A Framework for Institutions Governing Institutions

Thomas C. King[1], Tingting Li[2], Marina De Vos[3], Virginia Dignum[1], Catholijn M. Jonker[1], Julian Padget[3], M. Birna van Riemsdijk[1]

[1]*Delft University of Technology* {t.c.king-1, M.V.Dignum, C.M.Jonker, M.B.vanRiemsdijk}@tudelft.nl

[2]*Imperial College London* tingting.li@imperial.ac.uk

[3]*University of Bath* {mdv, jap}@cs.bath.ac.uk

ABSTRACT

Norms guide multi-agent systems away from being potentially anarchic towards a coordinated and collaborative society. Institutions provide an explicit, external representation of norms as well as the means to detect violations and other conditions. Each institution can be crafted individually to capture their designers' goals, but this creates a challenge at higher levels of authority in guiding the institutional design to be coordinated with other institutions and not imposing unacceptable limits on agents' rights. We propose to use institutions to govern and to revise institutions, following a principle widely encountered in the social world, where treaties, primary legislation, framework agreements and subsidiarity establish a regulatory space by defining norms on the form of a body of regulation. We set out a formal and computational framework, building on the InstAL model and implementation, to construct tiers of institutions, where the norms at each tier are governed by those at the tier above. Thus, agents' behaviour is governed and monitored by a tier-1 institution, whose norms are governed and monitored by a tier-2 institution, etc.. This allows us to check the compliance of an institution with the tier above. Compliance failure generates the necessary negative examples for automatic norm-revision.

Categories and Subject Descriptors

[**Knowledge representation and reasoning**]: Logic programming and answer set programming

General Terms

Legal Aspect; Verification

Keywords

Normative Systems; Institutions; Higher-order Norms

1. INTRODUCTION

An MAS (Multi-Agent System) is liable to exhibiting undesirable behaviour contrary to the system's goals, due to agents' intrinsic autonomy [27]. Institutions govern MAS, containing rules to impose norms (obligations and prohibitions) that guide agents' behaviour. However, institutions themselves can promote subjectively undesirable behaviour by imposing undesirable norms. This is due to the inherent autonomy of an institutions' designers who can specify an institution that, for example, imposes norms that violate agents' rights or lacks coordination with other institutions (e.g. national legislation not imposing an international policy).

To address this, we propose using institutions to govern, guide and revise institutions, with *tiers* of institutions, where the first-tier is an institution governed by the second-tier institution and so on. This is by each tier obliging and prohibiting the imposition of norms by the institution in the tier below, that is, imposing *higher-order* norms. Then, each tier *monitors* the tier below for whether its regulations impose norms that are compliant with these higher-order norms. To this end, we propose a vertical governance structure of institutions governing other institutions, which we call a *multi-tier* institution. This consists of a first-tier institution imposing *first-order* norms on an MAS' members, a second-tier institution imposing *second-order* norms on the first-tier and each tier-i imposing ith-order norms on the tier below.

Multi-tier governance is widely found in the social world, also known as vertical and multi-level governance, with benefits widely identified in Political Science literature [18]. The main benefit being, following the subsidiarity principle, what can be done at the local level is left up to the local level. So, an institution prescribing to a lower-tier institution the norms it should impose gives flexibility to the lower-tier designers in how and whether to comply without the redesign of the institution being regimented. For example, the designers can choose to remove a prohibited norm, replace it with a permitted norm that achieves the same goal, or forego redesign if the prohibited norm's benefits outweigh the costs of a non-compliant institution. Examples include service policies governing contracts offered for the provision of services, where service policy compliance is a prerequisite for contractual agreement; and in the political sphere EU Directives [24] governing member states' legislation, and the UK's Financial Conduct Authority governing financial institutions and the obligations they impose on clients [25].

The problem is that AI frameworks for modelling normative systems have focussed on institutions governing MAS (see [1] for a review), while work in Political Science has not operationalised institutions governing institutions. This raises the question, to what extent do institutional modelling techniques in AI model multi-tier institutions and higher-order norms? We address this with a framework for the representation, modelling and automated compliance-checking of multi-tier institutions that extends the InstAL framework [7] and provides input for existing norm revision techniques to make a multi-tier institution compliant.

In the rest of the paper, we first introduce a contracting case-study (2) and then an informal overview of the framework (3). Then, we propose a formal framework (4) that extends the InstAL framework for individual institutions [7] to institutions governing other institutions (4.1), to represent and model multi-tier institu-

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

tions (4.2). The formal framework is implemented with a computational framework in ASP (Answer-Set Programming) for automated compliance checking (5). The computational framework then provides input for ILP (Inductive-Logic Programming) based norm revision to make institutions compliant (6). We compare our framework with others (7) and conclude discussing the framework's implications and wider applications (8).

2. CASE STUDY: SERVICE CONTRACTS

The case study we look at focuses on crowdsensing, the crowdsourcing of people's services and mobile sensors [10], for obtaining a large number of photographic weather damage reports over a wide geographic area on demand over a period of time. We specifically look at the interaction between two agents, the service *requestor* and the potential *service provider*.

Since it is uncertain whether the service provider will provide photographic weather reports, the idea is to make a binding agreement (contract), potentially with rewards and punishments. A contract is a type of legal institution [22] (modelled as such in e.g. [4]) and in this case if accepted governs the interaction between the requestor and service provider, an MAS, and is therefore a *first-tier* institution. However, contracts can contain terms imposing undesirable norms from the service provider's point of view. To address this, the service provider has in turn specified a *service policy*, a *second-tier* institution governing any contracts offered where compliance is a prerequisite for agreement. We begin with a description of a contract for the provision of photographic weather report services.

Contract. The proposed contract is for the service provider to provide photographic weather damage reports on-demand. The service requestor wants to determine the service provider's location and request photographs if they are in an area of interest. Thus, when requested the provider is obliged to provide their location before they leave the area. Additionally, when requested, the service provider is obliged to provide a photographic report before leaving their current location.

The service provider has in turn defined a *service policy* governing the contract by obliging and prohibiting it from imposing norms, where any non-compliant contract is rejected. Outside of the norms the service policy obliges and prohibits, the service requestor has freedom in the contract specification whilst remaining compliant.

Service Policy. The service policy prescribes which norms should be imposed by a contract and in what order. The service provider should only be obliged to provide a photograph after *accepting* a request. Thus, after a photograph is requested, the service policy imposes a second-order norm obliging the request is accepted *before* the the obligation to provide the photograph is imposed. The provider does not want to provide their location if it is sensitive, so when the service provider is in a sensitive location, it is prohibited to oblige them to reveal their location.

Composing the contract and service policy gives us a multi-tier institution with two tiers, that is, a *two-tier* institution. The case study is intentionally simple for illustrative purposes (for example we assume a trusted third-party can verify the service provider is in a sensitive location without revealing it). In principle, the framework can be applied to cases with further institutional tiers. For example, if the service provider is employed by an organisation to engage in contracts with clients and provide the service, then the employer could give them the task of crafting the service policy, whilst guiding them in including certain second-order norms with a third-tier institution imposing third-order norms (e.g. the service policy should demand payment for services).

3. OVERVIEW

This section overviews the framework for representing individual institutions, composing multi-tier institutions from individual institutions and modelling multi-tier institutions. Following Searle's definition [23], an individual institution is a set of regulative and constitutive rules that govern a system. The idea is then that, since individual institutions govern systems, be it an MAS or another institution, they can be placed in different tiers of a multi-tier institution. The different tiers are then linked such that each tier monitors the legality of the tier below (an MAS or another institution). Thus, some institutions govern other institutions and through monitoring we can determine the compliance of the norms one institution imposes, such as on an MAS, with the institution it is governed by.

Each institution specifies the events that can occur and the fluents that can hold (properties whose truth value changes over time) in the institution, and how it evolves over time. An institution's evolution is specified in terms of which institutional events and state changes occur when - in response to specific external events occurring in specific institutional contexts. The occurrence of an institutional event is according to the specification of what Searle calls *constitutive rules* [23, p. 35] stating when one event counts as another in a given institutional context (e.g. killing someone when not at a time of war counts as murder). The evolution of the institutional state is described by which fluents do and do not hold from one state to the next state following the occurrence of events. The specification of an institution's evolution is used to produce an institutional model: a sequence of events and a sequence of states, where events transition between states, in response to the events and state changes occurring in the tiers below.

An institution's norms are rules that impose obligations and prohibitions, thus we model permissive societies which permit anything not prohibited, represented as fluents. The following describes their meaning:

- **Obligation:** Expresses an event/fluent should occur/hold before a deadline event/fluent. If neither the aim nor deadline are normative fluents, it is a first-order norm. Otherwise it is a normative fluent about *other* normative fluents and thus a higher-order normative fluent. It is discharged when the event/fluent it obliges occurs/holds strictly before the deadline, and violated if the deadline occurs/holds before the aim.
- **Prohibition:** The same as obligation but is discharged if the deadline occurs/holds first, and violated if the aim it prohibits occurs/holds before or at the same time as the deadline.

The approach we take to *representing* multi-tier institutions is to compose them from individual institutions structured as follows. Tier-0 is the MAS and represented as a domain of propositions denoting observable events, and fluents. Each tier above is an institution governing the tier below by imposing norms about the events/fluents occurring in the tier below. Tier-1 imposes first-order norms about the MAS events and fluents. Each tier-i above imposes ith-order norms about the normative fluents in the tier below.

The approach we take to *modelling* multi-tier institutions is to extend the InstAL [7] framework from modelling normative fluents about events to normative fluents about events and fluents. This includes other normative fluents and thus allows us to model higher-order norms. Then, we model a multi-tier institution as a whole by linking each tier with a vertical governance structure such that the normative events that occur and fluents that hold in one tier are 'sent up' to the tier above for monitoring, and each tier is modelled

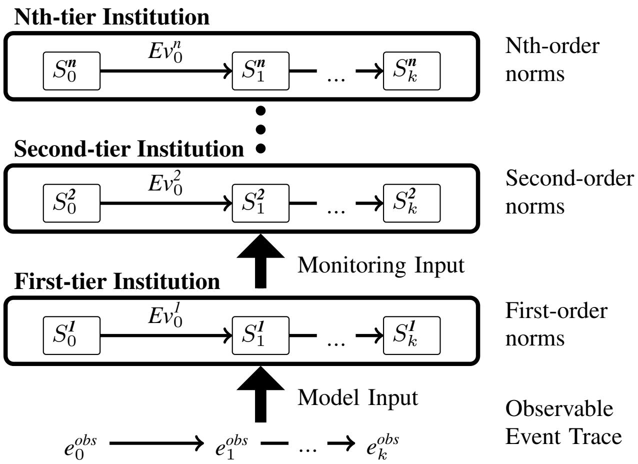

Figure 1: Modelling a multi-tier institution

as an individual institution with input from the tier below.

In more detail, a multi-tier institution model, depicted in Figure 1, is produced in response to a trace of observable events occurring in the MAS. The first-tier is modelled in response to the trace of observable events, such that the MAS' initial event (e_0^{obs}) causes events to occur in the first-tier institution (at Ev_0^1), such as norm dischargement/violation, according to the initial institutional state (S_0^1). This advances the first-tier institution to the next state (S_1^1). This continues for every event and state in the observable event trace until a corresponding sequence of sets of events and a sequence of events is produced, that is, the model of the first-tier is produced. Each tier above the first is also modelled in response to the observable event trace with additional input from the normative events and state changes in the tier below.

Each institutional model also acts as a legal monitor for the tier below, the first-tier monitors the legality of events in the observable event trace and the fluents they cause to hold, non-compliance causing a first-order norm violation event. Higher tiers, meanwhile, monitor the legality of the normative fluents in the tier below, non-compliance causing higher-order norm violation events. For example, if in the second-tier institution's state S_0^2 a second-order normative fluent prohibits a first-order normative fluent being imposed, but the first-tier's state S_0^1 imposes the prohibited norm, then a corresponding second-order norm violation event occurs next (in Ev_0^2). A multi-tier institutional model shows the compliance of each tier with the tier above for an event trace.

4. FORMAL FRAMEWORK

This section presents the formal framework for representing and operationalising individual (4.1) and multi-tier institutions (4.2).

4.1 Individual Institutions

The formal framework for *individual institutions*, consists of two main parts: the specification of individual institutions and their operational semantics. We start with the representation of normative fluents, which oblige/prohibit an event occurring or another fluent holding (the aim) before an event occurs or fluent holds (the deadline). The language of normative fluents is over a set of propositions denoting fluents and events describing the system being governed. If the set of propositions includes only non-normative events and fluents, then only first-order normative fluents can be expressed. If, however, the set of propositions contains first-order normative fluents, then second-order normative fluents can be expressed and so on. Such higher-order normative fluents are categorised as: obliging/prohibiting a normative fluent holds before an event or non-normative fluent holds, obliging/prohibiting an event or non-normative fluent before a normative fluent holds, and obliging/prohibiting a normative fluent to hold before another normative fluent holds.

Definition 1. **Normative Fluents** Let P be a set of propositions denoting fluents and events, a be the norm's aim, d the deadline and $a, d \in P$. The set of normative fluents $\mathcal{N}|_P$ is the set of all norms n expressed as:

$$n \ ::= \ obl(a, d) \ \mid pro(a, d)$$

The contract imposes a normative fluent obliging the service provider to share their location, when requested, before they leave the area. The corresponding normative fluent is below (upper-case terms denote variables acting as shorthand for their instantiations):

$$obl(provide\_photo(Location, Agent\_1, Agent\_2),\\ leave(Location, Agent\_1))$$

The normative fluents the second-tier institution specifies always include, in the aim or the deadline, a first-order normative fluent. Thus, they are second-order normative fluents. The following second-order normative fluent prohibits the aforementioned first-order norm, when the service provider enters a sensitive area, until the service provider leaves the area.

$$pro(obl(provide\_location(Location, Agent\_1, Agent\_2),\\ leave(Location, Agent\_1)),\\ leave(Location, Agent\_1))$$

Second-order normative fluents can also oblige an event or fluent before a normative fluent is imposed. The following second-order normative fluent is imposed when the service provider is requested to provide a photograph, obliging the request is *accepted* before the service provider is obliged to provide a photograph.

$$obl(accept\_photo\_request(Location, Agent\_1, Agent\_2),\\ obl(provide\_photo(Location, Agent\_1, Agent\_2),\\ leave(Location, Agent\_1)))$$

An institutional specification gives the *signature* of events ($\mathcal{E}$) that can occur and fluents ($\mathcal{F}$) that can hold in the institution. The signature is specified from a set of propositions P to which the institution $\mathcal{I}|_P$ is restricted to (just $\mathcal{I}$ is used if P is unimportant). The events $\mathcal{E}$ is the set of observable events ($\mathcal{E}_{obs}$), and the set of institutional events ($\mathcal{E}_{inst}$) which consists of events signifying something unrelated to a norm has happened ($\mathcal{E}_{instact}$), or a norm has been discharged/violated ($\mathcal{E}_{norm}$). The set of an institution's fluents ($\mathcal{F}$), describe the state of a domain ($\mathcal{F}_{dom}$) such as that being governed (e.g. an agent is at a location), and the normative fluents ($\mathcal{F}_{norm}$) that can hold in the institution.

An institutional specification also describes how the institution evolves over time starting with an initial state (Δ). This is in terms of the events and state changes that occur in response to the occurrence of events, where, an institutional state is a set of fluents that are true at that point in time, $\Sigma = 2^{\mathcal{F}}$ denoting the set of all possible institutional states. Describing the events and state changes that occur are an *event generation* function and *state consequence* function. Both functions' arguments are a condition on a state, describing the things that must and must not hold in a state to cause the events/state change, and a set of events. A condition on a state is described with state formulae, $\mathcal{X} = 2^{\mathcal{F} \cup \neg \mathcal{F}}$ denoting the set of all state formulae, where $\neg \mathcal{F} = \{\neg f \mid f \in \mathcal{F}\}$ is the weak negation of all fluents denoting they do not hold. The event generation function ($\mathcal{G} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{E}_{inst}}$) states when, conditional on a state, one event counts-as another. The consequence function ($\mathcal{C} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{F}} \times 2^{\mathcal{F}}$) provides the fluents that are initiated and terminated by events from one state to the next. We use $\mathcal{C}(X, e) = \langle \mathcal{C}^{\uparrow}(X, e), \mathcal{C}^{\downarrow}(X, e) \rangle$ to denote the consequence function's result, where $\mathcal{C}^{\uparrow}(X, e)$ is the set of fluents initiated and $\mathcal{C}^{\downarrow}(X, e)$ is the set of fluents terminated by the event e when the state entails the state condition X.

Definition 2. **Individual Institution** An institution is a tuple $\mathcal{I}|_P = \langle \mathcal{E}, \mathcal{F}, \mathcal{C}, \mathcal{G}, \Delta \rangle$, restricted to the set of propositions P, given $\mathcal{E}_{obs}, \mathcal{E}_{instact}, \mathcal{F}_{dom} \subseteq P, \mathcal{I}|_P$ is defined as:

- $\mathcal{F}_{norm} \subseteq \mathcal{N}|_P$ is a set of normative fluents.
- $\mathcal{F} = \mathcal{F}_{dom} \cup \mathcal{F}_{norm}$ is a set of fluents.
- $\mathcal{E}_{norm} = \{disch(n), viol(n) \mid n \in \mathcal{F}_{norm}\}$
- $\mathcal{E}_{inst} = \mathcal{E}_{instact} \cup \mathcal{E}_{norm}$ where $\mathcal{E}_{instact}$ and $\mathcal{E}_{norm}$ are disjoint.
- $\mathcal{E} = \mathcal{E}_{obs} \cup \mathcal{E}_{inst}$.
- $\mathcal{C} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{F}} \times 2^{\mathcal{F}}$ is a state consequence function.
- $\mathcal{G} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{E}_{instact}}$ is an event generation function.
- $\Delta \subseteq \mathcal{F}$ is the initial institutional state.

To ascertain an institution's evolution over time, we extend InstAL's [7] operational semantics from producing models for institutions governing MAS to models of institutions governing other institutions. The main changes are modelling normative fluents about events or other fluents and thereby modelling higher-order norms. We also introduce norm dischargement events and model permissive societies (anything not prohibited is permitted) with prohibitions rather than prohibitive societies containing just obligations and permissions (anything not permitted is prohibited). Given these minor changes we describe rather than reintroduce InstAL's operational semantics, and how we extend them.

The operational semantics for an institution $\mathcal{I}$ are defined with an *event generation operation* and a *state transition operation*. The event generation operation, $GR : \Sigma \times 2^{\mathcal{E}} \rightarrow 2^{\mathcal{E}}$, takes as input a state and a set of events and outputs: the events input, events generated by the institution's event generation function, and norm dischargement/violation events caused by an event occurring or a fluent holding. The event generation function's fixpoint is denoted with GR^{ω}, representing all the events that occur in response to a set of events in a state.

The state transition operation, $TR : \Sigma \times 2^{\mathcal{E} \cup \{e_{null}\}} \rightarrow \Sigma$ is a total function producing the next state, from the current state $S_i \in \Sigma$ and a set of events E. If only the null event e_{null} occurs then the next state S_{i+1} is the same as the previous. Otherwise, S_{i+1} contains all the fluents in S_i not terminated (inertia), all initiated fluents according to the institution's consequent function $\mathcal{C}$, ($f \in \mathcal{C}^{\uparrow}(X, e)$) due to the occurrence of an event $e \in E$ for some X entailed by the state S_i. The next state S_{i+1} does not contain any fluents terminated by the institution's consequence function ($f \in \mathcal{C}^{\downarrow}(X, e)$) or discharged/violated normative fluents ($disch(f) \in E$ or $viol(f) \in E$).

Together, these operations allow an individual institution model to be produced in response to a trace of events. Since we focus on multi-tier institutions, we use these operations to give the multi-tier operational semantics in the next section.

4.2 Multi-tier Institutions

In this section we present the specification of *multi-tier* institutions as a composition of individual institutions in a vertical governance structure, and their operational semantics.

The approach we take to representing a multi-tier institution is to restrict each ith-tier such that it can contain events and fluents from all the tiers below for monitoring, but can only govern the tier directly below by imposing ith-order norms. This restriction is defined by, starting with a set of propositions P which describe the MAS' events and fluents, the first-tier $\mathcal{I}^1|_{P^1}$ imposes normative fluents over the events and fluents of the MAS (i.e. first-order normative fluents) such that $P^1 = P$.

Then, each ith-tier above the first $\mathcal{I}^i|_{P^i}$ can contain the normative fluents and events (dischargement and violation) from the tiers below for monitoring ($P^i = P^{i-1} \cup \mathcal{N}|_{P^{i-1}} \cup \mathcal{E}^{i-1}_{norm}$), but each ith-tier is restricted in only initiating and terminating normative fluents over these (i.e. ith-order norms). This means, an institution can potentially also impose norms about the dischargement and violation of norms in the tier below. We leave this to the discretion of the designer, since in some cases it can make sense, for example obliging a norm is violated before an obligation to pay a fine is imposed. Finally, for monitoring, to ensure the ith-tier contains the events and fluents produced from the tier below, a function $\mathcal{GX}^i$ provides all the normative events recognised from the tier below, and $\mathcal{CX}^i$ provides normative fluents from the tier below.

Definition 3. **Multi-tier Institution** Let P be a set of propositions denoting the domain, a multi-tier institution is a tuple $\mathcal{M} = \langle \mathcal{T}, \mathcal{GX}^i, \mathcal{CX}^i \rangle$ where:

- $\mathcal{T} = \langle \mathcal{I}^1|_{P^1}, ..., \mathcal{I}^n|_{P^n} \rangle$ is an n-tuple of institutions s.t. $\forall i \in [n] : \mathcal{I}^i|_{P^i} = \langle \mathcal{E}^i, \mathcal{F}^i, \mathcal{C}^i, \mathcal{G}^i, \Delta^i \rangle, \mathcal{X}^i = 2^{\mathcal{F}^i \cup \neg \mathcal{F}^i}$, $\Sigma^i = 2^{\mathcal{F}^i}$
- $P^1 = P$ and $\forall i \in [2, n]$, $P^i = P^{i-1} \cup \mathcal{N}|_{P^{i-1}} \cup \mathcal{E}^{i-1}_{norm}$ - each ith-tier can contain the events and fluents that can be defined in the tier below and normative fluents over these.
- $\forall i \in [2, n], \forall S \in \mathcal{X}^i, \forall e \in \mathcal{E}^i : \mathcal{C}^{i\uparrow}(S, e) \cap \mathcal{N}|_{P^{i-1}} = \emptyset, \mathcal{C}^{i\downarrow}(S, e) \cap \mathcal{N}|_{P^{i-1}} = \emptyset$ - the ith-tier can only initiate and terminate ith-order norms.
- $\forall i \in [n-1] : \mathcal{GX}^i : 2^{\mathcal{E}^i} \rightarrow 2^{\mathcal{E}^{i+1}}$ is an event filtering function defined as $\forall i \in [n-1], \forall E \in 2^{\mathcal{E}^i} : \mathcal{GX}^i(E) = E \cap \mathcal{E}^i_{norm} \cap \mathcal{E}^{i+1}$
- $\forall i \in [n-1] : \mathcal{CX}^i : \Sigma^i \rightarrow \Sigma^{i+1}$ is a fluent filtering function defined as $\forall i \in [n-1], \forall F \in \Sigma^i : \mathcal{CX}^i(F) = F \cap \mathcal{F}^i_{norm} \cap \mathcal{F}^{i+1}$

To exemplify, the running example is formalised in Table 1. For brevity we leave out the observable events $\mathcal{E}^i_{obs}$ which correspond to institutional events in $\mathcal{E}^i_{inst}$ prepended with *ext_*. Similarly, the generation function does not include institutional events generated from observable events. Finally, the set of fluents is left out, which are inferred from the generation and consequence functions.

A multi-tier institution model is produced, consisting of a model of each individual institution, in response to a *composite trace* [16] of events, where each event is observable to at least one tier.

Definition 4. **Composite Event Trace** Let $\mathcal{M} = \langle \mathcal{T}, \mathcal{GX}, \mathcal{CX} \rangle$ be a multi-tier institution s.t. $\mathcal{T} = \langle \mathcal{I}^1, ..., \mathcal{I}^n \rangle$. $ctr = \langle e_0, ..., e_k \rangle$ is a composite trace for $\mathcal{M}$ iff $\forall j \in [0, k], \exists i \in [n] : e_j \in \mathcal{E}^i_{obs}$

Service Contract

$\mathcal{E}_{inst} = \{enter(Location\_1, Agent\_1)), leave(Location\_1, Agent\_1),$
$provide\_location(Location\_1, Agent\_1, Agent\_2),$
$provide\_photo(Location\_1, Agent\_1, Agent\_2)),$
$request\_location(Agent\_1, Agent\_2)$
$request\_photo(Location\_1, Agent\_1, Agent\_2)$
$accept\_photo\_request(Location\_1, Agent\_1, Agent\_2)\}\}$
$\mathcal{G}(\mathcal{X}, \mathcal{E})$:
$\langle \{at(Location\_1, Agent\_1)\}, enter(Location\_2, Agent\_1)\rangle \rightarrow$
$\{leave(Location\_1, Agent\_1)\}$
$\mathcal{C}^{\uparrow}(\mathcal{X}, \mathcal{E})$:
$\langle \emptyset, enter(Location\_1, Agent\_1)\rangle \rightarrow \{at(Location\_1, Agent\_1)\}$
$\langle \emptyset, request\_photo(Location\_1, Agent\_1, Agent\_2)\rangle$
$\rightarrow \{obl(provide\_photo(Location\_1, Agent\_1, Agent\_2),$
$leave(Location\_1, Agent\_1))\}$
$\langle \{at(Location\_1, Agent\_2)\}, request\_location(Agent\_1, Agent\_2)\rangle$
$\rightarrow \{obl(provide\_location(Location\_1, Agent\_2, Agent\_1),$
$leave(Location\_1, Agent\_2))\}$
$\mathcal{C}^{\downarrow}(\mathcal{X}, \mathcal{E})$:
$\langle \emptyset, leave(Location\_1, Agent\_1)\rangle \rightarrow \{at(Location\_1, Agent\_1)\}$

Service Policy

$\mathcal{E}_{inst} = \{enter(Location\_1, Agent\_1), leave(Location\_1, Agent\_1),$
$request\_photo(Location\_1, Agent\_1, Agent\_2),$
$request\_location(Agent\_1, Agent\_2),$
$accept\_photo\_request(Location\_1, Agent\_1, Agent\_2)\}$
$\mathcal{G}(\mathcal{X}, \mathcal{E})$:
$\langle \{at(Location\_1, Agent\_1)\}, enter(Location\_2, Agent\_1)\rangle \rightarrow$
$\{leave(Location\_1, Agent\_1)\}$
$\mathcal{C}^{\uparrow}(\mathcal{X}, \mathcal{E})$:
$\langle \emptyset, enter(Location\_1, Agent\_1)\rangle \rightarrow \{at(Location\_1, Agent\_1)\}$
$\langle \{sensitive(Location\_1)\}, enter(Location\_1, Agent\_1)\rangle \rightarrow$
$\{pro(obl(provide\_location(Location\_1, Agent\_1, Agent\_2),$
$leave(Location\_1, Agent\_1)),$
$leave(Location\_1, Agent\_1))\}$
$\langle \{\emptyset, request\_photo(Location\_1, Agent\_1, Agent\_2)\rangle \rightarrow$
$\{obl(accept\_photo\_request(Location\_1, Agent\_1, Agent\_2),$
$obl(provide\_photo(Location\_1, Agent\_1, Agent\_2),$
$leave(Location\_1, Agent\_1)))\}$
$\mathcal{C}^{\downarrow}(\mathcal{X}, \mathcal{E})$:
$\langle \emptyset, leave(Location\_1, Agent\_1)\rangle \rightarrow \{at(Location\_1, Agent\_1)\}$

Table 1: Formalisation of the service policy governing a service contract

Only events an institution recognises advance its state, potentially causing unsynchronised institutions. We address this with a *synchronised trace* for each institution [16] replacing unrecognised events in the composite trace with a null event e_{null} to advance the institution's state.

Definition 5. **Synchronised Trace** Let $\mathcal{I}$ be an institution, and $ctr = \langle e_0, ..., e_k\rangle$ be a composite 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.

A multi-tier institutional model in response to a composite event trace consists of a model for each tier-i institution ($M^i = \langle S^i, E^i\rangle$) - a sequence of states (S^i) and a sequence of sets of events (E^i) in response to the synchronised event trace str^i. The model for each tier-i institution represents its evolution over time, where each set of events E^i_j causes a transition from one state S^i_j to the next S^i_{j+1}.

Each institutional model starts with an initial state. For the first-tier this is the same initial state in its specification (Δ^1), for all the other tier-i institutions it is the initial state in its specification Δ^i combined with the initial state of the tier below filtered to only contain normative fluents by $\mathcal{CX}^{i-1}$. Each set of events E^i_j in each tier-i institution's model is produced by the institution's event generation operation fixpoint ($GR^{i\omega}$). The arguments for the first tier's event generation operation are the current institutional state (S^1_j) and an event in the synchronised event trace (se^1_j), for all other tier-i institutions the events used also include those produced by the tier below (E^{i-1}_j) filtered to only normative events by $\mathcal{GX}^{i-1}$. Each succeeding state S^i_{j+1} for each tier-i institution is produced with tier-i's state transition operation TR^i. For the first-tier, the arguments for the state transition operation are its current state S^1_{j+1} and the set of events produced E^1_j, for all other tier-i institutions the successor state S^i_{j+1} also includes the fluents of the state in the tier below S^{i-1}_{j+1} filtered to only normative fluents by $\mathcal{CX}^{i-1}$, minus any normative fluents that no longer hold in the tier below.

Definition 6. **Multi-tier Institution Model** Let $\mathcal{M} = \langle \mathcal{T}, \mathcal{GX}^i, \mathcal{CX}^i\rangle$ be a multi-tier institution s.t. $\mathcal{T} = \langle \mathcal{I}^1, ..., \mathcal{I}^n\rangle$. Let *ctr* be a composite trace for $\mathcal{M}$ and $\forall i \in [n]$: $str^i = \langle se^i_0, ..., se^i_k\rangle$ be a synchronised trace of *ctr* for $\mathcal{I}^i$. A model of $\mathcal{M}$ in response to *ctr* is $M^{\mathcal{M}} = \langle M^1, ..., M^n\rangle$, of the form $\forall i \in [n] : M^i = \langle S^i, E^i\rangle, S^i = \langle S^i_0, ..., S^i_{k+1}\rangle, E^i = \langle E^i_0, ..., E^i_k\rangle$ where:

- $\forall i \in [n] : GR^i : \Sigma^i \times 2^{\mathcal{E}^i} \rightarrow 2^{\mathcal{E}^i}$ - is the event generation operation for $\mathcal{I}^i$.
- $\forall i \in [n] : TR^i : \Sigma^i \times 2^{\mathcal{E}^i \cup \{e_{null}\}} \rightarrow \Sigma^i$ - is the state transition operation for $\mathcal{I}^i$.
- $S^1_0 = \Delta^1$ - the first state of the tier-1.
- $\forall j \in [0, k] : E^1_j = GR^{1\omega}(S^1_j, se^1_j)$ - events produced by the tier-1 institution.
- $\forall j \in [0, k] : S^1_{j+1} = TR^1(S^1_j, E^1_j)$ - states produced by the tier-1 institution
- $\forall i \in [2, n] : S^i_0 = \Delta^i \cup \mathcal{CX}^{i-1}(S^{i-1}_0)$ - the first state of each tier-i institution above tier-1.
- $\forall i \in [2, n], \forall j \in [0, k] : E^i_j = GR^{i\omega}(S^i_j, \{se^i_j\} \cup \mathcal{GX}^{i-1}(E^i_j))$ - events produced by each tier-i institution above the first.
- $\forall i \in [2, n], \forall j \in [0, k] : S^i_{j+1} = (TR^i(S^i_j, E^i_j) \cup \mathcal{CX}^{i-1}(S^{i-1}_{j+1})) \backslash (S^{i-1}_j \backslash S^{i-1}_{j+1})$ - states produced by each tier-i institution above the first.

An objective of modelling a multi-tier institution is assessing each tier's compliance with the tier above. This supports revising legislation to be compliant, which conducted pre-runtime for a contract makes it acceptable to the party governing the contract. A partial-compliance check can be performed for a single composite trace deemed interesting, or a full-compliance check can be performed for all possible composite traces. A multi-tier institution is non-compliant for a composite trace if a higher-order norm is violated, otherwise it is compliant for that trace. A compliance check provides the set of violations between institutions.

Definition 7. **Multi-tier Institution Violation** Let $\mathcal{M} = \langle \mathcal{T}, \mathcal{GX}, \mathcal{CX}\rangle$ be a multi-tier institution, $\mathcal{T} = \langle \mathcal{I}^1, ..., \mathcal{I}^n\rangle$ and $\forall i \in [n] : \mathcal{I}^i = \langle \mathcal{E}^i, \mathcal{F}^i, \mathcal{C}^i, \mathcal{G}^i, \Delta^i\rangle$. Let *ctr* be a composite trace for $\mathcal{M}$ and $M^{\mathcal{M}} = \langle M^1, ..., M^n\rangle$ be a model for $\mathcal{M}$ w.r.t. *ctr*. The tuple of multi-tier violation sets $V(\mathcal{M}, ctr) = \langle V^1, ..., V^{n-1}\rangle$ is defined as:

- $\forall i \in [n-1] : V^i = (\mathcal{E}^{i+1}_{viol} \backslash \bigcup^i_{j=1} \mathcal{E}^j_{viol}) \cap \bigcup^n_{i=0} E_i$

ctr is identified as an ith-tier institution *non-compliance trace* for $i \in [n-1]$ if there is at least one violation between the ith-tier institution and the institution above, s.t. $V^{i+1} \neq \emptyset$.

5. COMPUTATIONAL FRAMEWORK

In this section we provide the computational framework for multi-tier institutions representation and operational semantics as a set of ASP rules (5.1), or in some places examples of rules due to space constraints. Then, we give the results of using the computational framework for a compliance check on the running example (5.2).

ASP [2, 12], is a non-monotonic logic programming language, that given a problem description as an ASP program, grounds the program and then goes about finding solutions (answer-sets) computed according to the stable-model semantics [12]. There are many answer set solvers available (e.g. [9, 11]), we adhere to CLINGO's syntax [11] described as follows. A CLINGO ASP program is built from atoms and predicates, where predicates containing variables are ground instance schemas. Atoms and predicates can be preceded with any sequence of weak negation (`not`) and classical negation ($-$). A horn clause $\texttt{p\_0} : -\texttt{p\_1}, ..., \texttt{p\_n}$. states that `p_0` is true when $\texttt{p\_1}, ..., \texttt{p\_n}$ is true. Constraints, expressed with $: -\texttt{p\_1}, ..., \texttt{p\_n}.$, are shorthand for falsity in the head, meaning the body cannot be true in any answer set. Finally a choice construct of the form $l\{\texttt{p\_1}, ..., \texttt{p\_n}\}u$, where l and u are positive integers, states that at least l and at most u number of literals $p\_1, ..., p\_n$ need to be true in order for the choice construct to be true (when omitted l is 0 and u is infinity).

5.1 Representation and Operational Semantics in ASP

In this section, we first describe the operational semantics of individual institutions as ASP rules for producing institutional events, reflecting the formal framework's event generation operation, and states, reflecting the formal framework's state transition operation. Then, we provide a translation of multi-tier institutions from their formal to ASP rule representation, by translating each individual institution and linking them in a vertical governance structure with a set of ASP rules. Finally, we provide a set of rules for producing composite, synchronised, traces. The rules comprise an ASP program that produces an Answer-Set for a composite trace, reflecting a multi-tier institution model in the formal framework.

Beginning with event generation, institutional events are generated from ASP rules that produce, from an observable event, institutional events, and from the legality of the events that occur and fluents that hold norm dischargement and violation events. The following rule produces an institution event from an observable event if it occurs externally and is recognised by the institution.

```
occurred(E,In,I) :- evtype(E,In,ex), observed(E,In,I),
   instant(I), inst(In).
```

Rules producing norm dischargement/violation events check to see if a norm holds, and whether an event occurs or fluent holds that discharges/violates the norm, producing the norm dischargement/violation event if it does and terminating the norm at the same time. For brevity, we give an example rule for obligation dischargement and termination below, in total there are four rules for obligation and prohibition dischargement and violation.

```
2{occurred(disch(obl(A, D)),In,I),
  terminated(obl(A, D), In, I)} :-
   holdsat(obl(A, D),In,I), 1{holdsat(A,In,I),
   occurred(A, In, I)}, inst(In), not holdsat(D,In,I),
   not occurred(D, In, I), instant(I).
```

The rules to produce institutional states define the predicate $\texttt{holdsat}(\texttt{P}, \texttt{In}, \texttt{I})$. As in [16], it states that the fluent `P` holds in an institution with the name `In` represented with a fact $\texttt{inst}(\texttt{In})$, at time `I`. It is true if `P` was initiated in the previous state, or it held in the previous state and was not terminated (inertia).

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

$\mathcal{M} = \langle \mathcal{T}, \mathcal{GX}^i, \mathcal{CX}^i \rangle, \mathcal{T} = \langle \mathcal{I}^1, ..., \mathcal{I}^n \rangle, \forall i \in [n]$
$(\mathcal{I}^i = \langle \mathcal{E}^i, \mathcal{F}^i, \mathcal{C}^i, \mathcal{G}^i, \Delta^i \rangle) :$

$$
\begin{array}{rcl}
\mathcal{I}^i & \Leftrightarrow & \texttt{tier}(In,i).\ \texttt{inst}(In). \\
e \in \mathcal{E}^i_{obs} & \Leftrightarrow & \texttt{evtype}(\texttt{e},In,\texttt{ex}). \\
f \in \mathcal{F}^i & \Leftrightarrow & \texttt{ifluent}(\texttt{f},In). \\
\mathcal{C}^{i\uparrow}(X,e) = P & \Leftrightarrow & \forall p \in P : \\
 & & \texttt{initiated}(p,In,\texttt{I}) : - \\
 & & \texttt{occurred}(e,In,\texttt{I}),\ EX(X, In, I). \\
\mathcal{C}^{i\downarrow}(X,e) = P & \Leftrightarrow & \forall p \in P : \\
 & & \texttt{terminated}(p,In,\texttt{I}) : - \\
 & & \texttt{occurred}(e,In,I),\ EX(X, In, I). \\
\mathcal{G}^i(X,e) = E & \Leftrightarrow & \forall e' \in E : \\
 & & \texttt{occurred}(e',In,\texttt{I}) : - \\
 & & \texttt{occurred}(e,In,\texttt{I}),\ EX(X, In, I). \\
f \in \Delta^i & \Leftrightarrow & \texttt{holdsat}(f,In,\texttt{I}) : -\texttt{start}(\texttt{I}).
\end{array}
$$

Figure 3: Multi-tier institution translation into ASP.

A multi-tier institution's translation from its formal representation into ASP rules is given in Figure 3. The ASP rules state when a fluent is initiated (`initiated/3`) and terminated (`terminated/3`), when one event causes an institutional event to occur (`occurred/3`), and the set of fluents (`ifluent/2`) and events (`evtype/3`) recognised by the institution. Keeping with InstAL's convention, for some $X \in \mathcal{X}^i$, *EX(X, In,* `I`*)* is shorthand for a comma delimited set of ASP literals of the form $\texttt{holdsat}(f, \texttt{In}, \texttt{I})$ for all positive fluents in X and $\texttt{not holdsat}(f, \texttt{In}, \texttt{I})$ for all negated fluents in X denoting negation as failure.

Further rules in ASP link tiers by passing up norm dischargement and violation events, this reflects the formal definition of a multi-tier institution where, according to $\mathcal{GX}^i$, normative events are 'sent upwards'. An example of the rule ensuring norm dischargement occurring in one tier occurs in all tiers above is given below, for brevity we leave out the rule for norm violation.

```
occurred(disch(N), In_2, I) :-
    occurred(disch(N), In_1, I),
    tier(In_1, I_1), tier(In_2, I_2), I_2 == I_1 + 1.
```

ASP rules also reflect the formal framework's passing up of normative fluents according to $\mathcal{CX}^i$ and the definition of a multi-tier model. The rules ensure that if a normative fluent is initiated by the tier below it is also initiated by the tier above, and likewise for termination. The rule for obligation initiation is given below, further rules we do not include 'pass up' obligation termination, and prohibition initiation and termination.

```
initiated(obl(A, D), In_2, I) :-
   initiated(obl(A, D), In_1, I),
   ifluent(obl(A,D), In_2),
   tier(In_2, I_1), tier(In_2, I_2), I_2 == I_1 + 1.
```

A multi-tier institution is modelled by providing a composite event trace and then producing the answer-set reflecting the formal multi-tier institution model. To define and generate a compos-

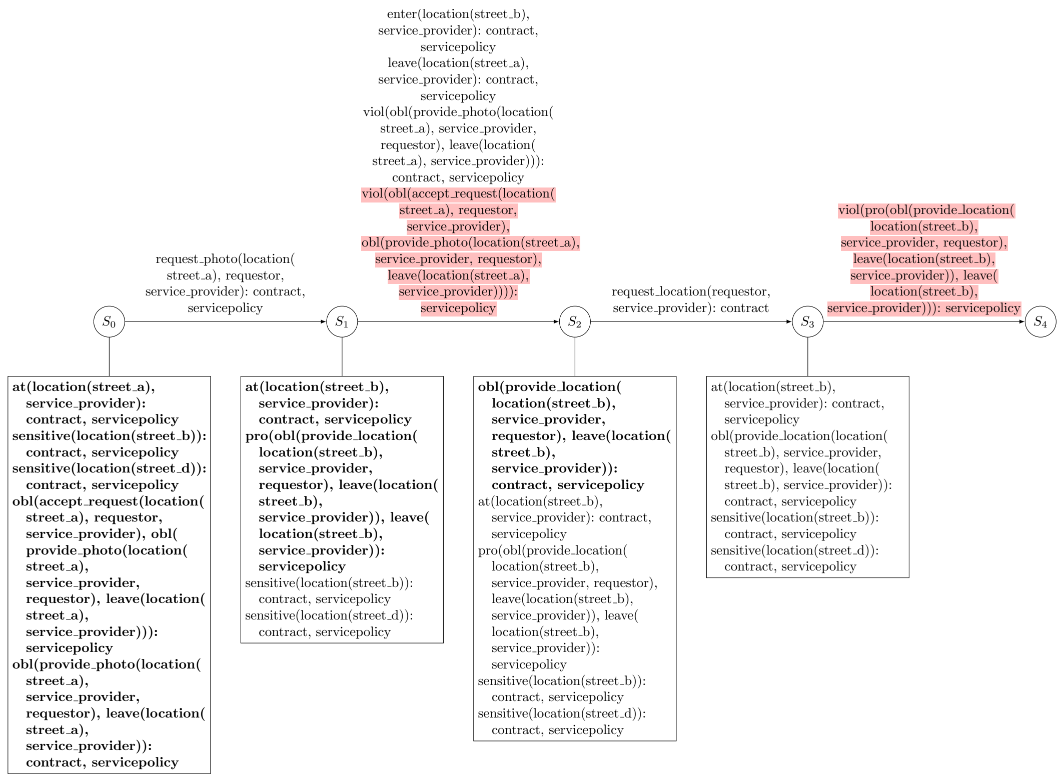

Figure 2: Results for a trace, highlighting violated second-order norms in the second-tier institution.

ite trace we provide a set of rules taken from modelling composite institutions in ASP [17]. These rules produce a composite trace where at each time point only one observable event occurs and it is recognised by at least one institution, and a synchronised trace for each institution such that each event not recognised is replaced with a null event. We assume a timeline is supplied to ground the variables for `instant/1`, `start/1`, `final/1` and `next/2`.

```
{compObserved(E, J)} :- evtype(E,In,ex),
  instant(J), not final(J), inst(In).
:- compObserved(E,J),compObserved(F,J),
  instant(J),evtype(E,InX,ex),
  evtype(F,InY,ex), E!=F,inst(InX;InY).
obs(I):- compObserved(E,I),evtype(E,In,ex),
  instant(I),inst(In).
:- not obs(I), not final(I), instant(I), inst(In).
observed(E,In,I) :- compObserved(E,I),
  inst(In), instant(I).
occurred(null,In,I) :- not evtype(E,In,ex),
  observed(E,In,I), instant(I), inst(In).
```

These rules produce answer-sets for all composite traces up to length `T`. If we are only interested in specific traces, we can supply a trace with `compObserved(E,I)` predicates. Alternatively, a constraint can limit traces to just those that produce at least one higher-order norm violation, such as given below, assuming `norm_order/2` defines each norm's order:

```
n_tier_violation :- occurred(viol(N), In, T),
  instant(T), tier(In, I), norm_order(N, I), I > 1.
:- not n_tier_violation.
```

5.2 Results: Monitoring the Contract and Service Policy

The results of translating the formalised example into a corresponding ASP program are given graphically in Figure 2, for a specific trace encoded in ASP below with `compObserved`.

```
compObserved(ex_request_photo(location(street_a),
             requestor, service_provider), 0).
compObserved(ex_enter(location(street_b),
             service_provider), 1).
compObserved(ex_request_location(requestor,
             service_provider), 2).
```

The results show the first-tier is non-compliant with the second with two highlighted second-order norm violations. The first service policy violation occurs in the trace because when the service provider is requested to provide a photograph, a first-order obligation to provide a photograph before leaving the area is initiated in S_0. Yet, simultaneously, a second-order norm is initiated obliging the request is accepted before such an obligation is imposed.

The second service policy violation occurs in the trace because when the agent enters a new location *street_b*, a second-order norm prohibiting an obligation for the service provider to provide their location is initiated in state S_1 , since *sensitive*(*street_b*) also holds. Yet, the service requestor then requests the service provider's location, initiating in state S_2 an obligation on the service provider to provide their location. This causes the second-order norm prohibiting such an obligation to be violated in state S_3.

If the contract is accepted as it is, then the service provider would be obliged to provide photographs regardless of if they had accepted the request, and they would be obliged to reveal their location when requested even if in a sensitive location. To resolve this the compliance check's results are used as input for a revision for compliance process, we give the output of in the next section.

6. ILP REVISION FOR COMPLIANCE

When any non-compliance is detected via the monitoring process outlined in Section 5.1, we then need a way to *revise* the non-compliant ith-tier institution to be consistent with the institution it is governed by. We use an *Inductive Logic Programming* (ILP) based approach for norm revision, which provides a suitable solution for our computational framework due to its implementation in ASP [8, 16]. This involves using the results of compliance failure produced by the computational framework as input for an ILP norm revision process, which due to space considerations we provide the results of rather than detail in full.

In the preceding section the running example was formalised in ASP (5.2). This gave us a non-compliance trace leading to two second-order norm violations in the second-tier, due to the contract obliging the service provider to provide a photo before they have accepted a request, and obliging the service provider to reveal their location even if they are in a sensitive location. The idea is that, ILP norm revision is provided a cost of a revision, such as the number of literals added and removed, and examples of undesirable properties for a trace, in our case the automatically detected second-tier violations. Then, ILP searches for all possible minimal revisions of the institution that resolve the undesirable properties (non-compliance) for a trace. In our case, two suitable suggestions are provided (of several) for the results from the previous section.

One suggestion is to revise the rule initiating the obligation to provide a photo by removing the body literal for the request photograph event occurrence. Then, replacing it with the body literal denoting the occurrence of the provider *accepting* the request:

```
initiated(obl(provide_photo(location(L),X,Y),
              leave(location(L),X)),contract,I):-
  occurred(request_photo(location(L),Y,X),contract,I),
  occurred(accept_request(location(L),Y,X),contract,I)),
  instant(I), agent(X;Y).
```

The second suggestion is initiating the obligation to provide locations only when the service provider's location is not sensitive.

```
initiated(obl(provide_location(location(L),X,Y),
              leave(location(L), X)), contract, I) :-
  occurred(request_location(Y,X),contract,I),
  not holdsat(sensitive(location(L)), contract, I),
  holdsat(at(location(L), X), contract, I),
  instant(I), agent(X;Y).
```

By using our computational framework we are able to automatically provide the required examples of undesirable properties, non-compliance of the first-tier with the second in this case. In turn, by using ILP-based norm revision, we are able to automatically revise the two-tier institution to be compliant, making the service contract acceptable to the service provider.

7. RELATED WORK

To the best of our knowledge, there are no frameworks for reasoning about institutions governing institutions or multi-tier institutions. Thus, the most closely related work is on authority hierarchies, legislating the legislators and interacting institutions, with less closely related work focussing on governing MAS.

Boella and van der Torre [3] formalise normative systems with authority hierarchies and the relationship between permissions and obligations. They use what they call 'meta-norms', which are not norms in the deontic sense but instead describe the hierarchical relationship between norms issued by different authorities. Then, permissions from a higher authority *derogate* obligations to the contrary issued by a lower authority. In comparison, we address the autonomy of institutional designers, using higher-tier institutions to govern lower-tiers in the norms they impose with *violatable*, rather than derogative, higher-order norms and a vertical governance structure for higher-tier institutions monitoring lower-tiers.

Lopez and Luck [19] provide a representation for different norm categories including *legislative norms* which describe what norm rule creation/abolishment actions are permitted. In contrast, our approach is not to govern rule change but rather the normative effects of rules. This is by imposing higher-order norms in some contexts which can be discharged and violated when lower-order norm rules are *invoked*. Whilst our approach focuses on the effects rather than existence of rules, it can then be used to determine if adding or removing an institutional rule causes the institution to be non-compliant in some circumstances and therefore whether a specific institutional change action is non-compliant.

Li [15] looks at the problem of agents governed by multiple interacting institutions which being designed by separate people are liable to collectively impose conflicting norms. Like us, Li also uses the InstAL framework, in her case to detect conflicts between institutions. The main difference is that Li looks at the interaction between institutions (for example, the event in one triggering a change in the other) but not the governance of institutions by institutions, and not the imposition of higher-order norms.

Further afield, our framework bears *some* similarities to modelling social commitments [6, 28] and King et al. 'meta-norms' [14] in the Event Calculus, since we use EC-like constructs; agent compliance monitoring [5, 20], rather than *institution* compliance; and detecting and resolving normative conflicts [13, 21, 26], rather than higher-order norm *violation* and revision for compliance.

8. CONCLUSIONS

This paper addressed the problem of guiding institutions towards imposing desirable norms, with a novel framework for the representation and operationalization of multi-tier institutions. Taking the InstAL framework for institutions governing MAS, we generalised it to also model institutions governing other institutions. Then, the representation and modelling of multi-tier institutions was provided, where institutions govern other institutions with higher-order norms and monitor them for whether the norms they impose are compliant. We then showed how this provides results for automatic norm-revision of an institution to be compliant, giving suggestions to institutional designers to resolve the *cause* of non-compliance. Consequently we automated *guiding* and supporting rather than regimenting compliant institution design.

The framework proposal has addressed one part of institutional design and governance, we note there are other institutional design considerations for institution designing agents. One factor is the cost of remaining non-compliant before a deadline, for a contract offer this is rejection, but in other cases it invites punishment. Another factor is if multiple-tier institutions are non-compliant, which to revise first. Whilst it makes sense to resolve highest non-compliant tier first, this depends on which institutions the designer has the power to effect change.

For future work it will be interesting to formalise more complex case studies (e.g. EU Directives) and multi-institution system coordination where multiple institutions occupy each tier. Another aspect is extending the representation and reasoning to handle punishment for non-compliance after a runtime deadline. The framework can perform pre-runtime checks that create higher-order norm violations due to the *invocation* of rules. However, violations of instruments such as EU Directives occur not when rules in an institution are invoked, but due to legislation as a whole being non-compliant.

Acknowledgments

This work is partially supported by the SHINE[1] project funded by TU Delft.

[1] http://shine.tudelft.nl

REFERENCES

[1] G. Andrighetto, G. Governatori, P. Noriega, and L. van der Torre. Normative Multi-Agent Systems. *Dagstuhl Follow-Ups*, 4, 2013.

[2] C. Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, Cambridge, 2003.

[3] G. Boella and L. 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.

[4] G. Boella and L. van der Torre. Contracts as Legal Institutions in Organizations of Autonomous Agents. In *AAMAS 2004*, pages 948 – 955, 2004.

[5] N. Bulling, M. Dastani, and M. Knobbout. Monitoring norm violations in multi-agent systems. In *AAMAS 2013*, pages 491–498, 2013.

[6] F. Chesani, P. Mello, M. Montali, and P. Torroni. Representing and monitoring social commitments using the event calculus. *Autonomous Agents and Multi-Agent Systems*, 27(1):85–130, June 2012.

[7] O. Cliffe, M. De Vos, and J. Padget. Answer set programming for representing and reasoning about virtual institutions. *Computational Logic in Multi-Agent Systems*, 4371:60–79, 2006.

[8] D. Corapi, A. Russo, M. De Vos, J. Padget, and K. Satoh. Normative design using inductive learning. *TPLP*, 4-5:783–799, 2011.

[9] T. Eiter, W. Faber, N. Leone, and G. Pfeifer. The diagnosis frontend of the dlv system. *AI Communications*, 12(1):99–111, 1999.

[10] R. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, Nov. 2011.

[11] M. Gebser, B. Kaufmann, and R. Kaminski. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107 – 124, 2011.

[12] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Logic Programming, Proceedings of the Fifth International Conference and Symposium*, pages 1070 – 1080, 1988.

[13] G. Governatori, M. Palmirani, R. Riveret, A. Rotolo, and G. Sartor. Norm modifications in defeasible logic. In *In Proceedings of JURIX'05*, pages 13–22, 2005.

[14] T. C. King, M. B. V. Riemsdijk, V. Dignum, and C. M. Jonker. Supporting Request Acceptance with Use Policies. In *COIN @ AAMAS 2014 Pre-proceedings*, 2014.

[15] T. Li. *Normative Conflict Detection and Resolution in Cooperating Institutions*. PhD thesis, University of Bath, 2014.

[16] T. Li, T. Balke, M. De Vos, J. Padget, and K. Satoh. A model-based approach to the automatic revision of secondary legislation. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*, pages 202–206. ACM, 2013.

[17] T. Li, T. Balke, M. De Vos, K. Satoh, and J. Padget. Detecting conflicts in legal systems. *New Frontiers in Artificial Intelligence*, pages 174 – 189, 2013.

[18] H. Liesbet and M. Gary. Unraveling the central state, but how? Types of multi-level governance. *American political science review*, 97(2):233–243, 2003.

[19] F. L. y. López and M. Luck. Modelling Norms for Autonomous Agents. In *Proceedings of The Fourth Mexican Conference on Computer Science*, pages 238–245. IEEE Computer Society, 2003.

[20] S. Modgil, N. Faci, and F. Meneguzzi. A framework for monitoring agent-based normative systems. In *AAMAS 2009: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 153–160. IFAAMAS, 2009.

[21] N. Oren, M. Luck, S. Miles, and T. Norman. An argumentation inspired heuristic for resolving normative conflict. In *Proc. of the 5th Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN@AAMAS)*, Estoril, Portugal, 2008.

[22] D. Ruiter. A basic classification of legal institutions. *Ratio Juris*, 10(4):357 – 372, 1997.

[23] J. R. Searle. *Speech acts: An essay in the philosophy of language*. Cambridge university press, 1969.

[24] The EU Member States. Treaty and Functioning of the European Union, 1958.

[25] UK Government. Financial Services Act, 2012.

[26] W. W. Vasconcelos, M. J. Kollingbaum, and T. J. Norman. Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 19(2):124–152, Nov. 2009.

[27] M. Wooldridge. *An Introduction to Multi-Agent Systems*. Wiley, 2002.

[28] P. Yolum and M. 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.