcal{D}^i$. To exemplify, we take the previous example where we have an abstract obligation obliging Ada's data is stored non-consensually. Loosely speaking, the ECJ judged [21] that the EU-DRD, by obliging an obligation for non-consensual data processing, violated the EU-CFR's prohibition on *unfair data processing* (e.g., $pro($*unfairDataProcessing(ada),\ never*$)$). But how do we go from a second-order obligation for data to be processed non-consensually to violating a first-order prohibition on unfair data processing? One possibility is that the EU-CFR's fluent derivation function ($\mathcal{D}^{cfr}$) states that obliging non-consensual data processing counts-as unfair data processing, such that $\mathcal{D}^{cfr}(\emptyset, obl($*nonConsensualDataProcessing(ada)),\ now*$)) \ni$ *unfairDataProcessing(ada)*. Thus we have the following relation stating the second-order obligation for non-consensual data processing deontologically counts-as, more abstractly, obliging data is processed unfairly:

$$\begin{aligned}
&\langle\{obl(obl(nonConsensualDataProcessing(ada)),\ now),\ now)\},\\
&\quad obl(unfairDataProcessing(ada),\ now)\rangle \in DC^i(S^i)
\end{aligned}$$

However, obliging data is processed unfairly does not violate the EU-CFR prohibition on unfair data processing, $pro($*unfairDataProcessing(ada),\ never*$)$. This is unsurprising, the EU-CFR does not impose an explicit second-order prohibition, or contain any explicit higher-order norms for that matter (both in reality and in our formalisation). Unfair data processing is somehow derived from an obligation to oblige non-consensual data processing. One possibility is as follows: i according to the fluent derivation function obliging non-consensual data processing counts-as unfair data processing, thus ii obliging an obligation to process data non-consensually counts-as obliging unfair data processing. iii The EU-CFR considers whether data is processed unfairly or obliged to be processed unfairly as irrelevant, both are viewed as unfair data processing. iv Thus, an obligation to process data unfairly counts-as unfair data processing according to the fluent derivation function, $\mathcal{D}^{cfr}(\emptyset, obl(unfairDataProcessing(ada)), now)) \ni$ *unfairDataProcessing(ada)*. That is, normative fluents about abstract concepts are reduced to (ascribed as) those abstract concepts, in this way first-order norms can indirectly govern other norms.

The idea here does not mean what ought to be the case is the case (e.g., unfair data processing). Rather, unfair data processing is an abstract concept, which has many meanings, including obliging unfair data

processing itself. Note that this means not only is an obligation to process data unfairly reduced to unfair data processing, but so is a second-order obligation, a third-order obligation, etcetera. Formally:

$$\begin{aligned}&\langle\{obl(obl(unfairDataProcessing(ada), now), now)\},\\&\quad unfairDataProcessing(ada)\rangle \in DC^i(S^i)\\&\langle\{obl(obl(obl(unfairDataProcessing(ada), now), now), now)\},\\&\quad unfairDataProcessing(ada)\rangle \in DC^i(S^i)\\&...\end{aligned}$$

It is worth discriminating between *issuing* norms and *re-interpreting* norms at different abstraction levels, now that we have given a general argument for norm interpretation and its application according to constitutive rules. From an institution design perspective, it is most straightforward to issue norms at the abstraction level of an institutional fact we wish to regulate (e.g., prohibiting storing content data). An alternative method is to issue norms at a more concrete level of abstraction (e.g., prohibiting storing message content, telephone calls, etc.), where the concrete detached obligations/prohibitions are collectively re-interpreted as an obligation/prohibition on the more abstract institutional fact (e.g., prohibiting storing content data). The second approach is certainly possible, since an institution *defines* all of the ways in which an abstract institutional fact is constituted according to its counts-as rules, as we have discussed previously. Moreover, in our framework the domains we consider are finite and hence it is possible to enumerate all concretisations. However, this is a less convenient approach since an abstract concept can have a large number of concretisations that need to be accounted for. Moreover, when counts-as rules in an institution change over time (e.g., introducing a rule stating that storing message's subject counts-as storing content data) further concrete norms may need to be introduced (e.g., prohibiting storing message subjects) in order to continue regulating the same abstract fact (e.g., prohibiting storing content data). The most convenient approach to regulating abstract institutional facts or norms is to directly regulate those abstract institutional facts or norms and rely on abstraction of relatively concrete facts/norms in each social context to determine compliance.

Following this discussion, we formally define deontological counts-as, based on counts-as relations that hold in a state according to the event generation function, fluent derivation function and deontological counts-as itself. For convenience, we collect the event generation and fluent derivation counts-as relations into a single set $\mathcal{A}^i$ that forms the deontological counts-as function's *base cases*. Since deontological counts-as is also defined based on its own counts-as relations (in order to generalise to higher-order normative fluents), deontological counts-as is defined recursively. Formally, deontological counts-as is defined as:

Definition 7 Deontological Counts-as Given a state S^i, the deontological counts-as function $DC^i : \Sigma^i \rightarrow 2^{\mathcal{F}^i_{norm}} \times \mathcal{F}^i_{norm}$ is defined for the state $S^i \in \Sigma^i$ such that $DC^i(S^i)$ is the minimal (w.r.t. set inclusion) set of all pairs $\langle N', n'\rangle$ where $N' \neq \emptyset$ that satisfy the following:

$$N' = \{obl(a,d) \mid a \in A\} s.t. \langle A, b\rangle \in \mathcal{A}^i(S^i) \cup DC^i(S^i) \wedge n' = obl(b,d) \in \mathcal{F}'_{norm} \quad \textbf{or} \tag{D7.1}$$

$$N' = \{pro(a,d) \mid \langle A, b\rangle \in \mathcal{A}^i(S^i) \cup DC^i(S^i) \wedge a \in A\} \wedge n' = pro(b,d) \in \mathcal{F}^i_{norm} \tag{D7.2}$$

Where the set of abstracting counts-as relations $\mathcal{A}^i(S^i)$ for the state S^i is defined as:

$$\mathcal{A}(S^i) = \{\langle\{a\}, b\rangle \mid X \in \mathcal{X}^i, a \in \mathcal{E}^i, b \in \mathcal{G}^i(X,a) \wedge S^i \models pow(b)\} \cup \tag{D7.3}$$

$$\{\langle\{a\}, b\rangle \mid X \in \mathcal{X}^i, a \in \mathcal{F}^i, b \in \mathcal{D}^i(X,a)\} \tag{D7.4}$$

A state closed under deontological counts-as function is the function $\overline{DC}^i : \Sigma^i \rightarrow \Sigma^i$, such that $S' = \overline{DC}^i(S^i)$ iff it *minimally* (w.r.t. set inclusion) satisfies all of the following conditions:

$$S^i \subseteq S' \tag{D7.6}$$

$$\exists \langle N', n'\rangle \in DC^i(S^i) : N' \subseteq S' \wedge n' \in \mathcal{F}^i_{anorm} \Rightarrow n' \in S' \tag{D7.7}$$

In more detail. Concrete obligations count-as a more abstract obligation according to D7.1. Concrete prohibitions count-as a more abstract prohibition according to D7.2. These counts-as relations are derived from relations between concrete concepts counting-as an abstract concept defined by the event generation function and fluent derivation function according to D7.3 - D7.4 (the base cases) and with respect to deontological counts-as itself since it is defined recursively.

Deontological counts-as does not describe whether normative fluents in a state S^i are abstracted, but rather whether they *could be*. Closing a state under deontological counts-as is according to the operation $\overline{DC}^i$. Condition D7.6 ensures any fluents already in the state remain in the state. Condition D7.7 ensures if concrete normative fluents, should they hold in a state are abstracted to a normative fluent, and they do indeed hold, then the abstracted normative fluent also holds. Note that in D7.7 it is ensured only normative fluents that belong to the abstract set of normative fluents can hold in a state due to being derived from concrete normative fluents. Consequently, deontological counts-as only adds non-inertial abstract normative fluents to a state.

Note that $\overline{DC}^i$ is a partial function if there is a fault in the institutional specification. For example, if an institution obliges an event a to occur in some state, and the event a generates the event b in that state, then b is also obliged to occur in that state. However, if a generates the event b *conditional* on b not being obliged then there is a problem. We have that b is obliged since a is obliged. But, if b is obliged then a does not count-as b, thus obliged a does not count-as obliged b and so there is no obligation for b to occur. Again, in principle there is nothing wrong with the possibility of this paradox occurring since it is an institutional design fault. If we have $\overline{DC}^i(S) = \bot$ then we have detected an institutional design problem in the state S for the institution designer to rectify.

4.2.5 Models

In this section we provide a multi-level governance institutional *model* definition, which captures how each institution in a multi-level governance institution evolves from one state to the next, driven by observable events that potentially generate institutional events in state transitions. A model is defined in response to a *trace* of observable (exogenous) events.

The approach we take is to put together all of the previous operations according to the following description. An institution starts at an initial state that includes the institution's initial set of inertial fluents (Δ^i) and the state closed under the fluent derivation and deontological counts-as operations. The institution transitions between states with a set of events generated by the event generation operation in response to an observable event in the event trace. Each state transitioned to contains the fluents that held in the previous state that were not terminated, any newly initiated fluents as well as closing the state under the fluent derivation and deontological counts-as operations. Additionally, an institution's evolution is affected by the evolution of other institutions it governs. This means that a higher level institution's state includes normative fluents from lower level institutions it governs. These normative fluents are 'passed up' to the higher level institution in order to abstract the lower levels normative fluents and determine if they are compliant in their abstract interpretation.

We begin by defining the initial state of each individual institution. Formally and described subsequently:

Definition 8 Initial States The initial state S_0^i for each individual institution $\mathcal{I}^i$ w.r.t. $\mathcal{ML} = \langle \mathcal{T}, R \rangle$ and a tuple of initial states $\langle S_0^1, ..., S_0^n \rangle$ is the set S_0^i if and only if S_0^i *minimally* (w.r.t. set inclusion) satisfies the following:

$$S_0^i \subseteq \Delta^i \tag{D8.1}$$

$$\exists \langle h, i \rangle \in R, n \in (\mathcal{F}^h_{cnorm} \cup \mathcal{F}^h_{anorm}) \cap \mathcal{F}^i_{ninert} : n \in S_0^h \Rightarrow n \in S_0^i \tag{D8.2}$$

$$S_0^i = FD^i(S_0^i) \tag{D8.3}$$

$$S_0^i = \overline{DC}^i(S_0^i) \tag{D8.4}$$

- D8.1 - an institution's initial set of inertial fluents is included in the institution's initial state.
- D8.2 - if the institution governs a lower level institution then it contains any normative fluents holding in that lower level institution's initial state.
- D8.3 - the initial state is closed under the fluent dependency operation, such that all derived fluents are included.
- D8.4 - the initial state is closed under deontological counts-as such that all abstracted normative fluents are included.

Now we define which fluents are initiated and terminated from one state to the next in response to a generated set of events (i.e., by the event generation operation). The set of fluents that are initiated ($INIT^i$) and terminated ($TERM^i$) from one state to the next are formally defined as and subsequently described:

Definition 9 Fluent Initiation and Termination The sets of all initiated and terminated fluents for $\mathcal{I}^i$ are respectively defined with the functions $INIT^i : \Sigma^i \times 2^{\mathcal{E}^i} \rightarrow 2^{\mathcal{F}^i}$ and $TERM^i : \Sigma^i \times 2^{\mathcal{E}^i} \rightarrow 2^{\mathcal{F}^i}$:

$$INIT^i(S^i, E^i) = \{f \mid \exists e \in E^i, \exists X \in \mathcal{X}^i : f \in \mathcal{C}^{i\uparrow}(X, e) \wedge S^i \models X\} \tag{D9.1.1}$$

$$TERM^i(S^i, E^i) = \{f \mid \exists e \in E^i, X \in \mathcal{X}^i : S^i \models f \wedge f \in \mathcal{C}^{i\downarrow}(X^i, e) \wedge S^i \models X \quad \textbf{or} \tag{D9.2.1}$$

$$S^i \models f \wedge (viol(f) \in E^i \vee disch(f) \in E^i)\} \tag{D9.2.2}$$

Condition D9.1.1 specifies the set of initiated inertial fluents according to the institution's consequence function. An inertial fluent is initiated by the state consequence function conditional on an event occurring and a social context holding in the state. Conversely, D9.2.1 specifies that the set of terminated inertial fluents includes any inertial fluents terminated according to the institution's consequence function. Condition D9.2.2 states that any discharged or violated inertial (concrete) normative fluents are also terminated, meaning dishcarged/violated normative fluents do not persist by default[3].

A state transition operation ($TR^i(S^i, E^i)$) produces a new institutional state based on the previous state (S^i) due to the occurrence of events (E^i). The new state includes any inertial fluents that held in the previous state and have not been terminated, any newly initiated fluents, normative fluents holding, and the state's closure under the fluent derivation and deontological counts-as operations.

Definition 10 State Transitions The state transition operation $TR^i : \Sigma^i \times 2^{\mathcal{E}^i} \rightarrow \Sigma^i$ is defined for each institution $\mathcal{I}^i$, a state S^i and a set of events E^i w.r.t. the states of other institutions $\langle S^1_j, ..., S^n_j \rangle$ holding at the same time and $\mathcal{ML} = \langle \mathcal{T}, R \rangle$, such that $TR^i(E^i S^i) = S'$ iff S' *minimally* (w.r.t. set inclusion) satisfies all of the following conditions:

$$\forall f \in (S^i \cap \mathcal{F}^i_{inert}) \backslash TERM^i(S^i, E^i) \Rightarrow f \in S' \tag{D10.1}$$

$$\forall f \in INIT^i(S^i, E^i) \Rightarrow f \in S' \tag{D10.2}$$

$$\exists \langle h, i \rangle \in R, n \in (\mathcal{F}^h_{cnorm} \cup \mathcal{F}^h_{anorm}) \cap \mathcal{F}^i_{ninert} : n \in S^h_j \Rightarrow n \in S' \tag{D10.3}$$

$$S' = FD^i(S') \tag{D10.4}$$

$$S' = \overline{DC}^i(S') \tag{D10.5}$$

- D10.1 - non-terminated inertial fluents persist from one state to the next, following the common sense law of inertia.
- D10.2 initiated fluents hold in the next state.
- D10.3 a higher level institution's state contains all normative fluents that hold in the same state of a lower level institution the higher level governs.

[3] Meaning, if you discharge or violate an obligation you are no longer obliged and likewise for prohibitions. In some cases, it can make sense for a discharged/violated normative fluent to persist. For example, if you violate a prohibition on murder, it is often the case that you are still prohibited from committing murder. For an extensive discussion on when it does and does not make sense for obligations and prohibitions to persist after discharge/violation see [36].

- D10.4 the newly transitioned to state includes all normative fluents that can be derived according to the fluent derivation operation.
- D10.5 the newly transitioned state contains all normative fluent abstractions according to deontological counts-as.

We now proceed to event traces. The trace a model is defined in response to is a sequence of observable events recognised by the institutions involved in a multi-level governance relationship. That is, it is a trace of only those events that can affect the institutions involved, driving their evolution and the multi-level governance institution's evolution as a whole. Each event in a trace needs to be recognised by at least one institution to drive its evolution over time. We call such a trace, a *composite event trace*, formally:

Definition 11 Composite Event Trace Let $\mathcal{ML} = \langle \mathcal{T}, \mathcal{B} \rangle$ be a multi-level governance institution where $\mathcal{T} = \langle \mathcal{I}^1, ..., \mathcal{I}^n \rangle$. $ctr = \langle e_0, ..., e_k \rangle$ is a composite trace for $\mathcal{ML}$ iff $\forall j \in [0,k], \exists i \in [n] : e_j \in \mathcal{E}^i_{obs}$

Synchronisation issues can arise between institutions. These issues occur if a composite trace includes an event recognised by one institution, therefore driving its state forward, but not recognised by another institution meaning its state does not evolve. If an event in a composite trace is not recognised by an institution, then the institution should still transition to a new state to ensure it is evolving at the same rate as other institutions. We replace unrecognised events by the event of no change, the null event, in a *synchronised trace* for each institution derived from a composite trace. Formally:

Definition 12 Synchronised Trace Let $\mathcal{I}$ be an institution, and $ctr = \langle e_0, ..., e_k \rangle$ be a composite event trace. A trace $str = \langle se_0, ..., se_k \rangle$ is a *synchronised trace* of ctr for $\mathcal{I}$ iff $\forall i \in [0,k]$: if $e_k \in \mathcal{E}_{obs}$, $se_k = e_k$ and $se_k = e_{null}$ otherwise.

We now define a *multi-level governance institution model*. A model comprises sequences of states (S) and events (E). One state sequence for each individual institution (S^i) and one sequence of event sets for each individual institution (E^i) driving its state transitions. A model is defined in response to a composite trace such that the corresponding synchronised trace for each institution drives its evolution over time, causing events to occur and driving state transitions forward. Each state and set of transitioning events is defined for each institution assuming that the states and set of transitioning events exist for every other institution. Formally:

Definition 13 Multi-level Governance Institution Model Let $M = \langle M^1, ..., M^n \rangle$ be a tuple of state and event sequence pairs for each institution $\mathcal{I}^i$ with typical element $M^i = \langle S^i, E^i \rangle$ where $S^i = \langle S^i_0, ..., S^i_{k+1} \rangle$ and $E^i = \langle E^i_0, ..., E^i_k \rangle$. Let ctr be a composite trace for $\mathcal{ML} = \langle \mathcal{T}, R \rangle$ and $str^i = \langle se^i_0, ..., se^i_k \rangle$ be a synchronised trace of ctr for each institution $\mathcal{I}^i$. Let $\forall i \in [1,n], \forall j \in [0,k] : GR^i(S^i_j, E^i_j)$ be the event generation operation for $\mathcal{I}^i$ w.r.t. $\langle S^1_j, ..., S^n_j \rangle$ and $\langle E^1_j, ..., E^n_j \rangle$. Let $\forall i \in [1,n], \forall j \in [0,k] : TR^i(S^i_j, E^i_j)$ be the state transition operation for each institution $\mathcal{I}^i$ w.r.t. $\langle S^1_j, ..., S^n_j \rangle$. The tuple M is a model of $\mathcal{ML}$ w.r.t. ctr if and only if:

$$\forall i \in [1,n] : S^i_0 \text{ is the initial state of each institution } \mathcal{I}^i \text{ w.r.t. } \langle S^1_0, ..., S^n_0 \rangle \quad \text{(D13.1)}$$

$$\forall i \in [1,n], \forall j \in [0,k] : E^i_j = GR^i(S^i_j, \{se^i_j\}) \quad \text{(D13.2)}$$

$$\forall i \in [1,n], \forall j \in [0,k] : S^i_{j+1} = TR^i(S^i_j, E^i_j) \quad \text{(D13.3)}$$

- D13.1 - the initial state of each individual institution, which is defined with respect to the initial state of every other institution (meaning a higher-level institution includes normative fluents from a lower-level institution).
- D13.2 - each institution's set of events transitioning to a new state comprises all events generated from the corresponding event in the synchronised trace and the previous state according to the event generation operation. The event generation operation is also defined with respect to the states and events from every other institution, such that norm compliance events are 'passed up' between governance levels.

– D13.3 - the next state transitioned from the previous state by the set of transitioning events. The state transition operation is also defined with respect to the states and events from every other institution, such that normative fluents are 'passed up' between governance levels.

This concludes multi-level governance institution semantics.

4.2.6 Compliance Monitoring

A multi-level governance institution model monitors the compliance of other institution's regulations and their outcomes. A model determines if the concrete regulatory effects of one institution are non-compliant with the more abstract regulations of a higher level institution in a particular context. This is by 'passing up' any concrete normative fluents from a lower level institution to the higher level institution that governs it. Then, abstracting those concrete normative fluents in the higher level institution according to the higher level institution's abstracting constitutive rules (i.e., under the semantics of deontological counts-as). Then, taking the more abstract interpretation of the lower levels' concrete normative fluents, generating any discharge and violation events of the higher level institution's higher-order norms that oblige/prohibit the abstracted lower level institution's concrete norms. All that is needed to determine if there is non-compliance is to collect a set of violation events from the multi-level governance model for each institution. Formally, the set of sets of violation events for each individual institution denoting non-compliance is:

Definition 14 Multi-level Governance Violations Let $\mathcal{ML} = \langle \mathcal{T}, R \rangle$ be a multi-level governance institution and $M = \langle M^1, ..., M^n \rangle$ a model of $\mathcal{ML}$ w.r.t. a composite trace *ctr* such that $\forall i \in [n] : M^i = \langle S^i, E^i \rangle, S^i = \langle S^i_0, ..., S^i_{k+1} \rangle, E^i = \langle E^i_0, ..., E^i_k \rangle$. The tuple $V = \langle V_1, ..., V_n \rangle$ is the set of multi-level governance violations for $\mathcal{ML}$ w.r.t. *ctr* if and only if:

$$\forall i \in [1,n] : V_i = \{e \mid \exists f, j : f \in \mathcal{F}^i_{cnorm} \cup \mathcal{F}^i_{anorm}, j \in [k] \wedge viol(f) \in E^i_j \wedge e = viol(f)\} \qquad \text{(D14.1)}$$

Non-compliance is found if the set of violation events is non-empty. For an institution governing a society this implies that the society is non-compliant (either in reality if compliance checking is performed before run-time or hypothetically if not). For a higher level institution governing a lower level institution non-compliance denotes that the regulatory effects are non-compliant if the violated norms belong to the higher level institution. Such non-compliant regulatory effects can be due to having a more abstract, non-compliant, meaning.

5 Computational Framework

In this section we provide a practical approach to reasoning about multi-level governance with a computational framework that corresponds to the formal framework. The idea of the computational framework is to use Answer-Set Programming, a declarative programming language where the commonly accepted syntax is AnsProlog, which we describe in section 5.1. An AnsProlog program in our framework produces models of a multi-level governance institution for a trace of events.

There are two main components in the computational framework. Firstly, a general AnsProlog program that implements the multi-level governance semantics from the previous section. Secondly, specific AnsProlog programs that represent each individual institution, their multi-level governance relationships and norm abstractions. By executing these AnsProlog programs together we can automate compliance checks. We describe, for brevity, these AnsProlog programs in section 5.2 and give the representation in full in Appendix B. To show that the computational framework provides a practical implementation of the formal framework we provide *soundness* and *completeness* theorems between the two frameworks in section 5.5. The theorems are proven in Appendix C. Moreover, we provide general properties of the computational framework's complexity for given institutions in a multi-level governance relationship in Section 5.6

The corresponding AnsProlog programs that represent institutions, their multi-level governance relationships and norm abstractions are specific to each set of institutions in a multi-level governance relationship. Consequently, it would be a lot of effort for a user to manually write AnsProlog programs for each institution they wish to use in a multi-level compliance check. For this reason, we use a compiler that takes as input institutions described in a high-level language, similar to the formal representation used for institutions albeit with additional useful constructs such as variables and types. The compiler outputs executable AnsProlog programs for individual institutions, their norm abstractions and the general AnsProlog program representing multi-level governance semantics. By executing these compiled programs together, we can automatically detect compliance without having to write AnsProlog code by hand. We give an overview of the implemented compiler and demonstrate the result of executing the AnsProlog programs for our running case study, with a compliance check corresponding to the real-world judgements of the European Court of Justice, in Section 5.3.

5.1 Answer Set Programming

Answer-Set Programming is a non-monotonic logic-programming language [8, 32], for declaring problems according to the syntax of AnsProlog as a set of first-order rules. AnsProlog is fully declarative in the sense that the ordering of logical formulae (horn clauses and facts) makes no semantic difference. Executing an AnsProlog program solves the declared problems by first running a grounder, which grounds all rules, replacing variables with ground terms, and then running a solver against the ground program. A solver computes the set of *answer-sets*, where each answer-set is a *model* of the AnsProlog program and a solution to the problem declared. Answer-sets are computed according to the stable-model semantics [32].

We use AnsProlog for two main reasons. Firstly, it provides a natural representation of individual and multi-level governance institutions, where institutions' functions are represented as AnsProlog rules. Secondly, it supports meeting the goal of our framework: automatically checking different contexts, or traces of exogenous events, for whether regulations in a lower level institution are non-compliant. Using AnsProlog, a single trace of events can be supplied to check for compliance, but we can also specify a partial trace and that all variants of that trace must be used to check compliance or even all possible traces up to a specific length must be checked for compliance. It is also possible to declare that each answer-set produced must have a particular property, such as 'there must be at least one violation of a norm in a higher level institution'. In this case the property implies that if no answer-sets are produced then there is full compliance for all traces up to a certain length. In summary, Answer-Set Programming provides a natural representation of multi-level governance institutions and an easy way to perform a contextual search for compliance.

There are many answer-set solvers available (e.g., [20, 30]). We briefly reintroduce the main definitions to give context for what follows, focussing on the syntax of the CLINGO [30] grounder and solver making use of a number of its unique constructs. In more detail, an AnsProlog program is built from atoms and predicates. Predicates can be ground, such as `lays_eggs(slinky)` or non-ground predicates containing variables representing the ground instance schemas, such as `bird(X)`. Atoms and predicates can be weakly negated, such as `not` [4]. A rule r is typically of the form `p_0 :- p_1, ..., p_n` comprising a head atom denoted $H(r)$ and a set of body literals denoted $B(r)$, which can be delineated into the positive body atoms $B^+(r)$ and atoms appearing negated in the body $B^-(r)$. A rule r can also be a fact by having an empty body such that $B(r) = \emptyset$ containing only a single head atom such as `lays_eggs(slinky)`. To give an example adapted from [8], the following program declares that a bird is an animal that lays eggs and is not a reptile, a reptile is an animal that lays eggs that is not a bird and slinky is an animal that lays eggs:

```
bird(X) :- lays_eggs(X), not reptile(X).
reptile(X) :- lays_eggs(X), not bird(X).
```

[4] we ignore the case of strong negation since it is unnecessary in our use of AnsProlog

```
lays_eggs(slinky).
```

A (total) interpretation of an answer-set program is a truth-assignment to literals, comprising a set of the atoms assigned the value of 'true'. An answer-set is a *minimal* interpretation containing all atoms that are *justified* in being true. Precisely, for a rule r, the head atom denoted $H(r)$ is *justified* in being true if all positive body atoms, denoted $B^+(r)$, are true, and none of the weakly negated body atoms, denoted B^-, are true. This implies facts are always justified in being true (e.g., `lay_eggs(slinky)`). Looking at the previous example there can be more than one answer-set. If `bird(slinky)` is in an interpretation then `reptile(slinky)` cannot be in the interpretation for it to be an answer-set, and vice versa. These answer-sets are:

- { `bird(slinky), lays_eggs(slinky)` }
- { `reptile(slinky), lays_eggs(slinky)` }

Determining if an interpretation is an answer-set requires knowing which atoms are justified according to the program's rules. In the presence of weak negation this means we should only consider the rules that do not contain weakly negated atoms that are in the answer-set. Furthermore, for those rules that remain we do not need to consider their weakly negated literals to determine if the head is justified. Removing all rules in a program with weakly negated literals that are in an interpretation and all weakly negated literals from the remaining rules is called the *reduct* of the program, formally from [31]:

Definition 15 Reduct Let Π be an Answer-Set Program and X an interpretation of Π, the reduct denoted Π^X is the set:

$$\{H(r) \leftarrow B^+(r) \mid r \in \Pi \text{ and } B^-(r) \cap X = \emptyset\}$$

We want to determine for a reduct and a set of atoms, whether that set of atoms is closed under the program (containing all justified atoms) and whether it is minimal (containing no atoms that are not justified). To give an example, if we have a reduct
$\Pi = \{$ `lays_eggs(slinky). bird(slinky) :- lays_eggs(slinky).` $\}$, then the set
{`lays_eggs(slinky), bird(slinky), some_atom`} is closed since
`lays_eggs(slinky)` and `bird(slinky)` are justified but it is not minimal due to the presence of `some_atom`. Formally adapted from [31]:

Definition 16 Let Π be a reduct and X a set of atoms. The set of atoms X is closed under Π^X if for all $r \in \Pi^X$, we have $H(r) \in X$ iff $B^+(r) \subseteq X$. The smallest set of atoms closed under Π^X is denoted $Cn(\Pi^X)$.

An answer-set is simply a minimal interpretation of a reduct of the program for the interpretation:

Definition 17 Answer-Set Let Π be an Answer-Set Program and X be an interpretation of Π and Π^X be the reduct of Π w.r.t. X. X is an answer-set of Π iff $X = Cn(\Pi^X)$.

In addition to the Answer-Set Programming semantics given above we use three constructs present in CLINGO [30]. Namely, *constraints*, *choice rules* and *conditional literals*. Constraints are a special type of rule of the form $:-\mathtt{b}_1,...,\mathtt{b}_n.$ representing a rule with falsity in the head such that if all of $\mathtt{b}_1$ to $\mathtt{b}_n$ are true in an interpretation then there is a contradiction and therefore the interpretation is not an answer-set. Choice rules are of the form $\{\mathtt{a}_1,...,\mathtt{a}_n\} :- \mathtt{b}_1,...,\mathtt{b}_n.$, meaning that any atom in $\mathtt{a}_1,...,\mathtt{a}_n$ can arbitrarily be picked for inclusion in an answer-set if $\mathtt{b}_1,...,\mathtt{b}_n.$ is true. Aggregates are present in the body of rules and are of the form $\mathtt{l}\{\mathtt{b}_1;...;\mathtt{b}_n\}\mathtt{u}$ where $\mathtt{l}$ and $\mathtt{u}$ are positive integers meaning that at least $\mathtt{l}$ and at most $\mathtt{u}$ elements of $\mathtt{b}_1,...,\mathtt{b}_n$ must be true for the aggregate to be true. Omitting $\mathtt{l}$ or $\mathtt{u}$ removes the respective constraint. Finally, conditional literals are special literals that can be contained in the body of a rule or within in an aggregate and are of the form $\mathtt{b}_1 : \mathtt{b}_2,...,\mathtt{b}_n$. They follow the semantics of material implication; conditional literals are true if the head is true or the body is false. Note that there are no conditional literals within aggregates, rather a rule of the form $\mathtt{l}\{\mathtt{b}_1 : \mathtt{b}_2,...,\mathtt{b}_n\}\mathtt{u}$ means that $\mathtt{b}_1$ is counted as being true when restricted to the domain of $\mathtt{b}_2,...,\mathtt{b}_n$. Without variables this simply means that $\mathtt{b}_1$ is counted as true when $\mathtt{b}_2,...,\mathtt{b}_n$ is true.

5.2 Mapping

In this section we give the general idea behind mapping between the formal representation and semantics of multi-level governance institutions and their representation in AnsProlog. For a detailed account, we refer the reader to Appendix B.

The approach we take is to represent the events and fluents that can hold in each institution, as AnsProlog *facts*, and the functions as non-factual AnsProlog *rules*. Each rule antecedent corresponds to the parameters the functions take. For the event generation and state consequence functions, the corresponding AnsProlog rules' antecedents comprise the occurrence of events and the state conditions. For the fluent derivation function, expressing constitutive rules of the form "fluent A derives (counts-as) fluent B in context C" the corresponding AnsProlog rules' antecedents comprise conditions on the state containing fluent A and modelling the context C. The consequence of a rule corresponds to the *effect* of the function's returned value on a multi-level governance institution model. The resulting effect is an event caused to occur according to $\mathcal{G}$, the initiation and termination of fluents according to $\mathcal{C}$, and non-inertial fluents holding in a state according to a fluent derivation function $\mathcal{D}$.

Multi-level governance institution semantics is represented in AnsProlog as more general rules. For example, stating that if an inertial fluent is initiated then it holds until it is terminated. The exception is the semantics of deontological counts-as, which is represented as a set of specific AnsProlog rules that ensure normative fluent abstractions are included in states. Finally, composite traces are mapped to a corresponding AnsProlog representation as sets of facts, each stating that an event has been observed at a particular point in time.

The computational framework's AnsProlog rules make use of the same common predicates used previously in work extending InstAL to settings with multiple institutions [54, 55, 56]. In turn these are similar to Event Calculus [53] constructs. To give context for what follows, the predicates are summarised in their non-ground form:

- `holdsat(F,In,I)` denotes that the fluent `F` holds in the institution `In` at time `I`.
- `observed(E,In,I)` denotes that the event `E` is observed by the institution `In` at time `I` corresponding to the exogenous event that has occurred in the synchronised trace for the institution.
- `occurred(E,In,I)` denotes that the event `E` occurs in the institution `In` at time `I`.
- `initiated(F,In,I)` denotes that the fluent `F` is initiated in the institution `In` at time `I`.
- `terminated(F,In,I)` denotes that the fluent `F` is terminated in the institution `In` at time `I`.
- `instant(I)` denotes `I` is a time instant.
- `start(I)` denotes `I` is the first time instant.
- `final(I)` denotes `I` is the last time instant.
- `next(I,J)` denotes `J` is a time instant that is strictly after `I` such that there is no time instant between `I` and `J`.

The aforementioned predicates are used in both antecedents and consequents of rules. Such as, stating *conditional* on particular fluents (not) holding in a state one event causes another event to occur. This means events in function parameters correspond to `occurred/3` whilst state conditions correspond to sets containing positive and negative `holdsat/3` predicates[5].

5.2.1 Multi-level Governance Translation

The main idea is to translate the formal representation of a multi-level governance institution and its semantics into a a set of AnsProlog rules.

The translation for representing a multi-level governance institution, comprising a set of AnsProlog rules, comprises the translation of the individual institutions and the translation of the multi-level governance links between them.

[5] an empty state condition (the empty set) is always true and replaced with the special atom `#true` for technical reasons.

Starting with individual institutions, their institutional events and fluents are represented as AnsProlog facts. For example in the EU-CFR, `exConsent` is an exogenous event, `consent` is an institutional event and `consentedDataProcessing(ada,isp)` is an inertial fluent:

```
evtype(exConsent,cfr,ex).
evtype(consent,cfr,in).
ifluent(consentedDataProcessing(ada,isp),cfr).
```

An institution's event generation function is translated to rules. Each rule containing an `occurred/3` atom in the head representing the event that is caused to occur. Each rule's body comprising an `occurred/3` atom representing the *causal event*, and positive and negative `holdsat/3` atoms representing the rule's state conditions. For example, the following rule states that non-consensual data processing occurs if Ada's personal data has been stored and she has not consented, where non-consensual data processing is empowered to occur:

```
occurred(nonConsensualDataProcessing(ada),cfr,I) :-
   occurred(storeData(isp,ada,personal),cfr,I),
   holdsat(pow(cfr,nonConsensualDataProcessing(ada)),cfr,I),
   not holdsat(consentedDataProcessing(ada,isp),cfr,I), instant(I).
```

An institution's consequence function is translated to AnsProlog rules, using `initiated/3` and `terminated/3` atoms in the head for the initiation and termination of fluents. Each fluent initiation and termination rule's body comprises an `occurred/3` atom representing the event causing a fluent to be initiated/terminated, and positive and negative `holdsat/3` atoms representing the context in which the fluent initiation/termination is conditional on. For example, in the EU-CFR institution the fluent stating ada has consented to data processing is initiated if she consents. In the EU-DRD institution the obligation to oblige metadata is stored is initiated (i.e., imposed) when Ada uses electronic communications:

```
initiated(consentedDataProcessing(ada,isp),cfr,I) :-
   occurred(consent(ada,isp),cfr,I), instant(I).
initiated(obl(obl(storeData(isp,ada,metadata),now),now),drd,I) :-
   occurred(useElectronicCommunication(ada,isp),drd,I), instant(I).
```

An institution's fluent derivation function is represented as AnsProlog rules with `holdsat/3` atoms in the head and body. For example, in the institution 'unfair data processing' is derived from an obligation to process data non-consensually:

```
holdsat(obl(nonConsensualDataProcessing(ada),now),cfr,I) :-
   holdsat(unfairdataprocessing(ada),cfr,I), instant(I).
```

The links between institutions are also represented as rules. Generally, a link from a lower level institution institution L to a higher level institution H that governs L, comprises rules with `occurred/3` and `holdsat/3` atoms in the head. The occurred rules state a norm discharge/violation event occurs in the institution H when it occurs in the institution L. Likewise, a normative fluent holds in H when it holds in L. All of these rules are produced such that only the discharge/violation events occurring in L and consequently in H are about normative fluents L imposes. Likewise, further rules state only normative fluents hold in H when they hold in L for those normative fluents that L itself imposes. Thus, if L receives norm discharge/violation events or normative fluents from another institution, these do not get passed up to H from L.

To give an example, the following rule states that when the prohibition on deleting data before 12 months holds in the UK-DRR, then it also holds in the EU-DRD for checking compliance.

```
holdsat(pro(deleteData(isp, ada, metadata),time(m12)),drd,I) :-
   holdsat(pro(deleteData(isp, ada, metadata),time(m12)),drr,I),instant(I).
```

Abstraction according to deontological counts-as is also represented as rules, where the head is a holdsat atom representing the abstract normative fluent conditional on some concrete normative fluents holding. For brevity we do not give details but refer the interested reader to Appendix B.1.

Finally, the semantics is represented as more general rules. For example, the following rules state that an inertial fluent holds in a state if it is initiated or if it held in the previous state and was not terminated (capturing the common-sense law of inertia):

```
holdsat(P,In,J) :- holdsat(P,In,I),
    not terminated(P,In,I), next(I,J), ifluent(P, In).
holdsat(P,In,J) :- initiated(P,In,I), next(I,J), ifluent(P, In).
```

The translation, where here we give its intuition (see Appendix B.1 for details), allows us to automate compliance checks for institutions operating in a multi-level governance relation. Moreover, the translation to an AnsProlog program is equivalent in the sense of producing answer-sets that correspond to the formal models in the formal framework. That is, we have soundness and completeness properties which we present later in Section 5.5.

5.3 Specification Language and Compiler

The computational framework is implemented as a high-level specification language for declaring institutions in a multi-level governance relationship and a compiler for producing executable answer-set programs. The implementation extends the InstAL specification language and compiler [12]. The system is demonstrated with the results of formalising our case study in the high-level specification language and executing the compiled AnsProlog code for a trace of exogenous events.

The high-level specification language allows users to declare individual institutions and links between them. The language constructs provided to users correspond to much the same representation elements as in the formal framework, with the addition of types and variables to provide users with a concise way to represent institutions over large domains. Below, is a fragment of the EU-DRD institution from our case study written in the specification language, which we subsequently describe:

```
1  institution drd;
2  type Agent;
3  type CommServProv;
4
5  exogenous event exUseElectronicCommunication(Agent, CommServProv);
6  inst event useElectronicCommunication(Agent, CommServProv);
7
8  fluent pow(useElectronicCommunication(Agent, CommServProv);
9  obligation fluent obl(obl(storeData(CommServProv, Agent, Data), now), now);
10
11 exUseElectronicCommunication(Ag, Co) generates useElectronicCommunication(Ag, Co);
12 initially pow(useElectronicCommunication(Ag, Co));
13 useElectronicCommunication(Ag, Co) initiates
14         obl(obl(storeData(Co, Ag, metadata), now), now);
```

The name of the institution is declared on line 1, and the types of agent and communications service provider on lines 2 and 3. Lines 5 and 6 declare the institution's events. Lines 8 and 9 declare the institution's fluents. Line 11 corresponds to the notion of the mapping between events provided by an institution's event generation function. It states that the exogenous event of using electronic communications generates an institutional event of using electronic communications. Line 12 declares that a fluent holds in an institution's initial state, in this case the fluent empowering the event to use electronic communication to occur in the institution. Line 13 represents a mapping provided by the institution's consequence function, in this case the descriptive rule stating that if an agent uses electronic communication then it is obliged that the communications provider is obliged to store the communications' metadata immediately.

A separate file, called a domain file, declares the terms of each type, such as who is an agent or a communications provider. The compiler processes the files declaring the individual institutions and outputs a set of AnsProlog files representing each institution and their semantics.

The AnsProlog files are then processed by the grounder and AnsProlog solver CLINGO, together with a timeline program declaring a sequence of events and a special program declaring the links between each institution. A short example of a link between two institutions is the following program. The program specifies the governance relation between the institutions on lines 1 and 2. On line 3 a rule states an obligation in a lower-level institution that holds and is a non-inertial fluent in the higher-level institution, also holds in the higher-level institution.

```
1 governs(cfr, drd).
2 governs(drd, drr).
3 holdsat(obl(A, D), HIn, I) :-
4    holdsat(obl(A, D), LIn, I), nifluent(obl(A, D), HIn), governs(HIn, LIn).
```

The result of executing the AnsProlog programs together is a multi-level governance institution model for the sequence of events provided. The model can be inspected for compliance of regulations, denoted with compliance events, and other properties.

5.4 Running the Case Study

We have written the case study in the high-level computational framework specification language. By compiling from the specification language to an AnsProlog representation we are able to assess compliance in our case study's multi-level governance institution. This is by executing the resulting AnsProlog program together with a trace of events.

The case study is instantiated for a domain comprising four types. Firstly the agents acting in the system (`ada` and `charles`). Secondly, we specify various types of role, since we need to distinguish between the agents/organisations and their social status. The case study differentiates between citizens and law enforcement officials as well as Internet Service Providers (ISPs) thus we have the roles `lawEnforcement` and `isp`. Thirdly, we distinguish between different data types (`content`, `metadata` and `personal`).

The case study is run against an observable event trace. We chose an observable event trace that shows the framework's context-sensitivity to abstract norm reasoning. This is by testing the use of electronic communications and ISP's fulfilment of metadata storage obligations in different social contexts. Namely, the context that an agent, Ada, has not consented and the context that she has. The trace is given below:

```
1  observed(exUseElectronicCommunication(ada, isp), 0).
2  observed(exStoreData(isp, ada, metadata), 1).
3  observed(exRequestData(charles, isp, ada), 2).
4  observed(exSignedConsentForm(ada, isp, personal), 3).
5  observed(exUseElectronicCommunication(ada, isp), 4).
6  observed(exStoreData(isp, ada, metadata), 5).
7  observed(exSignedConsentForm(ada, isp, voicerecording), 6).
8  observed(exSignedConsentForm(ada, isp, emailsubject), 7).
9  observed(exRequestData(ada, isp, bertrand), 8).
10 observed(exCloseInvestigation(charles, ada), 9).
11 observed(exChangeDataStorageLocation(isp, newzealand), 10).
12 observed(exUnauthoriseDataStorageLocation(bertrand, newzealand), 11).
13 observed(exUnauthoriseDataStorageLocation(bertrand, noneu), 12).
14 observed(exRequestData(charles, isp, bertrand), 13).
15 observed(exAuthoriseDataStorageLocation(bertrand, newzealand), 14).
16 observed(exOpenInvestigation(charles, bertrand), 15).
17 observed(exRequestData(charles, isp, bertrand), 16).
```

First the agent Ada uses electronic communications provided by the service provider ISP. Then, the service provider, ISP, stores Ada's commnications metadata. An agent, Charles, requests data from ISP concerning Ada. Ada signs a consent form for her data being stored (after the fact). Ada uses ISP's electronic communications again. Finally, ISP stores Ada's metadata again. Thus Ada's data is stored without her consent and then Ada's data is stored after she has given consent.

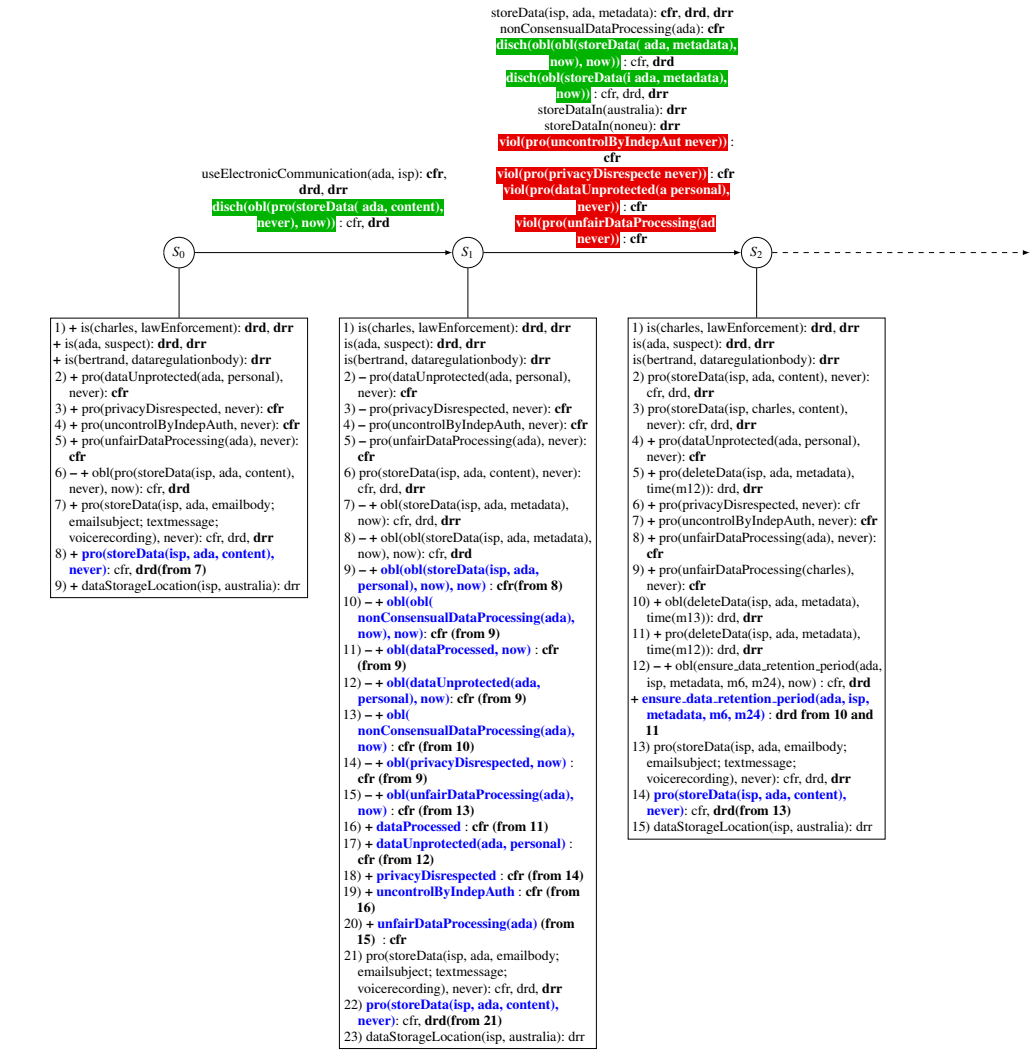

Fig. 7: Case study execution. The originating institutions for a fluent are in **bold**, '**+**' indicates an initiated fluent, '**–**' indicates a terminated fluent. non-inertial fluents are in **bold** denoting they are derived from other fluents according to the fluent derivation and deontological counts-as (norm abstraction) operations. Norm **discharge** and **violation** events are highlighted.

Then, Ada signs a consent form for voice recordings to be stored and subsequently email subjects. Ada requests Bertrand's data from ISP. Charles closes the investigation on Ada. ISP changes the data storage location to New Zealand. Bertrand unauthorises New Zealand as a storage location and subsequently all non-EU countries. Charles requests ISP to provide Bertrand's Data, then Charles opens an investigation about Bertrand and finally Charles requests Bertrand's data again. The main points about this latter half of the trace are that Ada consents to storing two types of content data, a ban is placed on storing data first in a specific non-EU country and then any country outside of the EU and finally Charles tries to obtain Bertrand's data before an investigation is opened and then afterwards.

The resulting multi-level governance institution model is depicted in Figure 7, for brevity edited to just containing those fluents that are relevant to the discharge and violation of norms or demonstrate semantic features. The model is described subsequently.

We first look at the interaction between the UK-DRR and the EU-DRD which governs the UK-DRR. Accordingly:

- State S_0 - Contains fluents stating the agent `charles` is playing the role of `lawEnforcement` `ada` is a suspect and `bertrand` is a `dataregulationbody` officer. A fluent states that `isp` stores data in Australia. The EU-DRD obliges that it is prohibited for `isp` to store the `content` of `ada`'s data. The UK-DRR does indeed prohibit `isp` from storing the content of ada's communications data. Thus, the obligation to prohibit storing content data in the EU-DRD is immediately discharged as denoted by the discharge event occurring in the transition to the next state. The transition to the next state also includes the event of `ada` using electronic communications provided by `isp`, due to the occurrence of the *exogenous* event in the timeline program stating the same.
- State S_1 - Includes new fluents. Firstly, the EU-DRD imposes an obligation on the UK-DRR to oblige `isp` to store `ada`'s communications' metadata. Secondly, the UK-DRR imposes an obligation for `isp` to store `ada`'s communications' metadata. The UK-DRR's first-order obligation to store metadata discharges the EU-DRD's second-order obligation to oblige an obligation to store metadata

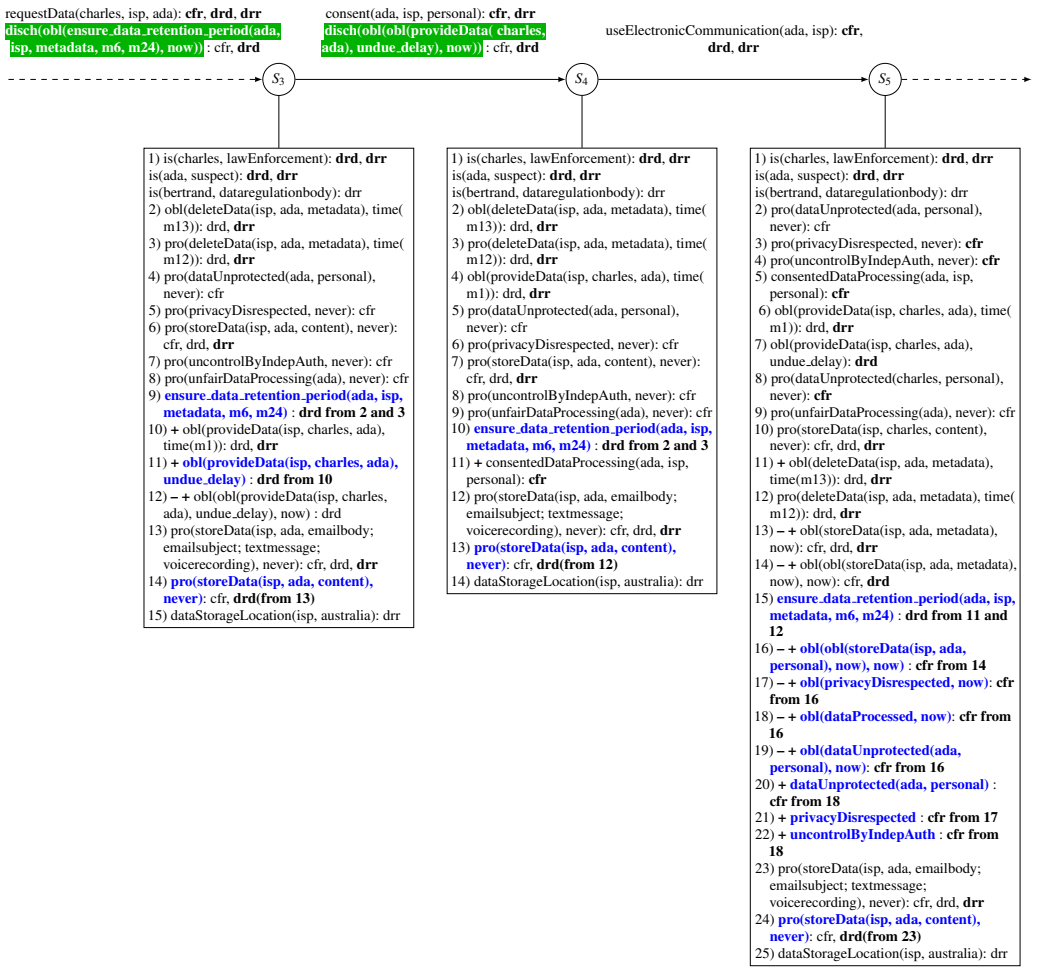

Fig. 7 (cont.)

fact that all types of communications content storage was prohibited on a case by case basis: voice recordings, email subject's, email bodies, etc.).

- Similarly state S_8 no longer contains a prohibition on storing Ada's email subject data.
- When transitioning to state S_9 Ada requests Bertrand's data from ISP, but no obligation is imposed on ISP in the `drr` because Ada is not a law enforcement officer.
- In state S_10 Charles closes the investigate on Ada, causing her to no longer be a suspect.
- State S_11 has the data storage location of ISP changed to New Zealand in the `drr`.
- State S_12 contains a prohibition on storing data in New Zealand after Bertrand, the data regulation body officer, unauthorises data storage in New Zealand. However, note that there is not consequently a prohibition on storing data in non-EU countries, because although New Zealand is a non-EU country there are also other non-EU countries where data can, permissibly, be stored.
- In state S_13 data is prohibited from being stored in non-EU countries after Bertrand explicitly places a blanket ban on storing data in any non-EU country.
- Although Charles requests Bertrand's data from ISP when transitioning to state S_14, there is no such obligation since Bertrand is not a suspect.

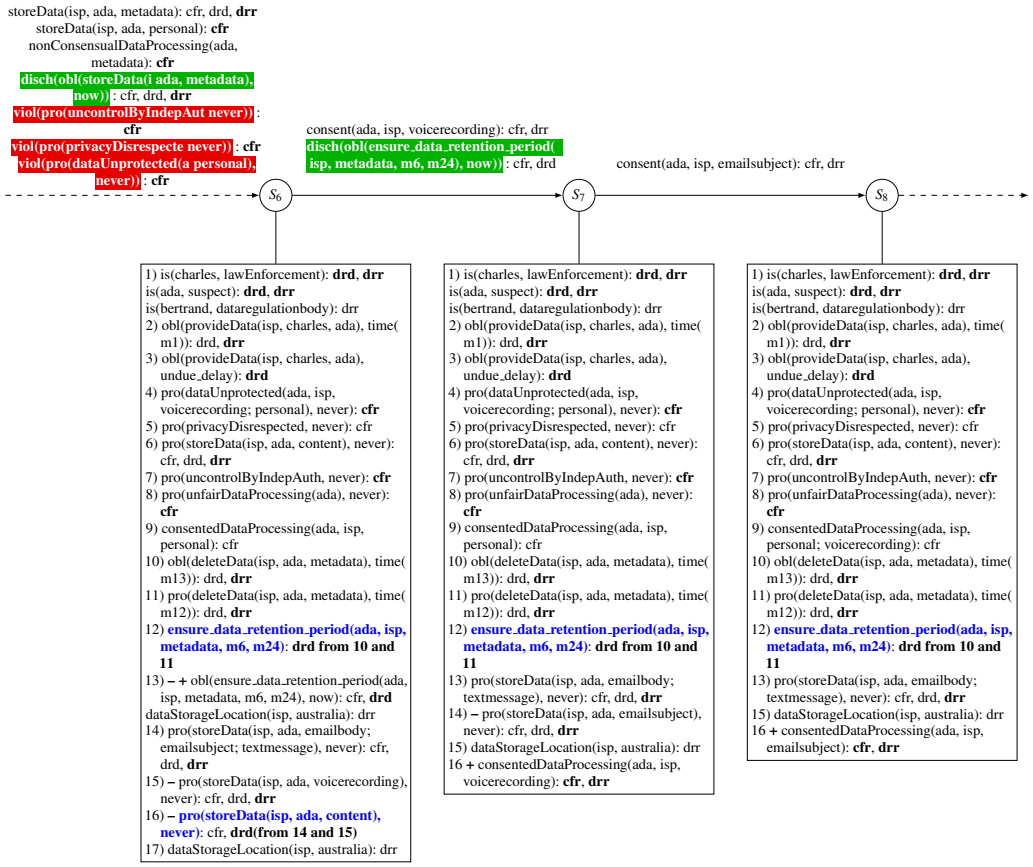

Fig. 7 (cont.)

- State $S_1 5$ lifts the ban on storing data in New Zealand, although there is still a ban on storing data in all non-EU countries.
- Finally, Charles opens an investigation into Bertrand, he becomes a suspect in state $S_1 6$. Then when Charles requests Bertrand's data from ISP there is an obligation to provide it within one month in state $S_1 7$ since Bertrand is now a suspect, and consequently there is an obligation to provide the data before any undue delay as interpreted by the `drd`.

In conclusion, for this trace of events the UK-DRR is compliant with the EU-DRD. All of the EU-DRD's normative fluents it imposes are discharged and none are violated. In comparison, the EU-DRD is *non-compliant* with the EU-CFR as we will see:

- State S_0 - the EU-CFR prohibits the EU-DRD's regulations from being uncontrolled by an independent authority. What this means is that data retention should be within the EU jurisdiction. Likewise, the EU-CFR also prohibits data from being unprotected (i.e., stored without anonymisation), privacy from being disrespected (i.e., personal data being stored) and data being processed unfairly (i.e., personal data being stored without an agent's consent).
- State S_1 - a number of the EU-CFR's prohibitions are violated:
  - **Violation of the CFR's prohibition on regulations not being controlled by an independent authority** (meaning, compliance with the EU-CFR's data protection rights must be observable by an independent authority, such as by ensuring data is retained within the EU). The EU-DRD

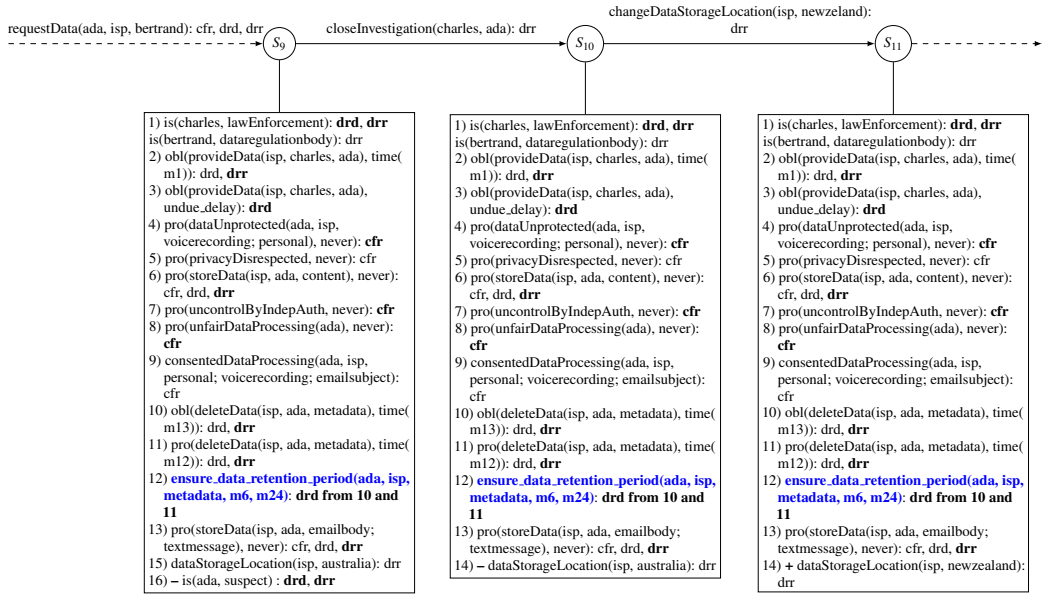

Fig. 7 (cont.)

obliges the UK-DRR to oblige `ada`'s communications' metadata is stored. According to the EU-CFR obliging storing data (of any type) counts-as data being processed, hence an obligation to oblige storing metadata is abstracted to an obligation to process data, which is abstracted further to processing data. The EU-CFR views processing data without a prohibition on it being stored outside of the EU counting-as regulations not being controlled by an independent authority. Hence, the prohibition on regulations being uncontrolled by an independent authority is violated.

– **Violation of the CFR's prohibition on unfair data processing**. The EU-CFR interprets storing metadata as storing personal data, thus it determines that there is an abstract obligation to oblige personal data is stored. In the EU-CFR, storing personal data in the context that an agent has not consented counts-as non-consensual data processing (`nonConsensualDataProcessing(ada)`). Thus the EU-CFR determines that there is an obligation to oblige non-consensual data processing of `ada`'s data. According the EU-CFR an obligation to store data non-consensually counts-as unfair data processing, hence an obligation to oblige non-consensual data processing is abstracted to an obligation to process data unfairly. An obligation to process data unfairly in turn, counts-as unfair data processing (i.e., from the perspective of the EU-CFR it does not matter if data is actually processed unfairly or just obliged, both are unfair data processing). This causes the EU-CFR's prohibition on processing data unfairly to be violated.
– **Violation of the CFR's prohibition on disrespecting privacy**. The obligation to oblige storing metadata imposed by the EU-DRD is abstracted to an obligation to oblige storing personal data. In the EU-CFR obliging storing personal data counts-as the non-inertial fluent for privacy to be disrespected. Hence, obliging an obligation to store personal data is further abstracted to obliging privacy is disrespected which also counts-as simply disrespecting privacy. Hence the EU-CFR's prohibition on disrespecting privacy is violated.
– **Violation of the CFR's prohibition on data being unprotected**. The obliges an obligation for Ada's metadata to be stored (according to the an obligation to oblige personal data to be stored) even in the context that it is not anonymised. The EU-CFR views an obligation to oblige storing personal data as being the same thing as processing data, which in the context that the data is

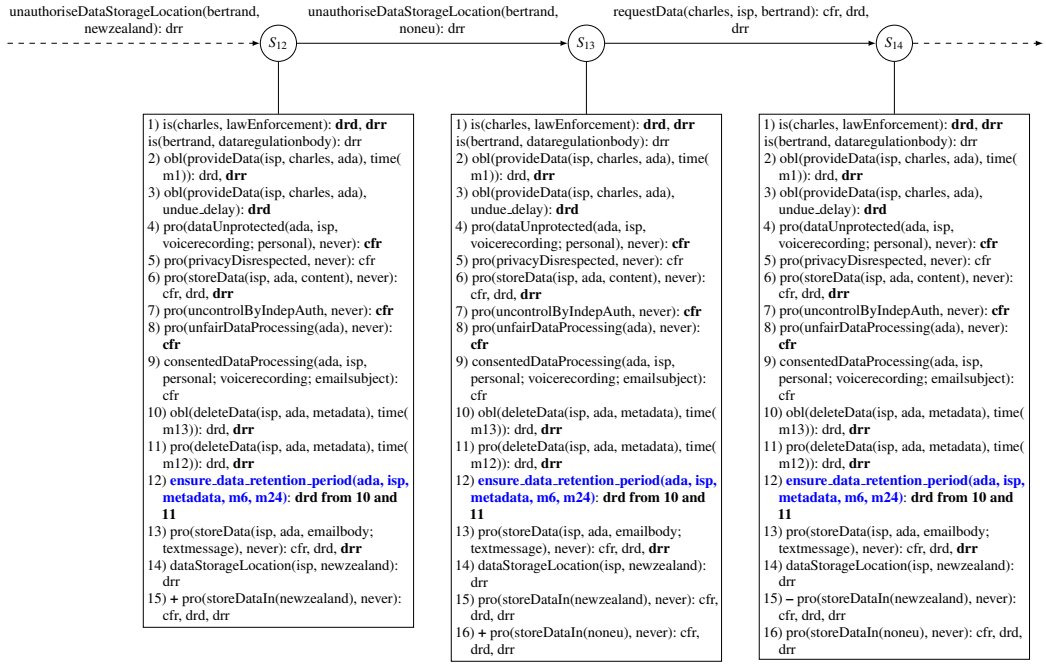

Fig. 7 (cont.)

not anonymised is abstractly the same thing as data being unprotected. Hence, the EU-CFR's prohibition on data being unprotected is violated.

Each violated prohibition in the EU-CFR is initiated in the next state.

- States S_2 and S_3 contain nothing of interest from the perspective of the EU-CFR. In the transition to state S_4 Ada consents to her personal data being stored.
- State S_4 contains a fluent stating Ada has consented to her personal data being stored.
- State S_5 also contains prohibitions in the EU-CFR which are violated by the EU-DRD, as in state S_1, with one difference:
  - **The CFR's prohibition on data being processed unfairly is *not* violated**. The EU-DRD, from the perspective of the EU-CFR, obliges an obligation to store personal data. However, since Ada has consented the obligation to oblige personal data being stored is not abstracted to an obligation to oblige non-consensual data processing and not subsequently abstracted to 'unfair data processing'. Hence, in state S_5, unlike in state S_2 the EU-CFR's prohibition on unfair data processing is not violated since the context is different (Ada has consented to her data being stored). Meanwhile, the rest of the EU-CFR's prohibitions are violated (for the second time).
- Subsequent states are less interesting to the `cfr`, however it is important to note that in the final state S_17 there is no violation of the EU-CFR's prohibition on regulations not being controlled by an independent authority (within the EU), since there is now a prohibition on storing data in non-EU countries. Hence, the context change caused by Bertrand prohibiting data storage outside of the EU results in different compliance effects.

From this case study we can see the UK-DRR is compliant with the EU-DRD (i.e., the UK's legislation correctly implements the directive). On the other hand, the EU's data retention directive is non-compliant

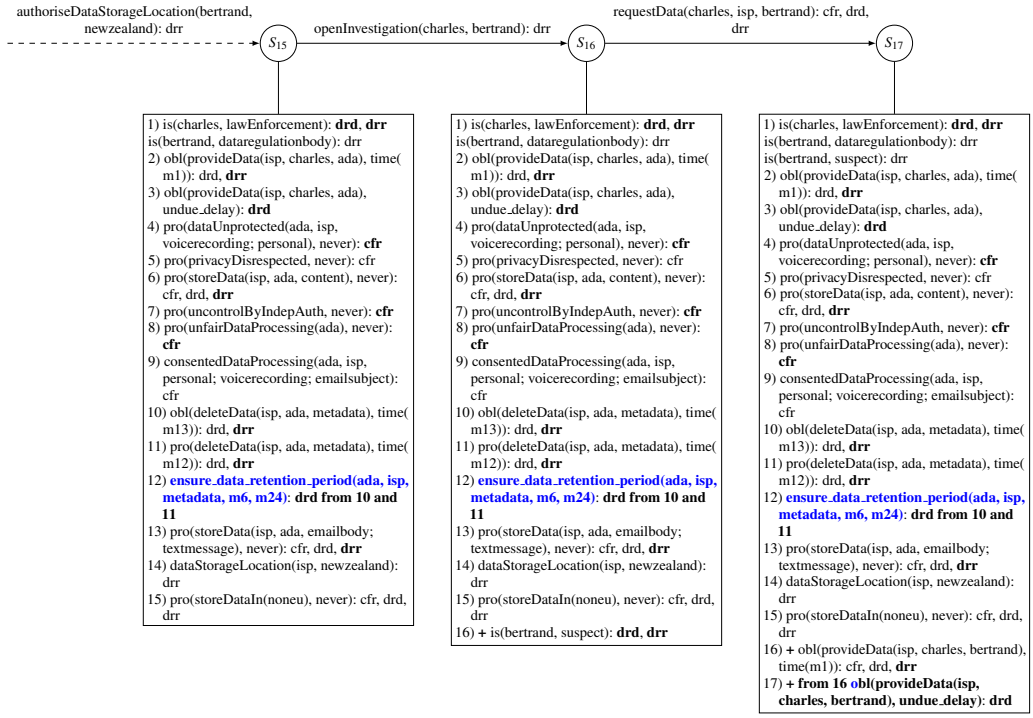

Fig. 7 (cont.)

with the EU-CFR. In particular, the EU-DRD was found to be non-compliant in a particular social context with particular prohibitions issued by the EU-CFR. In different contexts the same prohibitions might not be violated. As we saw in the context that Ada had consented to her personal data being processed, the directive did not the violate the prohibition on unfair data processing. This is because the directive's normative fluents were not interpreted by the charter as more abstractly counting-as unfair data processing. Hence, whether there is compliance depends on the context, which determines the abstract meaning of normative fluents.

5.5 Computational Framework Soundness and Completeness

We now demonstrate that the computational framework provides an executable implementation of the formal framework. This is with theorems stating the computational framework is sound and complete with respect to the formal framework (proofs are provided in Appendix C). We begin by packaging, for convenience, the AnsProlog programs of the computational framework, given in Appendix B, into a single AnsProlog program $\Pi^{\mathcal{ML}(k)}$.

Definition 18 Multi-level Governance AnsProlog Program Let $\mathcal{ML} = \langle \mathcal{T}, R \rangle$ be a multi-level governance institution. Let *ctr* be a composite trace for $\mathcal{ML}$ of length n. Let Π^{insts} and Π^{abstr} be the institutional and deontic abstraction programs obtained for $\mathcal{ML}$. Let, $\Pi^{trace(n)}$ be the trace program obtained for *ctr* and let $\Pi^{base(n)}$ be a multi-level governance base program. A multi-level governance institution AnsProlog program for $\mathcal{ML}$ and a composite trace *ctr* is:

$$\Pi^{\mathcal{ML}(n)} = \Pi^{base(n)} \cup \Pi^{trace(n)} \cup \Pi^{abstr} \cup \Pi^{insts}$$

We now give the soundness property for the deontic abstraction representation in AnsProlog with respect to the formal definition of deontological counts-as. In doing so, we demonstrate that we have provided a transformation that flattens the deontological counts-as function described in the formal framework to an executable set of AnsProlog rules. The property states that a state in the answer-set for a multi-level governance answer-set program is equivalent to the same state in the formal model for the formal framework with the deontological counts-as function $\overline{DC}^i$ applied.

Lemma 1 *Let $\mathcal{ML} = \langle \mathcal{T}, R \rangle$ be a multi-level governance institution s.t. $\mathcal{T} = \langle \mathcal{I}^1, ..., \mathcal{I}^n \rangle$, and ctr be a composite trace of length k. Let $\forall i \in [1,n]$ In^i be a unique label for $\mathcal{I}^i$. Let $\Pi^{\mathcal{ML}(k)}$ be the multi-level governance AnsProlog program for $\mathcal{ML}$ and ctr. Let M_P be an answer-set for the program $P^* = ground(\Pi^{\mathcal{ML}(k)})$. Given a set S^i_j such that*

$$\forall i \in [1,m], \forall j \in [k] : M_P \models \texttt{holdsat}(f, In^i, j) \Rightarrow f \in S^i_j$$

then $S^i_j = \overline{DC}^i(S^i_j)$.

Proof. See Appendix C.1. □

The next property we are interested in is soundness for the translation to an AnsProlog program as a whole. Specifically, the property states any answer-set for a multi-level governance AnsProlog program for a given trace of events corresponds to a multi-level governance institution model in the formal framework for the same trace of events.

Theorem 1 *(Soundness) Given a multi-level governance institution $\mathcal{ML} = \langle \mathcal{T}, R \rangle$ s.t. $\mathcal{T} = \langle \mathcal{I}^1, ..., \mathcal{I}^n \rangle$. Let $ctr = \langle e_0, ..., e_k \rangle$ be a composite trace for $\mathcal{ML}$. Let $\Pi^{\mathcal{ML}(k)}$ be the multi-level governance AnsProlog program for $\mathcal{ML}$ and ctr. Let $\forall i \in [1,n] : str^i = \langle se^i_0, ..., se^i_k \rangle$ be a synchronised trace for $\mathcal{I}^i$ w.r.t. ctr. Let M_P be an answer-set for the program $P^* = ground(\Pi^{\mathcal{ML}(k)})$. Then $M = \langle M^1, ..., M^n \rangle$ with $\forall i \in [n]$: $M^i = \langle S^i, E^i \rangle, S^i = \langle S^i_0, ..., S^i_{k+1} \rangle, E^i = \langle E^i_0, ..., E^i_k \rangle$ such that:*

$$\forall h \in [1,n], \forall j \in [k], : M_P \models \texttt{holdsat}(f, In^h, j) \Rightarrow f \in S^h_j \tag{T1.1}$$

$$\forall h \in [1,n], \forall j \in [k], \forall e \neq \texttt{null} : M_P \models \texttt{occurred}(e, In^h, j) \Rightarrow e \in E^h_j \tag{T1.2}$$

$$\forall h \in [1,n], \forall j \in [k] : M_P \models \texttt{occurred}(\texttt{null}, In^h, j) \Rightarrow e_{null} \in E^h_j \tag{T1.3}$$

is the model of $\mathcal{ML}$ w.r.t. ctr.

Proof. See Appendix C.2. □

The next property we are interested in is completeness. This states that for any model of a multi-level governance institution in the formal framework, for a trace of events, the multi-level governance AnsProlog program produces a corresponding answer-set for the same trace of events.

Theorem 2 *(Completeness) Given a multi-level governance institution $\mathcal{ML} = \langle \mathcal{T}, R \rangle$ s.t. $\mathcal{T} = \langle \mathcal{I}^1, ..., \mathcal{I}^n \rangle$. Let $ctr = \langle e_0, ..., e_k \rangle$ be a composite trace for $\mathcal{ML}$. Let $\forall i \in [1,n] : str^i = \langle str^i_0, ..., str^i_k \rangle$ be a synchronised trace for $\mathcal{I}^i$ w.r.t. ctr. Let $M = \langle M^1, ..., M^n \rangle$ be the multi-level governance institution model $\mathcal{ML}$ w.r.t. ctr where $\forall i \in [1,n] : M^i = \langle S^i, E^i \rangle, S^i = \langle S^i_0, ..., S^i_{k+1} \rangle, E^i = \langle E^i_0, ..., E^i_k \rangle$. Let $\Pi^{\mathcal{ML}(k)}$ be the multi-level*

structure AnsProlog program for $\mathcal{ML}$ and a composite trace ctr. Let M_P be the set of atoms:

$$\forall i \in [1,n], \forall j \in [k+1] : S^i_j \models f \Rightarrow M_P \models \texttt{holdsat}(f, In^i, j) \tag{T2.1}$$

$$\forall i \in [1,n], \forall j \in [k] : e \in E^i_j \Rightarrow M_P \models \texttt{occurred}(e, In^i, j) \tag{T2.2}$$

$$\forall i \in [1,n], \forall j \in [1,k] : f \in (S_j \backslash S_{j-1}) \cap \mathcal{F}^i_{inert} \Rightarrow M_P \models \texttt{initiated}(f, In^i, j-1) \tag{T2.3}$$

$$\forall i \in [1,n], \forall j \in [1,k] : f \in (S^i_j \backslash S^i_{j+1}) \cap \mathcal{F}^i_{inert} \Rightarrow M_P \models \texttt{terminated}(f, In^i, j+1) \tag{T2.4}$$

$$\forall i \in [1,n], \forall j \in [k] : e = ctr_j \Rightarrow M_P \models \texttt{observed}(e, In^i, j),$$
$$M_P \models \texttt{observed}(e, j),$$
$$M_P \models \texttt{obs}(j) \tag{T2.5}$$

$$\forall i \in [1,n], \forall j \in [k] : e = str^j_j \neq e_{null} \Rightarrow M_P \models \texttt{occurred}(e, In^i, j) \tag{T2.6}$$

$$\forall i \in [1,n], \forall j \in [k] : e_{null} = str^j_j \Rightarrow M_P \models \texttt{occurred}(\texttt{null}, In^i, j) \tag{T2.7}$$

$$\forall i \in [1,n], \forall e \in \mathcal{E}^i_{obs} : M_P \models \texttt{evtype}(\texttt{e}, In^i, \texttt{ex}) \tag{T2.8}$$

$$\forall i \in [1,n], \forall e \in \mathcal{E}^i_{inst} : M_P \models \texttt{evtype}(\texttt{e}, In^i, \texttt{inst}) \tag{T2.9}$$

$$\forall i \in [1,n], \forall f \in \mathcal{F}^i_{inert} : M_P \models \texttt{ifluent}(\texttt{f}, In^i) \tag{T2.10}$$

$$\forall i \in [1,n], \forall f \in \mathcal{F}^i_{ninert} : M_P \models \texttt{nifluent}(\texttt{f}, In^i) \tag{T2.11}$$

$$\forall i \in [1,n] : M_P \models \texttt{inst}(In^i) \tag{T2.12}$$

$$\forall i \in [k] : M_P \models \texttt{instant}(i) \tag{T2.13}$$

$$M_P \models \texttt{start}(0) \tag{T2.14}$$

$$\forall i, j \in [k] : j = i+1 \Rightarrow M_P \models \texttt{next}(i, j) \tag{T2.15}$$

$$M_P \models \texttt{final}(k) \tag{T2.16}$$

Then, M_P is an answer set of $P^* = ground(\Pi^{\mathcal{ML}(k)})$.

Proof. See Appendix C.3. □

This concludes the demonstration of the soundness and completeness of the formal and computational frameworks, with respect to each other.

5.6 Computational Framework Complexity

The question remains over the computational framework's complexity. There are three concerns surrounding an ASP program's complexity relevant to our case. Firstly, the program grounding complexity, which we measure as the worst-case growth in size of a ground program for a given input of institutions in a multi-level governance relationship represented in the formal framework. The growth in size as a function of institutions represented in the formal framework also accounts for any growth due to performing the transformation from the formal framework's institutional representation to an ASP program. Secondly, the answer-set computation complexity, which we measure in terms of how many literals need to be tested for inclusion in an answer-set as a function of the institutions in a multi-level governance relationship represented in the formal framework. Thirdly, the number of answer-sets to compute. Assuming the institutions are self-consistent then there will be at most one answer-set if a full event trace is provided as input. If a full event trace is not provided but the institutions are self-consistent, then for a given number of events $|\mathcal{E}|$ and number of undefined events in the trace m there is a combinatorial explosion of answer-sets $|\mathcal{E}|^m$. If the institutions are not self-consistent then there are potentially zero, one or more formal models and as a corollary of the soundness and completeness theorems the same number of answer-sets. However,

as far as we are aware there is no feasible way to give a general analysis of the number of answer-sets for the inconsistent institution design case, since they are dependent on the resulting ASP program's structure. Hence, we refer the interested reader to dynamic programming algorithms for the tricky problem of counting answer-sets *a-priori* [24]. Consequently, we focus on complexity in terms of program size and computing literal inclusion in an answer-set which can be determined together (i.e., grounding complexity is given by the growth from the input to the resulting program size, and the resulting program size gives a worst-case for computing one answer-set).

We give the ground program size for a multi-level governance institution $\mathcal{ML}$ as a function of, for each institution $\mathcal{I}^i$: the number of events and fluents, and the sum of rules and their sizes (i.e., the size of each rule is its context condition size plus two for the input and output event/fluent). The number of events is denoted as $|\mathcal{E}^i|$ and non-normative domain and empowerment fluents respectively as $|\mathcal{F}^i_{dom}|$ and $|\mathcal{F}^i_{pow}|$. It is important to note that higher-order norms have a more detrimental effect on complexity than first-order norms. Hence, we also delineate between the number of normative fluents of a particular order such that $\mathcal{F}^i_{norm(a:d)}$ denotes the set of normative fluents, for $\mathcal{I}^i$, that have the order of complexity (nesting) a for the aim and d for the deadline. For example, $\mathcal{F}^i_{norm(1:1)}$ contains all of the first-order normative fluents, $\mathcal{F}^i_{norm(2:1)}$ contains all of the second-order normative fluents where a first-order normative fluent is the aim and an event or domain fluent is the deadline, and so on. For rules, we mean specifically the number and size of: state consequence rules for fluent initiation and termination ($|\mathcal{C}^i|$), event generation rules ($|\mathcal{G}^i|$) captured by the event generation function, and fluent derivation rules ($|\mathcal{D}^i|$) captured by the fluent derivation function.

An upper-bound on the size of the ground AnsProlog program (i.e., the number of ground facts, and the sum of ground rules and their sizes) for a multi-level governance institution $\mathcal{ML}$ and composite trace of length k, denoted as $|ground(\Pi^{\mathcal{ML}(\|)})|$, is given below.

$$
\begin{aligned}
|ground(\Pi^{\mathcal{ML}(k)})| \leq\ & \Big(\sum_{i}^{[1,n]} 1 + |\mathcal{E}^i| + |\mathcal{F}^i_{dom}| + |\mathcal{F}^i_{pow}|\Big) + \\
& \Big(\sum_{i}^{[1,n]}\sum_{j}^{[0,k]} |\mathcal{C}^i| + |\mathcal{G}^i| + |\mathcal{D}^i|\Big) + \\
& \Big(\sum_{i}^{[1,n]}\sum_{j}^{[0,k]}\sum_{a,d}^{\mathbb{N}} 2^{a+d} \times |\mathcal{F}_{norm(a:d)}| \times 2\Big) + \\
& \Big(\sum_{i}^{[1,n]}\sum_{j}^{[0,k]}\sum_{a,d}^{\mathbb{N}} |\mathcal{G}^i| \times |\mathcal{F}_{norm(a:d)}| + |\mathcal{D}^i| \times |\mathcal{F}_{norm(a:d)}|\Big)
\end{aligned}
$$

The first line is the number of facts in the ground ASP program representing each institution's name, events and non-normative fluents. The second line is the size and number of each institution's rules for each point in time in the ground ASP program. The third line is the number of rules for computing equivalences between norms and their size (one head and one body literal). The fourth line represents the worst-case number and size of rules that abstract normative fluents (note that it assumes the set of normative fluents only contains obligations, since there are many more rules for abstracting obligations than prohibitions and thus it represents the worst-case complexity). In summary the biggest impact on program size is the complexity order of normative fluents, which require rules capturing normative fluent equivalences and thus causing the ground program size to grow exponentially.

6 Related Work

This paper builds on our previous work for reasoning about what we called multi-tier institutions in [49, 52] (which in turn built on preliminary work by King et al. [50]). In our prior work higher-

tier institutions govern lower-tier institutions, which we extended in this paper to representing and reasoning about multi-level governance. In turn, our work is influenced by the InstAL framework [13] for institutional reasoning. Our framework bears the most similarity to other computational-focussed institutional reasoning frameworks, hierarchical governance and higher-order normative reasoning, and work on norm abstraction. We compare our work with each of these individual aspects in the literature. However, we find no work that combines higher-order normative reasoning and abstraction, as required for multi-level governance reasoning, or provides an obvious way to combine the two.

6.1 Institutional Reasoning and Verification

There have been many different approaches proposed to reason about institutions, normative systems and organisations which we split into three broad types. Firstly, those proposing a high-level institution specification language (e.g., [61, 62, 19]) for institution designers to precisely specify an institution's software implementation. Secondly, those proposing or studying formal logics of norms and other institutional rules (e.g., [10, 16, 63, 39, 41, 75]). Thirdly, those contributing frameworks for formally representing and reasoning about institutions and normative systems, with an aim for practical implementations using an algorithmic or logic-programming based approach (e.g., [12, 13, 46, 54, 35, 36]). Our work most closely relates to the latter practical frameworks, which we discuss in more detail.

The most closely related framework, on which we build, is the Institutional Action Language (InstAL) first proposed by Cliffe et al. [13, 12]. Li et al. have made developments on InstAL for detecting conflicts between norms [56], in particular in interacting institutions [55] and cooperating institutions [54]. In the work of Li et al. institutions are linked with special *bridge* institutions such that events occurring in one institution can cause events to occur in another institution and likewise for fluents being initiated or terminated. Such bridge institutions have a similar role to our links between different levelled institutions, but can be flexibly defined to ensure specific fluents are initiated in one institution by another. In our framework, we do not require such flexibility since we only need to capture multi-level governance relationships where regulatory effects are passed between institutions.

Further developments on InstAL were realised by Pieters et al. [65, 66] for reasoning about institutions as a means to police and enforce security policies. In their work, Pieters et al. [65, 66] extend InstAL with rules for non-inertial fluents that (in our own words) state "when context C holds then so does fluent B". These bear similarity to our fluent derivation rules of the form "fluent A counts-as (derives) fluent B in context C". But, in our case we view fluent derivation rules as firstly ascribing a special meaning to a concrete fluent 'A' (hence they have a different form) and secondly serving as a basis for abstracting normative fluents. Using a variant of Searle's money example [70], in our framework a counts-as rule might state "possessing a piece of paper marked with a Euro symbol counts-as (derives) possessing money in the context of the Eurozone", hence if it holds that such a piece of paper is possessed in the context of the Eurozone, then it also holds that money is possessed. In the framework of Pieters et al. [65, 66] the same rule would be "in the context of possessing a piece of paper marked with a Euro symbol in the Eurozone then money is also possessed". In the former case it is clearly possessing the piece of paper that has the status symbol of possessing money, hence in the context of the Eurozone we can derive from an obligation to possess that piece of paper another obligation to possess money. In the latter case, it is not clear what, exactly, possessing money is. In this case, we cannot clearly derive that an obligation to possess the right piece of paper counts-as an obligation to possess money, because it is not explicit that the paper constitutes money. Hence in comparison to [65, 66] our rules for ascribing non-inertial fluents are counts-as rules in the usual sense for the reason that it enables us to derive relations between concrete and abstract normative fluents.

Finally, our work in this paper also extends our previous work, which was loosely based on InstAL [49, 52], for reasoning about multi-tier institutions and higher-order norms. The main differences between all of these developments and this paper is that we have extended InstAL for representation and reasoning about multi-level governance. In more detail, there are differences in reasoning about permissive societies

	InstAL [13, 12]	Li et al. [55, 56, 54]	Pieters et al. [65, 66]	King et al. [49, 52]	This paper
Individual Institutions	✓	✓	✓	✓	✓
Empowerment	✓	✓	✓		✓
Bridged vs. **L**inked Institutions		**B**			**L**
Non-**I**nertial vs. Fluent **D**erivation rules			**NI**		**D**
Permissive Society				✓	✓
Instantaneous and Indefinite Norms					✓
Higher-order Normative Reasoning				✓	✓
Norm Abstraction					✓

Table 7: Comparison between closely related developments on InstAL.

(where anything not prohibited is permitted), instantaneous and indefinite norms, bridged versus linked institutions, non-inertial fluent rules versus fluent derivation rules, and our main focus in this paper: combining higher-order normative reasoning and norm abstraction. We summarise all of these differences in Table 7.

Work that addresses reasoning about artificial societies or events and their effects bears resemblance where similar techniques, such as the Event Calculus [53], are used. A series of papers by Artikis (et. al.) uses Event Calculus-based reasoning to capture MAS' normative dimension based on the events that occur and consequently fluents that hold, or to determine the events occurring in the MAS themselves. In [6, 7] Artikis et al. formerly use the Event Calculus and latterly the $\mathcal{C}+$ language in a similar fashion to our proposal. That is, in order to reason about the same core institutional concepts we adopt: deontic positions, empowerment and counts-as rules. Whilst we adopt a generic notion of empowerment that applies to events and can be applied to events that (presumably) denote agent actions, Artikis et al. offer an empowerment fluent that specifically applies to agents. In the latter case where Artikis et al. use $\mathcal{C}+$ as their foundational logic, the institutional language is richer in some ways compared to our proposal. For example, both defaults (e.g., that by default everyone is empowerment to make a payment) and constraints on performing actions can be expressed. In principle, the expressiveness of Artikis et al. could be incorporated into our proposal, where ours differs significantly in aims (institutions governing other institutional designs, where compliance is verified for supplied or generated event traces).

Social commitments (e.g., contracts, promises) have also been formalised [15, 43, 77] with 'lifecycle' elements not present in our notion of norms, such as the creation and deletion of the commitment/rule (e.g., through an utterance) which in turn imposes obligations in particular circumstances. Higher-order commitments are grammatical in some commitment-focused approaches (e.g., [43, 77]) but they do not coincide with our notion of higher-order norms, as we now explain. In our case, a higher-order norm represents a statement such as 'if event A occurs then it is obliged that the *outcome* of your rules does not oblige B in context C'. On the other hand, nested commitments represent statements such as 'you have promised to me that you will not create a *commitment rule* stating that when A occurs there is an obligation to do B in context C'. In the case of commitments, the nesting is really a promise to (not) make a certain commitment rule. Whilst in our case the nesting in a higher-order norm represents that there should (not) be certain obligations and prohibitions imposed from *any* normative rule, regardless of their form, in specified contexts. Moreover, in our case, the higher and first-order obligations and prohibitions may have more abstract meanings which need to be determined through interpretation. Consequently, commitments and nested commitments, come from a fundamentally different perspective and are not aligned with our formalisation of regulations that govern other regulations nor do they capture norm/commitment abstraction.

Another practical institutional reasoning approach is temporal defeasible deontic logic. Defeasible logic is a non-monotonic logic designed to be implemented in Prolog [4, 64]. There are three rule types in many defeasible logics, *strict* rules ($\rightarrow$) whose conclusion is true so long as the antecedent is true,

defeasible rules ($\Rightarrow$) whose conclusion is true when the antecedent is true unless the rule is rebutted or undercut by another rule, and *defeating* rules ($\rightsquigarrow$) whose conclusion is never true but if the antecedent is true rebuts or undercuts other rules that have a contradictory consequent or antecedent (respectively). A defeasible logic often comprises a proof procedure where rule conclusions are tested for whether they are true by first asserting them as an argument, then finding all counter-arguments by applying defeating rules, and then recursively counter-attacking all attacks with further arguments, terminating thanks to constraints on non-repeatability of arguments (e.g., [69]). Defeasible *temporal deontic logics* formalised by Governatori et al. [35, 36] extend defeasible logic with rule types and proof procedures for obligations and temporalised outcomes. In these proposals various legal concepts are formalised, including constitutive rules and norms. But as far as we know there have been no developments on these approaches towards norms governing norms and/or norm abstraction, such as for reasoning about compliance in multi-level governance.

6.2 Hierarchical Governance and Governing Regulations

There appears to be little literature on hierarchical governance and the regulation of regulations. In [61] López y López and Luck propose a framework for reasoning about norms governing agents, created from a top-down governance perspective. Their framework, based on the $\mathbb{Z}$ specification language, gives a precise specification language of a normative system/institution. In comparison, our framework comprises a specification language and operationalisation (semantics) for institutions operating in multi-level governance. Like our framework, theirs offers similar expressivity with temporal norms, rewards, punishments, etcetera. In particular López y Lopez and Luck formalize what they call *legislative norms*, which are special norms governing the act of norm changes in the sense of making it possible to amend norms. This still presents a substantial difference to the method of hierarchical governance and regulation governing regulations that we propose, since we use higher-order norms that govern the *outcome* of other norms from which (non-)compliance is determined (typically pre-runtime). López y López and Luck's legislative norms on the other hand govern the changes to the norms (rules) themselves.

Boella and van der Torre [10] offer a conceptual formalisation of *hierarchical* normative systems in the Input/Output Logic (a logic aimed at studying conditional norms [63]). In particular, they focus on the role of permissions in hierarchical normative systems, where permissions are issued by higher authorities (e.g., existing in higher level institutions) and act to *derogate* (except) obligations to the contrary (prohibitions) issued by lower level authorities. Their work is similar to ours with respect to governance hierarchies, but at the same time quite different in that they are not concerned with the regulation of regulations and non-compliant regulatory outcomes or a corresponding computational framework.

Lopes Cardoso and Oliveira [57, 58] focus on norms applied to different levels of what they call context hierarchies. In their work institutions share concepts with our own formalisation, comprising descriptive rules that create institutional facts and norms that create deontic positions. What differentiates their work from our own is the idea that a norm can defeat another if it is applied to a lower-level context and there is a normative conflict. In this sense, the more specific norms (i.e., applied to a narrower context) are preferred in a similar vein to the *lex specialis* principle. In this way, agents are able to interact according to a super-contract that applies in the top-most context and through this super-contract inherit new norms in new sub-contractual relationships applied to more specific lower-level context as deemed appropriate for a given social interaction. In contrast with our work, a semantics of abstraction is not defined and instead the focus is on defeasibility based on context application.

García-Camino et al. [29] also investigate hierarchical normative structures. In this case, where there are hierarchical relations between activities and their constituent sub-activities. For example, 'trading' is an activity that has the sub-activity 'auction'. In this hierarchical setting, activities are governed by norms and so are their sub-activities. The central problem García-Camino et al. investigate is not norms at higher-levels governing those at lower-levels, but instead the possibility for conflicts to occur between

norms in activities that are propagated down to their sub-activities. García-Camino et al. propose a conflict resolution mechanism to address this issue.

6.3 Abstracting Norms

There has already been a reduction of Standard Deontic Logic [76] to a logic of counts-as conditionals representing evaluative norms [2], colloquially known as 'Anderson's reduction' (as studied in [42, 38]). For example, 'B counts-as a violation in a context C'. Following this idea, Aldewereld et al. [1] propose implemented reasoning for concretising abstract norms. This is done by representing abstract norms as counts-as statements such as 'B counts-as a violation in a context C' and so B is forbidden in C. Then, making use of the fact that more concrete concepts count-as more abstract concepts (e.g., 'A counts-as B in context C'). Finally, applying transitivity to concretise abstract norms (e.g., 'A counts-as a violation in context C', since A counts-as B and B counts-as a violation). Alderwereld et al. provide a computational approach to the normative reasoning with a rule-based computational language. The same warning against this approach for multi-level governance that we make in the background on the governance concepts (Section 2), applies to what differentiates it from our work. Specifically, that in our approach we can represent higher-order norms simpler, such as 'it is prohibited to oblige a user's metdata to be stored in the context that they have not consented". That is, when compared to the more complicated representation required using Anderson's reduction, such as "(storing metadata counts-as being good in a context C) counts-as being bad if context C is somehow compatible with the user not consenting". Specifically, by ignoring deontic modalities it is difficult to describe and reason about higher-order norms. Although concretisation of norms is possible, higher-order normative reasoning (regulation governing regulations) is not and neither is the abstraction of higher-order norms.

A description-logic based formalisation for reasoning about abstract institutional concepts is also proposed by Grossi et al. [40]. Unlike our work, Grossi et al. do not propose abstraction of norms themselves (in fact, they propose concretising concepts), since normative reasoning is not considered. Rather, they offer guidance on how normative reasoning can be incorporated, either by the reduction of norms to counts-as, which like us they acknowledge does neither supports nesting of deontic modalities nor higher-order norms. They also offer an alternative path to normative reasoning that involves the use of explicit deontic modalities (the same approach we adopt). However, this part of their proposal is not formalised. Furthermore, our work still differs in that we are interested in abstracting rather than concretising norms in a temporal-like setting.

In comparison, a series of papers by Fornara and Colombetti [26], Fornara [25] and Fornara, Okouya and Colombetti [27] combine the semantic-web focussed description logic OWL2DL with normative reasoning. In their proposal, obligations are about events with a time-indexed deadline. Time is not integrated within the underlying logic, rather it is reasoned about using an external process that adds facts to the knowledge-base (e.g., that an action has occurred, time has passed, etc.). Like our proposal and many others, the deadline of an obligation occurring before the aim triggers violations and potentially causes punishing obligations to be imposed. In comparison to the work of Grossi et al. they do explicitly look at representing and reasoning about norms in description logic but do not aim to reason about the relationship between concrete and abstract concepts or the concretisation/abstraction of norms. The same differences apply when compared to our own work with the additional difference that we do not restrict norms to being about events. Rather, in our proposal normative fluents can be higher-order and about events or other fluents.

Criado et al. [14] look at agent reasoning for fulfilling agent desires about abstract institutional concepts. Such desires may come about due to the presence of regulative norms (e.g., an obligation to be married), but their focus is on the concretisation of these abstract institutional concepts (e.g., if an agent wants to get married, what are the brute facts that need to be realised?). In relation to our work, Criado et al. also view counts-as, as providing interpretive rules in which abstract institutional concepts can be reasoned about. However, they do not explicitly look at how to transform abstract norms into concrete

ones, or as we do concrete (higher-order) norms into abstract ones to check compliance. Rather, their focus is on the interpretation of the abstract *concepts* in order to fulfil agents' desires.

Related to our abstraction of temporal norms Lopes Cardoso and Oliveira propose reasoning and monitoring for norms with *flexible* deadlines [59, 60]. In their proposal, the idea is that deadlines in contractual norms are not always strict, but instead one party can violate a deadline (e.g., to deliver goods) whilst the counter-party may be okay to waive the violation if the obligation is discharged within a reasonable time after its violation (e.g., if the goods being delivered are more important than the sanction for the misdemeanour). In contrast, our proposed semantics re-interprets temporal norms by abstracting the constituent aim and deadline in the contexts it is applied. To some extent, we investigated temporal conditions with a flexible meaning, for example where the data-retention regulations required data be provided on request within a specific time limit and this constituted, according to the ECJ's interpretation, an obligation to provide data before any *undue delay*. Although we look into the idea of ambiguity surrounding temporal conditions, Lopes Cardoso and Oliveira capture deadlines that can be defeated under defeasible reasoning at run-time.

To summarise, published work proposing ways to reason about abstract and concrete norms or using techniques that can be extended to do so is quite different from that which we describe here. Whilst some work does look at the concretisation of abstract norms, there is apparently no work that looks at the abstraction of concrete, potentially *higher-order*, norms. Furthermore, the aforementioned work that explicitly looks at concretisation is not in a temporal setting. In contrast, our proposal focuses on the temporal aspects where, as the institutional context evolves, so does the abstract meaning of concrete norms and thus their compliance with abstract norms at higher governance levels.

6.4 Legal Power and Counts-as Rules

As we discussed previously, the notion of power we adopt differs from that of Jones and Sergot's [47], where an action is empowered to be taken if it can be ascribed by constitutive rules. This counts-as based notion of empowerment has also been used to characterise rule change governance. Whilst in this paper we focussed on the governance of institution designs, the jurist Hart [44] conceptualised *secondary* legal rules that act to make legally possible the institutional action of rule change (e.g., through a majority vote or physically changing the rule book). Later, Biagoli [9] argued that secondary legal rules are a sub-class of Searle's counts-as rules.

Based on these developments, Boella and van der Torre [11] formalised the notion of constitutive rules that make it possible to legally change rules, which in themselves may also be legally changed. In their formalisation, Boella and van der Torre focus on legislating games where participants aim to modify rules for their own purposes in a static setting. Governatori et al. [34, 33] meanwhile adopt a kind of meta-rule that acts to introduce rule changes into a legal system, in a similar vein to Boella and van der Torre's formalisation of Hart's secondary rules, which legally empower rule change. In Governatori et al.'s work the focus is on formalising a temporal defeasible logic of rule change and different classes of rule change (e.g., annulment, abrogation, etc.). Later King et al. [51] and King [48, p.136–154] formalised the legality of rule change, using counts-as rules, in a temporal setting where the focus was on the legality of rule change being conditional on hypothetical effects (e.g., whether it would criminalise formerly innocent people) and whether it would cause a paradox (e.g., a self-modifying rule). These papers capture a kind of governance of institutional designers or legislators, compared to our formalisation of institutional design governance by higher-level institutions.

Artikis [5] also formalises a notion of the legal power to change rules (legally). In Artikis' approach the Event Calculus is used to specify social protocols comprising familiar institutional concepts to direct and guide agent interactions at the bottom-most level of the society. Adopting a hierarchical structure, agents can also dispute and change a protocol according to a meta-protocol specifying the social choice procedure that must be followed in order for the object-level protocol to be modified in a legally empowered way. In turn the meta-protocol is also legally modifiable by a meta-meta- protocol and so on. Similar to our

proposal, a governance hierarchy is employed, but unlike our work the focus is not on governing the outcomes of institutional rules (obliging/prohibiting obligations and prohibitions) nor on abstraction in multi-level governance. Instead, Artikis' proposal bears closer resemblance to the aforementioned work on the legal power to change rules.

7 Conclusions

In this paper we introduced a novel framework for determining compliance in multi-level governance. Our framework contributes a formal representation and semantics, giving a rigorous account of multi-level governance compliance independent of any particular implementation. We ensure our proposal is practical by complementing the formal framework with a corresponding computational framework. We adopt the usual notion of *counts-as* between concrete and abstract institutional facts. Based on the *counts-as* ontological rules, we semantically defined the abstraction of norms, both first-order and higher-order, with a semantics of *deontological counts-as*. In so doing, we proposed a novel semantics for assessing the different contexts in which norms can be applied and abstracting the normative effects of lower level institutions based on those contexts, where the abstract meaning evolves as the social context evolves. Our framework uses this abstracting mechanism to determine if concrete norms in lower-level institutions are non-compliant with more abstract higher-order norms in higher-level institutions. That is, we gave a semantics of compliance in multi-level governance.

We assessed our proposal along three fronts. Firstly, with a comprehensive case study based on three-levelled governance in EU law where abstraction and context-sensitivity are important in determining non-compliance. Secondly, by proving that the practical implementation in Answer-Set Programming, the *computational framework*, is indeed *sound* and *complete* with respect to the formal framework. We used the fact that the formal framework corresponds to the computational framework to implement the proposal by extending the InstAL compiler, thereby offering users a high-level language to specify institution designs under multi-levelled governance and the means to automatically determine institutional design compliance. Thirdly, we analysed the program complexity in terms of its size compared to its input institution specifications. That is, our framework provides both a rigorous formal foundation for multi-level governance representation, semantics and compliance checking, and the practical computational means to automatically determine compliance.

Our approach and the framework we proposed show potential for further development, refinement and wider application. Firstly, through investigation into the abstraction of temporal normative fluents to non-temporal normative fluents based on counts-as between temporal formulae. For example, an obligation to send communications' metadata before one month *counting-as* an obligation to send communications' metadata quickly. In part, such abstraction was not captured by our proposal due to the fact that temporal formula cannot be said to count-as another (non-)temporal formula (e.g., "A before D *counts-as* B in context C"). Consequently, we lacked an ontological rule on which to base a deontological counts-as between temporal and explicitly non-temporal normative fluents. We foresee that a move to a full temporal logic is necessary for this kind of abstraction. Further investigation is needed, since using a temporal logic raises questions such as at which point in time from "A before D *counts-as* B in context C" we can derive an institutional fact "B". Defining answers to such questions in a temporal logic would enable us to apply the intuitions of deontological counts-as we have set out here to a fully-temporal setting.

Another important future development is extending our semantics to support ontological alignment of institutions. In our proposal lower level institutions' regulatory effects were re-interpreted at the abstraction of higher level institutions. In one sense, this means that lower level institutions' regulations were *aligned* with the abstraction of higher level institutions. For example, obliging the storing of communications metadata in a lower level institution is abstracted to obliging the storing of personal data in a higher level institution. However, we assumed that the terms shared by lower and higher level institutions have the same meaning and are already ontologically aligned, by which we mean storing metadata in the lower level institution coincides with storing metadata in the higher level institution. Thus, if the lower level

obliges the storing of metadata, and the higher level views the storing of metadata as storing personal data, then from the higher level institution's perspective the lower level is obliging that personal data is stored. The assumption of shared terms being aligned between institutions should be relaxed with a correct formal treatment in order to compare regulations between institutions.

Another avenue for future work is extending the application of our theoretical work on multi-level governance in human societies, to artificial societies. We envisage this as an important operationalisation of two proposals. In what Pitt et al. [67] called polycentric governance, it is argued that in complex artificial MAS a single one-size-fits-all institution is inadequate, since different localised parts of the MAS may need different regulations. Therefore, separate lower level institutions should be designed, appealing to subsidiarity, inline with overarching institutions abstractly prescribing what regulations should be implemented. Similarly, a design methodology for institutions/organisations governing artificial MASs has been proposed in the OMNI framework [18]. Here, the design methodology is focussed on a regulation abstraction hierarchy where at the most abstract level statutes comprising values, objectives and contexts should be designed, followed by abstract norms implementing these statutes and then concrete norms implementing the abstract norms. In our framework we showed how constitutive rules provide the ontological basis for capturing links between concrete and abstract norms an appropriate ontology for an artificial society. Hence, we foresee our contributions supporting the design of governance for artificial and socio-technical systems according to the design principles of [18, 67], based on an appropriate ontology of constitutive rules for an artificial or socio-technical society.

Acknowledgements

We would like to thank the anonymous reviewers of JAAMAS for helping to improve the article. Thomas C. King would like to thank John R. Searle for the correspondence on substitution-of-identicals which illuminated some formerly implicit assumptions (now explicit) made in this paper. Thomas C. King was supported by the SHINE [6] project of TU Delft.

References

1. Huib Aldewereld, Sergio Álvarez-Napagao, Frank Dignum, and Javier Vázquez-Salceda. Making Norms Concrete. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 807–814, 2010. ISBN 978-0-9826571-1-9. doi:10.1145/1838206.1838314.
2. A. R. Anderson. A reduction of deontic logic to alethic modal logic. *Mind*, 67(265):100–103, 1958. ISSN 00264423. doi:10.1093/mind/LXVII.265.100.
3. Giulia Andrighetto, Guido Governatori, Pablo Noriega, and Leendert van der Torre. *Normative Multi-Agent Systems*, volume 4. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013. ISBN 9783939897514.
4. G Antoniou, D Billington, G Governatori, and M J Maher. Representation results for defeasible logic. *ACM Transactions on Computational Logic*, 2(2):255–287, 2001.
5. Alexander Artikis. Dynamic Protocols for Open Agent Systems. In *8th International Conference on Autonomous Agents and Multiagent Systems*, pages 97–104, 2009.
6. Alexander Artikis, Jeremy Pitt, and Marek Sergot. Animated specifications of computational societies. In *Proceedings of the first international joint conference on Autonomous Agents and Multiagent Systems*, pages 1053 – 1061, New York, New York, USA, 2002. ACM Press. ISBN 1581134800. doi:10.1145/545068.545070.

[6] `http://shine.tudelft.nl`

7. Alexander Artikis, Marek Sergot, and Jeremy Pitt. Specifying Electronic Societies with the Causal Calculator. In *Proceedings of the Workshop on Agent Oriented Software Engeneering III (AOSE)*, volume LNCS 2585, 2003.
8. Chitta Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, 2003. ISBN 9780511543357. doi:10.1017/CBO9780511543357.
9. Carlo Biagioli. Towards a legal rules functional micro-ontology. In *Proceedings of the 1st LegOnt Workshop on Legal Ontologies*, 1997.
10. Guido Boella and Leendert van der Torre. Permissions and obligations in hierarchical normative systems. In *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, pages 109–118, 2003. ISBN 1581137478.
11. Guido Boella and Leendert van der Torre. Regulative and Constitutive Norms in Normative Multiagent Systems. In *Proceedings of 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR'04)*, pages 255–265. AAAI Press, 2004.
12. Owen Cliffe. *Specifying and Analysing Institutions in Multi-Agent Systems Using Answer Set Programming*. PhD thesis, University of Bath, 2007.
13. Owen Cliffe, Marina De Vos, and Julian Padget. Answer Set Programming for Representing and Reasoning About Virtual Institutions. *Computational Logic in Multi-Agent Systems*, pages 60–79, 2007.
14. N Criado, E. Argente, P. Noriega, and V. Botti. Reasoning about constitutive norms in BDI agents. *Logic Journal of the IGPL*, 22(1):66–93, 2013. ISSN 1367-0751. doi:10.1093/jigpal/jzt035.
15. Mehdi Dastani, Leendert van der Torre, and Neil Yorke-Smith. Commitments and interaction norms in organisations. *Autonomous Agents and Multi-Agent Systems*, 31(2):207—-249, 2017. ISSN 1573-7454. doi:10.1007/s10458-015-9321-5.
16. F Dignum. Abstract Norms and Electronic Institutions. In *International Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA'02)*, pages 93 – 104, 2002.
17. F Dignum, H Weigand, and E Verharen. Meeting the deadline: on the formal specication of temporal deontic constraints. *International Symposium on Methodologies for Intelligent Systems*, pages 243–252, 1996.
18. Virginia Dignum, Javier Vázquez-Salceda, and Frank Dignum. OMNI: Introducing Social Structure , Norms and Ontologies into Agent Organizations. In *Programming Multi-Agent Systems*, pages 181–198. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-24559-9.
19. Mark D'Inverno, Michael Luck, Pablo Noriega, Juan a. Rodriguez-Aguilar, and Carles Sierra. Communicating Open Systems. *Artificial Intelligence*, 186:3146–3150, 2012. ISSN 10450823. doi:10.1016/j.artint.2012.03.004.
20. Thomas Eiter, Wolfgang Faber, Nicola Leone, and Gerald Pfeifer. The diagnosis frontend of the dlv system. *AI Communications*, 12(1):99–111, 1999.
21. European Court Reports. C-293/12 Digital Rights Ireland Ltd v Minister for Communications, Marine and Natural Resources; Minister for Justice, Equality and Law Reform; Commissioner of the Garda Síochána; Ireland; and The Attorney General and Others. C-594/12 Digital Rights Irela, 2014.
22. European Parliament and the Council of the European Union. Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications netwo, 2006.
23. European Union. Charter of Fundamental Rights of the European Union 2000/C 364/01. *Official Journal of the European Communities*, 2000.
24. Johannes Klaus Fichte, Markus Hecher, Michael Morak, and Stefan Woltran. Counting Answer Sets via Dynamic Programming. *Informal Proceedings of the First Workshop on Trends and Applications of Answer Set Programming, TAASP 2016, Klagenfurt, Austria, September 26, 2016*, 2016.
25. N Fornara. Specifying and monitoring obligations in open multiagent systems using semantic web technology. *Semantic Agent Systems*, pages 25–45, 2011. doi:10.1007/978-3-642-18308-9_2.

26. N Fornara and M Colombetti. Representation and monitoring of commitments and norms using OWL. *AI Communications*, 23(4):341–356, 2010. ISSN 09217126 (ISSN). doi:10.3233/AIC-2010-0478.
27. Nicoletta Fornara, Daniel Okouya, and Marco Colombetti. Using OWL 2 DL for expressing ACL Content and Semantics. In *European Workshop on Multi-Agent Systems*, pages 97–113, 2012.
28. Dov Gabbay, John Horty, Xavier Parent, Ron van der Meyden, and Leendert van der Torre, editors. *Handbook of Deontic Logic and Normative Systems vol. 1*. 2013.
29. A García-Camino, Pablo Noriega, and Juan-Antonio Rodríguez-Aguilar. An algorithm for conflict resolution in regulated compound activities. In *Seventh Annual International Workshop Engineering Societies in the Agents World*, pages 193–208, 2006.
30. Martin Gebser, Benjamin Kaufmann, and Roland Kaminski. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107–124, 2011.
31. Michael Gelfond. Answer Sets. *Foundations of Artificial Intelligence*, 3:285–316, 2008. ISSN 15746526. doi:10.1016/S1574-6526(07)03007-6.
32. Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Logic Programming: Proceedings of the Fifth International Conference and Symposium*, pages 1070 – 1080, 1988.
33. Guido Governatori and Antonino Rotolo. Changing Legal Systems: Legal Abrogations and Annulments in Defeasible Logic. *Logic Journal of IGPL*, 18:157–194, 2010.
34. Guido Governatori, Monica Palmirani, Regis Riveret, Antonino Rotolo, and Giovanni Sartor. Norm modifications in defeasible logic. In *Legal Knowledge and Information Systems (JURIX 2005)*, pages 13–22. IOS Press, 2005.
35. Guido Governatori, Antonino Rotolo, and Giovanni Sartor. Temporalised normative positions in defeasible logic. In *Proceedings of the 10th international conference on Artificial intelligence and law*, pages 25 – 34, New York, New York, USA, 2005. ACM Press. ISBN 1595930817. doi:10.1145/1165485.1165490.
36. Guido Governatori, Joris Hulstijn, and Antonino Rotolo. Characterising deadlines in Temporal Modal Defeasible Logic. In *Proceedings of the 20th Australian Joint Conference on Artificial Intelligence*, pages 486–496, 2007.
37. Davide Grossi. Pushing Anderson's Envelope: The Modal Logic of Ascription. In *9th International Conference on Deontic Logic in Computer Science (DEON 2008)*, pages 263–277, 2008.
38. Davide Grossi. Norms as ascriptions of violations: An analysis in modal logic. *Journal of Applied Logic*, 9(2):95–112, 2011. ISSN 15708683. doi:10.1016/j.jal.2010.03.002.
39. Davide Grossi, John-Jules Meyer, and Frank Dignum. Modal logic investigations in the semantics of counts-as. In *Proceedings of the 10th international conference on Artificial intelligence and law (ICAIL '05)*, pages 1–19. ACM, 2005. ISBN 1595930817. doi:10.1145/1165485.1165487.
40. Davide Grossi, Huib Aldewereld, Javier Vázquez-Salceda, and Frank Dignum. Ontological aspects of the implementation of norms in agent-based electronic institutions. *Computational and Mathematical Organization Theory*, 12:251–275, 2006. ISSN 1381298X. doi:10.1007/s10588-006-9546-6.
41. Davide Grossi, John-Jules Ch Meyer, and Frank Dignum. Counts-as: Classification or constitution? An answer using modal logic. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4048 LNAI:115–130, 2006. ISSN 03029743. doi:10.1007/11786849_11.
42. Davide Grossi, J. J Ch Meyer, and Frank Dignum. The many faces of counts-as: A formal analysis of constitutive rules. *Journal of Applied Logic*, 6(2):192–217, 2008. ISSN 15708683. doi:10.1016/j.jal.2007.06.008.
43. A Günay and P Yolum. Detecting conflicts in commitments. *Declarative Agent Languages and Technologies IX*, pages 51–66, 2012.
44. Herbert Lionel Adolphus Hart. *The Concept of Law*. Clarendon Press., Oxford, 1961.
45. Liesbet Hooghe and Gary Marks. Unraveling the central state, but how? Types of multi-level governance. *American political science review*, 97(2):233–243, 2003.

46. Jie Jiang. *Organizational Compliance: An Agent-based Model for Designing and Evaluating Organizational Interactions*. PhD thesis, TU Delft, Delft University of Technology, 2015.
47. Andrew J. I. Jones and Marek Sergot. A Formal Characterisation of Institutionalised Power. *Journal of IGPL*, 4(3):427–443, 1996. ISSN 1367-0751. doi:10.1093/jigpal/4.3.427.
48. Thomas C. King. *Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance*. PhD thesis, Delft University of Technology, 2016. doi:10.4233/uuid:82438672-3e8b-477a-a39e-0ce189639e88.
49. Thomas C King, Tingting Li, Marina De Vos, Virginia Dignum, Catholijn M Jonker, Julian Padget, and M Birna Van Riemsdijk. A Framework for Institutions Governing Institutions. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, pages 473–481, Istanbul, Turkey, 2015. International Foundation for Autonomous Agents and Multiagent Systems.
50. Thomas C King, M Birna Van Riemsdijk, Virginia Dignum, and Catholijn M Jonker. Supporting Request Acceptance with Use Policies. In *Coordination, Organizations, Institutions, and Norms in Agent Systems X: COIN 2014 International Workshops, COIN@ AAMAS, Paris, France, May 6, 2014, COIN@ PRICAI, Gold Coast, QLD, Australia, December 4, 2014, Revised Selected Papers*, pages 114 – 131. Springer, 2015.
51. Thomas C King, Virginia Dignum, and Catholijn M Jonker. When Do Rule Changes Count-as Legal Rule Changes? In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016). Frontiers in Artificial Intelligence and Applications. Vol 285.*, pages 3 – 11, 2016. ISBN 9781614996729. doi:10.3233/978-1-61499-672-9-3.
52. Thomas C King, Tingting Li, Marina De Vos, Catholijn M Jonker, Julian Padget, and M Birna Van Riemsdijk. Revising Institutions Governed by Institutions for Compliant Regulations. *Coordination, Organizations, Institutions, and Norms in Agent Systems XI: Lecture Notes in Computer Science, vol 9628. Springer, Cham*, 9628:191 – 208, 2016. doi:10.1007/978-3-319-42691-4_11.
53. Robert Kowalski and Marek Sergot. A logic-based calculus of events. *New Generation Computing*, 4 (1):67–95, 1986.
54. Tingting Li. *Normative Conflict Detection and Resolution in Cooperating Institutions*. PhD thesis, University of Bath, 2014.
55. Tingting Li, Tina Balke, Marina De Vos, Julian Padget, and Ken Satoh. Legal Conflict Detection in Interacting Legal Systems. In *1st International Jurix Doctoral Consortium and Poster Sessions, JURIX-DoCoPe 2013, in Conjunction with the 26th International Conference on Legal Knowledge and Information Systems, JURIX 2013*, 2013.
56. Tingting Li, Tina Balke, Marina De Vos, Ken Satoh, and Julian Padget. Detecting conflicts in legal systems. *New Frontiers in Artificial Intelligence, Lecture Notes in Computer Science*, 7856:174 – 189, 2013.
57. Henrique Lopes Cardoso and E Oliveira. Norm defeasibility in an institutional normative framework. In *European Conference on AI (ECAI '08)*, pages 468 – 473, 2008. ISBN 9781586038915. doi:10.3233/978-1-58603-891-5-468.
58. Henrique Lopes Cardoso and Eugenio Oliveira. A context-based institutional normative environment. In *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, pages 140–155, 2009. doi:10.1007/978-3-642-00443-8_10.
59. Henrique Lopes Cardoso and Eugénio Oliveira. Monitoring directed obligations with flexible deadlines: A rule-based approach. In *International Workshop on Declarative Agent Languages and Technologies*, pages 77 – 92, Budapest, Hungary, 2010. ISBN 3642113540. doi:10.1007/978-3-642-11355-0_4.
60. Henrique Lopes Cardoso and Eugénio Oliveira. Directed deadline obligations in agent-based business contracts. In *Coordination, Organization, Institutions and Norms (COIN@AAMAS)*, volume 6069 LNAI, pages 225–240, 2010. ISBN 3642149618. doi:10.1007/978-3-642-14962-7_15.
61. Fabiola López Y López and Michael Luck. Modelling Norms for Autonomous Agents. In *Proceedings of The Fourth Mexican Conference on Computer Science*, pages 238–245. IEEE Computer Society,

2003.
62. Fabiola López y López, Michael Luck, and Mark D'Inverno. A normative framework for agent-based systems. *Computational and Mathematical Organization Theory*, 12(2-3):227–250, oct 2006. ISSN 1381-298X. doi:10.1007/s10588-006-9545-7.
63. D Makinson and Leendert van der Torre. What is input/output logic? *Trends in Logic*, 17:163–174, 2003.
64. Donald Nute. Defeasible logic. In Dov M. Gabbay, C. J. Hogger, and J .A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3. Oxford University Press, 1987. ISBN 978-3-540-00680-0. doi:10.1007/3-540-36524-9_13.
65. W Pieters, Julian Padget, and F Dechesne. Obligations to enforce prohibitions: on the adequacy of security policies. In *Proceedings of the 6th International Conference on Security of Information and Networks*, pages 54–61, 2013. ISBN 9781450324984.
66. W. Pieters, J. Padget, F. Dechesne, V. Dignum, and H. Aldewereld. Effectiveness of qualitative and quantitative security obligations. *Journal of Information Security and Applications*, 22:3 – 16, 2015. ISSN 22142126. doi:10.1016/j.jisa.2014.07.003.
67. Jeremy Pitt and Alexander Artikis. The open agent society: retrospective and prospective views. *Artificial Intelligence and Law*, 23(3):241–270, 2015. ISSN 0924-8463. doi:10.1007/s10506-015-9173-y.
68. Jeremy Pitt and Ada Diaconescu. Structure and Governance of Communities for the Digital Society. In *IEEE International Conference on Autonomic Computing (ICAC)*, pages 279–284, 2015. ISBN 9781467369718. doi:10.1109/ICAC.2015.62.
69. H Prakken and G Sartor. A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law*, 4:331–368, 1996.
70. John R. Searle. *Speech acts: An essay in the philosophy of language*. Cambridge university press, 1969.
71. John R. Searle. *Intentionality: An essay in the philosophy of mind*. Cambridge university press, 1983.
72. John R. Searle. *The Construction of Social Reality*. The Free Press, New York, 1995.
73. John R. Searle. What is an institution? *Journal of Institutional Economics*, 1:1–22, 2005. ISSN 1744-1374. doi:10.1017/S1744137405000020.
74. UK. The Data Retention (EC Directive) Regulations 2009, No. 859, 2009.
75. L van der Torre and Y Tan. The temporal analysis of Chisholm's paradox. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI'98)*, pages 650–655, 1998.
76. G H von Wright. Deontic logic. *Mind*, 60(237):1–15, 1951.
77. Pnar Yolum and MP Singh. Reasoning about commitments in the event calculus: An approach for specifying and executing protocols. *Annals of Mathematics and Artificial Intelligence*, 42(1-3): 227–253, 2004.

Appendices can be downloaded from http://thomascking.com/JAAMAS_Automated_Multilevel_Governance_Compliance_Checking/appendices.pdf.